Contacting Global Technical Support

If you have current support and maintenance agreements with Rocket Software, you can access the Rocket Customer Portal and report a problem, download an update, or read answers to FAQs. The Rocket Customer Portal is the primary method of obtaining support.

To log in to the Rocket Customer Portal or to request a Rocket Customer Portal account, go to www.rocketsoftware.com/support

Alternatively, you can contact Global Technical Support by email or telephone:

Email: support@rocketsoftware.com
Telephone: 1 855 577 4323 (Toll-free US) or +1 781 577 4323 (International)
# Contents

Notices............................................................................................................................................................................. 2
Contacting Global Technical Support........................................................................................................................................ 3
About this information............................................................................................................................................................. 14
  Who should read this information........................................................................................................................................ 14
  Conventions and terminology used in this information........................................................................................................ 14
  Accessibility features............................................................................................................................................................ 14
  How to send comments....................................................................................................................................................... 14
Chapter 1: CorVu overview........................................................................................................................................................... 15
  User component............................................................................................................................................................... 15
  Administrator component.................................................................................................................................................. 16
  Perspectives, Views, and Editors.................................................................................................................................. 16
  Mobile client...................................................................................................................................................................... 18
Chapter 2: Working with perspectives, views, and editors........................................................................................................... 19
  Opening perspectives......................................................................................................................................................... 19
  Closing perspectives......................................................................................................................................................... 19
  Opening views................................................................................................................................................................. 20
  Positioning views............................................................................................................................................................ 20
  Configuring perspectives.................................................................................................................................................. 20
  Resetting perspectives...................................................................................................................................................... 21
  Setting preferences.......................................................................................................................................................... 21
    Setting general preferences........................................................................................................................................ 22
    Setting appearance preferences..................................................................................................................................... 22
    Setting dashboard runtime preferences...................................................................................................................... 23
    Setting help preferences............................................................................................................................................ 23
      Including help content from a remote server............................................................................................................. 24
    Setting JDBC preferences............................................................................................................................................ 25
    Setting LOB preferences............................................................................................................................................. 26
    Setting log preferences................................................................................................................................................ 27
    Setting the directory location preference for objects generated by Export actions................................................... 28
    Setting single sign-on preferences............................................................................................................................. 28
    Setting third party library preferences..................................................................................................................... 29
    Setting Visual Designer preferences.......................................................................................................................... 29
  Working in the Workspaces view........................................................................................................................................ 30
    Discovering the contents of a workspace.................................................................................................................... 30
    Workspace menu commands........................................................................................................................................ 31
    Managing workspace folders.......................................................................................................................................... 32
    Filtering data source objects.......................................................................................................................................... 33
    Creating links to workspace objects........................................................................................................................... 33
    Creating links to repository objects................................................................................................................................ 33
    Viewing properties for workspace objects................................................................................................................... 34
    Refreshing workspace contents........................................................................................................................................ 34
    Moving workspace objects.......................................................................................................................................... 35
    Adding content to a workspace...................................................................................................................................... 35
    Creating additional workspaces..................................................................................................................................... 36
      Adding a workspace..................................................................................................................................................... 36
  Working in the Personal view.................................................................................................................................................. 36
    Personal view menu commands..................................................................................................................................... 36
  Working with database tables................................................................................................................................................ 37
  Working in the Personal view................................................................................................................................................ 36
  About repositories............................................................................................................................................................... 39
  Creating personal repositories............................................................................................................................................. 39
Connecting to a repository................................................................. 40
Specifying user information.................................................................. 41
Changing the application configuration information.......................... 41
  Setting up JDBC driver information.................................................. 41
  Importing the JDBC library settings file............................................. 41
  Setting JDBC preferences................................................................. 42
  Setting up repository connection information.................................. 43
  Creating connections by importing the repository connections file...... 43
  Creating the repository connection for CorVu.................................. 44
  Creating shared repository connections......................................... 44
  Creating file-based repository connections...................................... 45
  Creating network repository connections........................................ 46
  Creating web service repository connections.................................... 48
  Managing repository connections.................................................... 48
  Creating web links to data objects.................................................... 49
  Exporting mobile connection profile............................................... 50
  Configuring repository caching....................................................... 51

Changing your DB2 password............................................................. 53
  Downloading Rocket CorVu to your mobile device......................... 54

Chapter 3: Creating new objects or repository items................................. 55

Opening saved objects.......................................................................... 56
  Opening repository objects................................................................ 56
  Opening objects saved to a file......................................................... 57
Creating queries................................................................................... 57
  Creating queries using SQL.............................................................. 57
  Using Content Assist for queries and visual queries....................... 58
  Using Parameter Hints for queries and visual queries...................... 58
  Creating visual queries using the Diagram Query editor................... 59
  Adding tables using the Tables window............................................. 61
  Editing join conditions for multiple tables....................................... 62
Specifying substitution variables......................................................... 62
Setting fonts for SQL statements......................................................... 63
Query menu.......................................................................................... 63
Creating OLAP queries........................................................................ 64
  Retrieving cube data....................................................................... 64
  Opening existing OLAP queries....................................................... 65
  Modifying OLAP queries to obtain different views of cube data....... 65
    Building the OLAP query offline.................................................. 65
    Building an OLAP query online.................................................... 66
    Filtering cube data................................................................-------- 67
Working with U2 queries...................................................................... 68
  Creating U2 Queries with the Query Diagram editor....................... 68
  Creating U2 Queries with the SQL editor.......................................... 69
  Opening an existing U2 file............................................................... 69
  Retrieving data from a U2 query....................................................... 70
Updating Rocket U2 database server programs.................................... 71
Working with simple prompts............................................................. 72
  Adding simple prompts................................................................... 72
  Editing simple prompts.................................................................. 73
  Saving prompt values with a query.................................................. 74
Working with prompt hierarchies......................................................... 74
  Creating prompt hierarchies............................................................. 75
  Editing existing prompt hierarchies................................................ 76
Analytical queries and associated query types.................................... 77
  Adding an append query................................................................. 78
  Adding a join query...................................................................... 79
Contents

Creating Charts.............................................................................................................................................. 115
Filtering query results................................................................................................................................. 112
Results menu................................................................................................................................................ 110
Viewing the contents of a cell in the query results.................................................................................... 109
Working with LOB data in the query results.............................................................................................. 108
Specifying format options........................................................................................................................... 117
Specifying data series options.................................................................................................................... 116
Specifying format options........................................................................................................................... 117
Specifying Series chart format options...................................................................................................... 117
Specifying Category chart format options.................................................................................................. 118
Specifying Value(Y)Series chart format options........................................................................................ 118
Adding a crosstab query............................................................................................................................... 79
Adding an analytical query......................................................................................................................... 80
Adding a normalize query............................................................................................................................ 80
Adding a conditional grouping.................................................................................................................... 83
Adding a column filter query....................................................................................................................... 85
Selecting a new data source......................................................................................................................... 86
Setting fonts for SQL statements............................................................................................................... 86
Specifying substitution variables.............................................................................................................. 87
Correct format for identifiers..................................................................................................................... 87
Matching patterns....................................................................................................................................... 88
About data templates................................................................................................................................. 88
Profiling queries......................................................................................................................................... 89
Saving objects............................................................................................................................................. 89
  Saving objects in a repository or a repository workspace....................................................................... 90
  Saving objects to a file............................................................................................................................ 90

Chapter 4: Working with query results........................................................................................................ 91
  Working with calculated columns............................................................................................................ 91
    Adding calculated columns to a query result set................................................................................ 92
    Adding user-defined functions.............................................................................................................. 93
  Applying grouping and aggregation to query result columns............................................................... 94
    Grouping and aggregation fields........................................................................................................... 94
  Customizing how query results display.................................................................................................. 95
    Specifying general column formatting options................................................................................ 97
    Specifying font formatting options.................................................................................................... 97
      Font fields......................................................................................................................................... 98
    Specifying conditional formatting options........................................................................................ 98
  Exporting query results............................................................................................................................. 99
    Exporting query results to a database.................................................................................................. 99
    Exporting query results to a file............................................................................................................ 100
      Specifying file export options........................................................................................................ 102
    Importing query results from data files................................................................................................. 104
      Importing query results from CSV and text files........................................................................... 105
        Specifying separator options for CSV and TXT files................................................................ 105
        Specifying the result set structure of CSV and Text files.......................................................... 106
      Opening data files............................................................................................................................. 107
      Converting data files to dynamarts.................................................................................................. 107
      Managing source data files in dynamarts......................................................................................... 108
    Working with LOB data in the query results....................................................................................... 108
      Overriding LOB resource limit option.............................................................................................. 109
    Viewing the contents of a cell in the query results............................................................................. 109
    Viewing multiple result sets.................................................................................................................. 110
  Results menu............................................................................................................................................ 110
  Sorting query results............................................................................................................................... 112
  Filtering query results.............................................................................................................................. 112
    Building a compound condition filter................................................................................................. 113
    Building a free style condition filter.................................................................................................... 114

Chapter 5: Working with charts.................................................................................................................... 115
  Creating Charts....................................................................................................................................... 115
    Creating Charts.................................................................................................................................... 116
      Specifying chart type options........................................................................................................... 116
      Specifying data series options.......................................................................................................... 117
      Specifying format options................................................................................................................ 117
      Specifying Series chart format options........................................................................................... 117
      Specifying Category chart format options....................................................................................... 118
      Specifying Value(Y)Series chart format options............................................................................ 118
Specifying forecasting models options.................................................................156
Setting simple moving average parameters.......................................................157
Setting weighted moving average parameters....................................................157
Setting single exponential smoothing parameters.............................................157
Setting double exponential smoothing parameters.........................................158
Setting Holt-Winters method parameters..........................................................158
Setting multiplicative decomposition parameters.............................................159
Setting curve fitting parameters......................................................................159
Setting polynomial regression parameters......................................................160
Setting neural network parameters.................................................................160
Monitoring forecast model performance............................................................161
Observing trends...............................................................................................161
Observing seasonality......................................................................................162
Observing cyclicity...........................................................................................162
Editing chart properties...................................................................................163
Editing grid properties.....................................................................................163
Exporting forecasts.........................................................................................164
Specifying print options for a forecast..............................................................164

Chapter 9: Working with scheduled tasks..........................................................165
Adding an execute object action........................................................................165
  Specifying export query operation options......................................................166
  Specifying export query to mail operation options.........................................167
  Specifying print query operation options.......................................................169
Specifying export quick report to mail operation options...............................169
Specifying export quick report operation options.............................................170
Specifying export visual report to mail operation options................................171
Specifying export visual report operation options............................................172
Specifying export visual portfolio to mail operation options............................173
Specifying export visual portfolio operation options........................................174
Specifying export visual report operation options............................................174
Adding a resource manipulation task................................................................175
  Specifying copy resource parameters...........................................................176
  Specifying mail settings................................................................................177
Adding an action group....................................................................................177
  Action group properties................................................................................178
  Adding actions to an action group................................................................178
  Removing actions from an action group.........................................................179
  Setting Iterator parameters..........................................................................179
  Managing prompts in an action group...........................................................180
Specifying parameter values for actions and action groups............................181
Copying scheduled tasks................................................................................182
Scheduling tasks.............................................................................................182
Sending HTML tables in email........................................................................182
Working with remote scheduled tasks..............................................................183

Chapter 10: Working with visual projects.........................................................184
Visual project templates..................................................................................185
Creating template categories for visual projects..............................................186
Editing visual project connection data for Rocket CorVu.................................186
Working with visual reports............................................................................186
  Creating reports using the Visual Report wizard............................................187
  Creating a new visual report template...........................................................188
Designing the main page of a visual report......................................................189
Designing a fixed page for a visual report.......................................................190
Managing report data retrieval...................................................................191
  Assigning data retrieval schedules to visual report queries........................192
Chapter 11: Working with visual portfolios

Creating visual portfolios................................................................. 332
Creating a visual portfolio document.................................................. 332
Adding content to a visual portfolio................................................... 333
  Adding pages to a visual portfolio...................................................... 333
  Adding chapter headers................................................................. 334
  Adding sub-headers.................................................................. 334
  Adding CorVu objects................................................................ 335
  Adding OLE objects................................................................ 335
  Adding images........................................................................... 336
  Adding text objects................................................................ 336
Working with overlay pages......................................................... 337
Creating the Table of Contents.......................................................... 338
Managing visual portfolio pages...................................................... 338
Managing parameters for CorVu objects............................................ 339
Printing visual portfolios................................................................. 339
Publishing visual portfolios to PDF................................................... 339
Using visual portfolio in procedures and scheduled tasks............... 340

Chapter 12: Working with JavaScript Tables

Creating JavaScript Tables............................................................... 341
Writing scripts to populate JavaScript Table....................................... 341
  replaceData()........................................................................ 342
  appendData()...................................................................... 342
  eraseData()........................................................................ 343
  setTag()............................................................................ 343
  getTag()............................................................................ 344
  include()............................................................................ 344
  log().................................................................................. 344
  RSBIHttpRequest class............................................................ 345
JavaScript Table Example............................................................... 345
Editing JavaScript Tables................................................................. 348

Chapter 13: Text Analytics.............................................................. 349
Using Text Analytics.................................................................... 349
Appendix A: Accessibility..................................................................................................................................................398
  Standard keyboard equivalents in CorVu Workstation Pro..........................................................................................398
  Standard keyboard navigation.........................................................................................................................................398
  Operating system accessibility......................................................................................................................................398
  Assistive technology products......................................................................................................................................399
  Navigation in CorVu Workstation Pro..........................................................................................................................399
  Navigation in CorVu Web Pro.......................................................................................................................................401
Appendix B: Messages......................................................................................................................................................402
Appendix C: Glossary........................................................................................................................................................491
  perspectives..............................................................................................................................................................495
Index..................................................................................................................................................................................499
About this information

This information describes how to use CorVu Workstation Pro and CorVu Web Pro. It includes information about:

- The CorVu Workstation Pro user interface, including information about views, perspectives, and editors.
- Creating and working with queries, reports, procedures, visual dashboards and visual applications.
- Using CorVu Workstation Pro Text Analytics.

Who should read this information

This information is intended for persons using CorVu Workstation Pro or CorVu Web Pro to create queries, reports, procedures, visual dashboards, and visual applications.

Conventions and terminology used in this information

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.
- **Monospaced type** Used for system messages and command syntax. indicates examples of text that you enter exactly as shown such as commands and system messages.
- **Italic type** indicates variables that you should replace with a value, identifies publications, emphasizes significant words and specifies optional steps in a task.

Accessibility features

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use a software product successfully. The major accessibility features in CorVu applications enable users to:

- Use assistive technologies such as screen readers and screen magnifier software.
- Operate specific or equivalent features using only the keyboard.

How to send comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this information, send an email to http://www.rocketsoftware.com/support and include the name of the product and the topic title along with your comment.
Chapter 1: CorVu overview

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

It consists of the following applications:

- **CorVu Workstation Pro:**
  CorVu Workstation Pro is an Eclipse-based, cross-platform, rich workstation application.
- **CorVu Web Pro:** CorVu Web Pro is an application-server-based solution that extends CorVu key function to enterprise users by using a “thin client,” browser-based deployment model.

CorVu Workstation Pro and CorVu Web Pro include a User and Administrator component.

With CorVu you can transform business data into a visual information platform with visual data on-demand. You can create intuitive dashboards, queries, reports, and procedures that can be easily distributed across an entire enterprise.

CorVu:

- Delivers critical enterprise-wide business information across multiple end user and database platforms.
- Supports dynamic and static SQL, massive result sets, and stored procedure results.
- Supports simultaneously drawing data from DB2, Informix, Oracle, SQL Server and most other JDBC-compliant data sources, as well as multi-dimensional data sources that support XMLA connectivity.
- Reports, dashboards, visual applications, and queries can be optionally embedded within custom or third-party web applications, portals or workstation applications.
- Supports multi-dimensional queries, dashboards, visual applications reports, and layouts, as well as enhanced interoperability with existing BI infrastructure.
- CorVu Workstation Pro is an Eclipse-based, cross-platform, rich workstation application that can be run on Microsoft® Windows®, Linux®, and Macintosh®.
- CorVu Web Pro delivers broad functionality to enterprise users through ordinary Web browsers.

**User component**

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

To get started using CorVu, you should become familiar with the **User** component, its associated perspectives and views, the preferences that you can set, and how you can work with the data sources and objects that are available to you in the **Workspaces**. At any time, you can begin querying data sources, creating objects, formatting query results, or creating reports, dashboards or visual applications.

**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.
Chapter 1: CorVu overview

Administrator component

The Administrator component provides the views, menus, editors, and wizards that enable an administrator to configure and manage CorVu Workstation Pro and CorVu Web Pro.

An administrator uses the Administrator component to configure repositories, connect to databases and populate repositories with the data sources, workspaces and environments that define how users access and work with data. The data sources and objects that user access using CorVu Workstation Pro and CorVu Web Pro are displayed in the Workspaces view.

 Perspectives, Views, and Editors

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

Perspectives

CorVu provides the following perspectives:

- **Administrator** which provides the views, menus, and wizards that enable a user with administrative privileges to create and maintain repositories. Not all instances of CorVu are installed with the Administrator perspective as a selected feature.
- **Text Analytics** which provides the interface from which you create and work with Text Analytics projects.
- **User** which provides the views, menus, and wizards that you will use to query data sources and work with query results.
  
  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.
- **Visual Designer** which provides the views, menus, and wizards that you will use to create visual reports and dashboards.
- **The Scene Script** which provides the interface from which you can create and work with visual dashboards or applications by using JavaScript dynamically

Views

CorVu 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 all perspectives have a designated set of Views, which is predetermined by particular functions.

Views are represented as tabs in the user interface.

Each perspective of CorVu initially displays the views that are most commonly used to perform specific functions.

<table>
<thead>
<tr>
<th>View Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analyzer</td>
<td>The Analyzer view displays the weight count of each object lying on a scene.</td>
</tr>
<tr>
<td>Bookmarks</td>
<td>The Bookmarks view manages bookmarks for drill-down paths.</td>
</tr>
<tr>
<td>Cheat Sheets</td>
<td>The Cheat Sheets view displays a series of tutorials that outline the steps required to implement several basic user and administrator functions.</td>
</tr>
</tbody>
</table>
### Table 2: Administrator perspective: Views

<table>
<thead>
<tr>
<th>View</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory Usage</td>
<td>The Memory Usage view displays the memory usage status and provides a garbage collector function to improve the application's performance.</td>
</tr>
<tr>
<td>Repositories</td>
<td>The Repositories view displays the repositories that you created. This view is only available if you have the Administrator component.</td>
</tr>
<tr>
<td>Users and Groups</td>
<td>The Users and Groups view displays the users and groups created for your internal security provider.</td>
</tr>
</tbody>
</table>

### Table 3: Visual Designer perspective: Views

<table>
<thead>
<tr>
<th>View</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Events</td>
<td>The Events view displays navigation elements that you can apply to layout objects within a visual dashboard or application.</td>
</tr>
<tr>
<td>Layers</td>
<td>The Layers view displays the groups of layers.</td>
</tr>
<tr>
<td>Palette</td>
<td>The Palette view displays the objects that you can insert in a visual project.</td>
</tr>
<tr>
<td>View</td>
<td>Purpose</td>
</tr>
<tr>
<td>---------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Project Explorer</td>
<td>The Project Explorer view displays each element in a visual report,</td>
</tr>
<tr>
<td></td>
<td>dashboard, or application.</td>
</tr>
<tr>
<td>Properties</td>
<td>The Properties view displays the properties of the object that is</td>
</tr>
<tr>
<td></td>
<td>selected or active in the editor or another view.</td>
</tr>
<tr>
<td>CorVu Calculator</td>
<td>The CorVu Calculator is used for calculating expressions in visual</td>
</tr>
<tr>
<td></td>
<td>projects.</td>
</tr>
</tbody>
</table>

Table 4: Text Analytics: Views

<table>
<thead>
<tr>
<th>View Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entities Explorer</td>
<td>The Entities Explorer view displays the tree structure of the entities</td>
</tr>
<tr>
<td></td>
<td>which are specified in the knowledge base.</td>
</tr>
<tr>
<td>Source Explorer</td>
<td>The Source Explorer view displays the tree structure of the files,</td>
</tr>
<tr>
<td></td>
<td>belonging to the projects.</td>
</tr>
<tr>
<td>Text Analytics Explorer</td>
<td>The Text Analytics Explorer view displays the list of projects that</td>
</tr>
<tr>
<td></td>
<td>can be analyzed.</td>
</tr>
<tr>
<td>Word Map</td>
<td>The Word Map view displays the text analysis results in the form of a</td>
</tr>
<tr>
<td></td>
<td>word cloud.</td>
</tr>
</tbody>
</table>

Table 5: Scene Script: Views

<table>
<thead>
<tr>
<th>View Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code Library</td>
<td>The Code Library view contains a library of script templates for objects.</td>
</tr>
<tr>
<td>Object Explorer</td>
<td>The Object Explorer view contains the objects added to a visual project</td>
</tr>
<tr>
<td></td>
<td>from the Palette view.</td>
</tr>
<tr>
<td>Scripts</td>
<td>The Scripts view hierarchically lists all scripts of open visual dashboards and applications, global scripts, which are connected with the current visual project, and scene scripts.</td>
</tr>
</tbody>
</table>

Editors

Each of the CorVu perspectives include a single editor area.

By default the editor area opens in the center of the workstation window.

There are many editors in CorVu 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, or creating reports.

There are no editors that will open when you are performing the administrative tasks using the Administrator perspective. The functions that you perform to create and manage repositories using the Administrator perspective will update views.

Mobile client

With Rocket CorVu you can view visual dashboards and visual applications on mobile devices such as the iPad.

Rocket CorVu is a free companion product for CorVu Web Pro. With Rocket CorVu you can view visual projects from anywhere at anytime using your mobile device.
Chapter 2: Working with perspectives, views, and editors

You use perspectives, views, and editors in the Administrator and User components.
An administrator uses the perspectives, views, and editors of the Administrator component to configure CorVu Workstation Pro and CorVu Web Pro and to export the configuration to users.
A user uses the perspectives, views, and editors of the User component to access data sources to create queries, reports, and visual dashboards and applications.

Opening perspectives

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

A perspective will stay open until you close it. Each time CorVu starts, the last opened perspective opens.

Initially, the User perspective opens when the CorVu 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 CorVu 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 icon. Clicking the toolbar 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 remain closed until you re-open it. Each time CorVu starts, all the perspectives that were closed during the last session will remain 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. Select **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 CorVu 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 CorVu Workstation Pro, 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 on the toolbar and the menu bar.

The aspects of a perspective that you can control:

- The icons 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.
2. Select **Window → Customize Perspective**.
   The Customize Perspective window opens.
3. Click 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 CorVu Workstation Pro, 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**

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

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

- **General**
  - On the **General** page, you can specify the orientation direction, locale settings, and object default actions.

- **Appearance**
  - On the **Appearance** page, you can specify the default fonts that will be used for query results headings and columns and the strings that will be used for displaying and entering null and default values.

- **Dashboard Runtime**
  - On the **Dashboard Runtime** page, you can determine how null values are displayed in dashboards and specify the Google Map API key that you will use to enable usage of the Google Map layout object in visual dashboards.

- **Global Variables**
  - On the **Global Variables** page, you can change the default values for system global variables and define user global variables.

- **Help**
  - On the **Help** page, you can 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. (CorVu Workstation Pro only).

- **JDBC Libraries**
  - On the **JDBC Libraries** page, you can manage the JDBC files and drivers that are available to the application interface.

- **LOBs**
  - On the **LOBs** page, you can specify options that will control retrieving and saving LOB data.

- **Log**
  - On the **Log** page, you can activate a log file that will track information useful in debugging.

- **SMTP**
  - On the **SMTP** page, you can specify the list of available SMTP servers.

- **Server-Side File System**
  - On the **Server-Side File System** page, you can set root export or import directories for objects.
Chapter 2: Working with perspectives, views, and editors

Single sign-on
On the Single sign-on page, you can enable Kerberos security support.

Third-Party Libraries
On the Third-Party Libraries page, you can specify any third party libraries.

UniVerse/UniData
On the UniVerse/UniData page, you can enable or disable automatic update check for programs installed on a U2 database server. For this, select or clear the Automatically check for updates to server programs check box.

Visual Designer
On the Visual Designer page, you can specify preferences that will effect the creation of visual projects. (CorVu Workstation Pro only).

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 CorVu application interface session:

1. Select View → Preferences. The Preferences window opens. Select General from the tree.
2. From the Language drop-down list, select the language that will be used for the current CorVu application interface session. Select the value Auto to use the local language of the current machine.
3. From the Direction drop-down list, select the orientation that will be used to display information in the current CorVu 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. From the Locale setting for numbers, currencies, times, and dates drop-down list, select the language that will be used to display numbers, currencies, times and dates in the current CorVu 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 CorVu 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 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 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 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 CorVu application interface session:

1. Select View → Preferences. The Preferences window opens. Select Appearance from the tree.
2. From the Headings of query results drop-down list, select the font that will be used for all query results headings.
3. From the **Columns of query results** drop-down list, select the font that will be used for all query results column data.

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

5. 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.

6. 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.

7. From the **Color scheme in text editors** drop-down list, select what type of color scheme will be supported in the text editors. In most instances you will select **CorVu** 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**.

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 dashboard runtime preferences**

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

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

1. Select **View → Preferences**. The Preferences window opens. Select **Dashboard Runtime** from the tree.

2. 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.
   
   You can obtain the Google Map APIs from the Google Code web site.

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 help preferences**

If you are using CorVu Workstation Pro, 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 the **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 the **Open window context help in an infopop** from the drop-down list.
4. By default, context-sensitive help for windows is displayed in the dialog’s tray, which is similar to the Help view. If you prefer to display window context-sensitive help in infopops, select the 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 window 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 active.
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. Click Use port and then specify the number of a specific port the server will listen to.
8. Click OK.
   The Add new infocenter window closes and control returns to the Preferences window. The new infocenter is displayed in the Remote Infocenters window.
9. To edit an infocenter, click Edit.
   The Edit [infocenter name] infocenter window opens.
10. To delete an infocenter, click Delete.
    The selected infocenter is removed from the list.
11. To view the properties of a specific infocenter, click View Properties.
    The Properties for [infocenter name] window opens with the name, host, path, port, and URL displayed.
12. To test an infocenter’s connection to the local IP interface, click Test Connection.
    The Test Connection window opens displaying the test results.
13. To disable an infocenter without deleting it, click Disable.
    The infocenter is now disabled, and its status in the Enabled column is changed to disabled.
14. Click Apply to save your preferences choices.
15. Click Restore Defaults to restore the default preference values.
16. Click OK to close the Preferences window.
Setting JDBC preferences

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

CorVu 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 CorVu. CorVu 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. Enter information about the new driver library and click OK.

   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 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.
   - 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.
   - 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.

4. If you are using CorVu Workstation Pro, 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.

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

6. 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 CorVu Workstation Pro, 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.
Chapter 2: Working with perspectives, views, and editors

- 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 CorVu Workstation Pro, to remove a JAR file from a JDBC driver library, select an existing file from the list and click **Remove**. The file is removed.

7. 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 window 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. Select **Override LOB options if possible** to grant yourself the ability to override the LOB options that have been specified by your CorVu 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 CorVu 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 CorVu session:

1. Select **View → Preferences**. The Preferences window opens. Select **Log** from the tree.

2. Select **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. Select one or more of the log file options to indicate what information should be gathered. It is recommended that all options remain selected. 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. Select **Enable report** to deliver the processing information in report format. If this check box is selected, the report file option fields are enabled. Select 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**. 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 a preference in CorVu so that objects generated from various CorVu actions are exported to a specific location on either the local file system or the server file system.

If you are using CorVu Workstation Pro, create a folder on the local file system.

If you are using CorVu Web Pro, 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 of preferences.
2. Select the **Enable access to the server file system** check box.
3. Type the path in the **Root output directory** field to indicate where generated objects will be exported.

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

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

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

Setting single sign-on preferences

You use the Single sign-on page of the Preferences window to enable Kerberos security support.

If you use the Kerberos Network Authentication Protocol to manage user login information, you can enable Kerberos single sign-on from the Preferences window. Kerberos is a computer network authentication protocol, which allows nodes communicating over a non-secure network to prove their identity to one another in a secure manner. To specify the single sign-on preferences:

1. Select **View > Preferences**. The Preferences window opens. Select **Single sign-on** from the tree.
2. Select the **Enable single sign-on support** check box to enable the current version of CorVu to connect to your Kerberos security client when the application is started.
3. Specify a string in the **Login context** field that will identify CorVu as a client to your Kerberos security system. This string will enable CorVu to communicate with Kerberos.
4. Click **Apply** to save your preferences choices.
5. Click **Restore Defaults** to restore the default preference values.
6. Click **OK** to close the Preferences window.
**Setting third party library preferences**

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

To specify the third party library preferences:

2. The third-party software whose location has been defined to CorVu is listed by name, package name and location. In most instances, the location information for third part libraries is specified when CorVu 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 window opens. You can change the path to the selected software in the Path field. You can change the packages name in the Packages (optional) field. Click OK. The changes that you have made are applied. Control returns to the Preferences window.
5. To remove third party library information, click the third party library in the Third party libraries list and click Remove. The third party library information is removed from the list.
6. Click Apply to save your preferences choices.
7. Click Restore Defaults to restore the default preference values.
8. Click OK to close the Preferences window.

**Setting Visual Designer preferences**

If you are using CorVu Workstation Pro, 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. Specify a string in the Display null values as field that CorVu will use to display null values that are retrieved from a database data source.
4. 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.
5. Expand Visual Designer in the tree and select Fonts.
6. 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.
7. Select Project Explorer from the tree.
8. 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.
9. 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.

10. 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.

11. Click **Apply** to save your preferences choices.
12. Click **Restore Defaults** to restore the default preference values.
13. Click **OK** to close the Preferences window.

### Working in 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 CorVu 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 (schemas and tables) that are on the data source.
   - The **ER Diagrams** folder lists the available ER diagrams. This folder is only included if the data source has been configured to enable ER diagrams.
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 any object listed in your workspace will either run or open the object. The default action for objects is set by using the Default Action for Objects radio buttons that can be found on the General page of the Preferences window. The Default Action for Objects radio buttons 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 CorVu 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 a:
  - Repository
  - File

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

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

**Run As**
- Runs the dashboard that you have selected in the Workspaces or Repository Explorer views in a specified rendering mode.

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

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

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

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

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

**Paste**

Paste 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.

**Publish Visual Project**

Publishes your visual project as a Rocket CorVu application that runs on a mobile device.

**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** view gives you specific information about your selection. The information provided varies based on your selection.

**Show in Repository**

Displays the object that a link entry in the **Workspaces** view points to in the **Repository Explorer** view.

---

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 CorVu 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 is 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 CorVu Workstation Pro, 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 click Browse to search for the folder.

4. Specify the object for which you want to create a link in the Linked object field. You can click Browse 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 window 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. In the Properties window 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 from 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. (CorVu Workstation Pro 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. (CorVu Workstation Pro 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 only have to 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.
Moving workspace objects

If you are using CorVu Workstation Pro, using your mouse, you can drag 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 CorVu 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

You can add a workspace to place and organize CorVu objects in a logical manner.

To add a workspace, 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.

You have added a workspace to the repository.

You can now add objects to 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 *Visual 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.
   - *Visual 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:

1. With the table opened in the *Table Editor*, select the *Results* tab.
2. 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.
### Editing content directly in the table

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

- Placing your cursor in the cell containing the information you want to edit, right-clicking and selecting **Edit** from the 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:
  
  1. 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.
  2. Right-click and select **Delete Row** from the context menu.
  3. 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:
  
  1. Click the row number of the row that you want to edit.
  2. Right-click and select **Edit Row** from the context menu.

  The row and its content are displayed in an Edit Row window.
  
  3. Make changes to data in the **Value** of the Edit Row window and click **OK**.
  4. Click **OK** to apply your updates and commit changes to the database, or click **Cancel** to rollback the changes.

<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.</td>
</tr>
<tr>
<td></td>
<td>Enter the search criteria in the Find dialog and click <strong>Find next</strong>.</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.</td>
</tr>
<tr>
<td></td>
<td>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>
• Inserting a row:
  1. 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 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.

• **Default Editor** to display the table using the editor that was last used.

---

**About repositories**

Content created within CorVu is normally stored in the CorVu repository, along with information related to the data sources available within the product.

The repository includes all the information CorVu needs to connect users to data sources. It includes all the information that the application needs to run and perform its query and reporting functions. In addition, it is where users' objects such as queries, forms, and reports are saved. CorVu supports personal and shared repositories. Personal repositories are local to individual users and shared repositories are located on a database and can be shared by many users.

Each user of CorVu must connect to a repository. You can have multiple CorVu repositories, containing different information such as different data sources that can be accessed.

You use the CorVu Administrator perspective to create and manage repositories. The CorVu Administrative perspective component is optional and cannot be installed on all computers. CorVu administrators are responsible for setting up repository connection information for all users that do not have the Administrator component installed.

**Creating personal repositories**

CorVu users can create their own personal repositories. Personal repositories are local to the individual user and are generally only used by that individual.

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

To create a personal repository:

1. Select **File → New → Other**. Expand the **Repository** folder, select **Personal Repository** and click **Next**.
The Create New Personal Repository window opens.

2. Specify the name for the repository in the **Name** field.

3. **Optional:** 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**.

You have created a personal repository.

The repository that you created is stored in the home directory of the machine that is running CorVu. You can view the repository by opening the Repository Explorer view of the workbench. To open the Repository Explorer view, select **Window → Show View → Other → CorVu Workbench → Repository Explorer**.

You can now save objects in your personal repository.

**Connecting to a repository**

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

A repository is a centralized storage area that was created by your CorVu administrator. It is the place where your objects such as queries, dashboards, and reports can be saved. It is also where CorVu 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 CorVu, 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 → Other**....

2. Expand **CorVu Workbench** and select **Repository connections**

3. Right-click the repository connection name for the repository to which you want to connect.

4. Select **Connect**. You are connected to the repository.

Administrators are responsible for creating repository connections for each user of CorVu. If necessary, each CorVu Workstation Pro 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 CorVu 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 CorVu 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 window 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 CorVu 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 if you want CorVu to 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 CorVu 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. CorVu will use this user information the next time it connects to the repository or data source.

Changing the application configuration information

The CorVu Workstation Pro 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 CorVu Workstation Pro 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 CorVu Workstation Pro users can change the application configuration information:

Setting up JDBC driver information

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

CorVu 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 configuration 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.

CorVu 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 click **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 by using the **Conflict resolving** radio buttons.
   - 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.

CorVu 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 CorVu. CorVu 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. Enter information about the new driver library and click **OK**.

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
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 CorVu administrator.

CorVu 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 CorVu.

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 click **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 by using the **Conflict resolving** radio buttons.
   - Click **Override existing settings** to replace all conflicting information with new information.
   - Click **Keep existing settings** to not update any existing information with new information.
   - Click **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 CorVu

CorVu 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 CorVu is to create it using the CorVu connection wizard.

CorVu administrators are responsible for distributing the connection information that is needed in order 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 **repositories.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 **Repository Connections** view and selecting **New → Repository Connection.**
2. Select the type of connection that you will create by clicking 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 CorVu Workstation Pro and the server (workstation) that runs CorVu Web Pro must have JDBC drivers installed.
   - **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 CorVu Workstation Pro. This type of connection is not applicable to CorVu Web Pro users.

- **Web service repository** to create a repository connection to a shared repository through the CorVu Web Pro 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 CorVu Workstation Pro. This type of connection is not applicable to CorVu Web Pro.

- **CorVu repository**. This type of connection provides a built-in repository for shared repository storage functionality, but does not require an external relational database.

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 page 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 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. From the Connection type radio group, click **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 selected JDBC:**
     - In the **JDBC Driver** field specify the library for the JDBC driver that will be used to access the database that is hosting the repository storage.
       You can select from a drop-down list of libraries that you defined when you set up the JDBC driver information. You can click the ellipsis (...) to open the JDBC Libraries window where you can create additional JDBC driver libraries.
     - 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.
Chapter 2: Working with perspectives, views, and editors

- 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.

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 repository in the **User name** and **Password** fields.

   c. If you are using CorVu Workstation Pro, 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. In the **Repository name** field, select the repository to connect to. If no repositories are listed, click **Refresh** to the right of the field.

9. 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 CorVu Workstation Pro, 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. Current versions of CorVu Workstation Pro and CorVu Web Pro 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. Specify the path to the directory where the file-based repository is saved in the **Base directory** field. You can click **Browse** to search your local files.

5. 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 CorVu Workstation Pro. This type of connection is not applicable to CorVu Web Pro.

CorVu Workstation Pro users can connect to repositories via a centralized machine running the CorVu 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 CorVu Workstation Pro (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 CorVu 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 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** to the right of the **Repository connection name** field. 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 CorVu Web Pro 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 CorVu Workstation Pro. This type of connection does not apply to CorVu Web Pro.

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

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

To complete the creation of a web service repository connection:

1. Ensure that CorVu Web Pro 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 CorVu Web Pro whose web services API will be used for this connection. This is the URL address that was created when the instance of CorVu Web Pro was deployed.
6. Click Refresh to the right of the Repository connection name field. The available repository connections that have been defined for the instance of CorVu Web Pro 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.

Managing repository connections

You manage your repository connections from the Repository Connections view.

Each CorVu 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 might have been supplied to you by the administrator.

2. To edit a repository connection’s property information, right-click the repository connection that you want to edit. Select **Edit** from the context menu. The Edit Repository Connection wizard opens. The Edit Repository Connection wizard presents edit windows based on the type of connection that you have selected. In these windows you can make changes to the repository connection information. Upon completion, the modified connection information is saved.

3. To remove a repository connection, right-click the repository connection that you want to remove. Select **Remove connection** from the context menu. A message asking you to confirm the removal is issued. 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. 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 CorVu objects. You can use these links to open the objects directly in web browsers.

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

To create a web link to data objects, perform the following procedure.

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**. When 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 by clicking **Open from Repository**).

3. Specify settings for the following **Predefined Parameters**:

   - The **Toolbar** check box to indicate whether to show or hide the toolbar. By default the check box is selected, which means the toolbar is shown.
   - The **Run** check box to indicate whether to run the object when it is opened. By default the check box is selected, which means the object is run.
   - 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.

   If you want to use an environment other than the default one, select the check box and select from the list of environments.
The check box is disabled if there are no environments.

- The **Renderer mode** check box is available for visual report 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:

   \[ http://\{host\}:\{port\}/\{app_context\}/g?staff&name=value \]

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

   On this page you can perform the following actions:

   - Edit existing logins and passwords to the data sources
   - Click **Add** to add logins and passwords to the data sources
   - Click **Delete** 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:

   \[ http://{host}:{port}/{app_context}/g?STAFF \]

   becomes:

   \[ http://{host}:{port}/{app_context}/g?staff \]

   If there is a link with the same name, the system prompts you to change the name. You can change the name explicitly 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 **Web link name**.

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

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

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

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

### Exporting mobile connection profile

You can export mobile connection profiles from CorVu Web Pro and copy them to your mobile device using appropriate tools (for example, iTunes if you are using an iPad).

This profile contains the information required for connecting Rocket CorVu to a specific instance of CorVu Web Pro.

To export a connection profile to your mobile client:

1. Select **File > Export > Mobile connection profile** to open the Mobile Connection Profile wizard.
2. Specify the **Profile name**.
If you are connected to a secured repository, indicate whether the repository username and password will be saved to the exported profile. To save the repository user name and password to the exported profile, select the **Export repository username and password** check box.

3. Click **Finish**. The Export wizard closes.

**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 (visual dashboards, visual applications, visual queries, visual reports, and other objects) to the memory on workstation client or on the web server when using CorVu Web Pro.

This task describes how to configure repository caching.

**Considerations:**
- Caching happens in local memory on your machine or in the memory on the web server (CorVu Web Pro),
  - 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.

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:
Chapter 2: Working with perspectives, views, and editors

- 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 CorVu Workstation Pro and on the web server if you are using CorVu Web Pro.
      
      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.

---

**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. In the Change Password window you can 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.
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.
5. Click Change. The Change Password window opens.
6. 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.

7. Type the new password in the New password field.
8. Retype the new password in the Confirm new password field.
9. Click OK. The Change Password window closes and the new password is confirmed.

**Downloading Rocket CorVu to your mobile device**

You can download Rocket CorVu from online distribution platforms such as the Android Market Business site for Android devices and the App Store for iPads.

For mobile devices running the iOS platform, download Rocket CorVu from Apple's Apps for Business site.

For mobile devices running the Android platform, download Rocket CorVu from the Android Market Business site.
Chapter 3: Creating new objects or repository items

CorVu supplies numerous wizards to help you create new objects.

In 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.
2. The wizards that are available are listed in the Wizards list box. Expand the CorVu Objects folder to see the wizards that are available to create the querying and reporting objects. Expand the Repository folder to see the wizards that are available to create objects that pertain to repositories. The wizards that are most frequently used to create CorVu objects are also listed outside their folders for access convenience.
3. If you are using CorVu Workstation Pro, from the Business Intelligence and Reporting Tools folder you can select 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.
4. If you are using CorVu Workstation Pro, from the Connection Profiles folder you can select one of the following wizards:
   - Connection Profile to create a new connection profile.
   - Connection Profile Repository to create a new connection profile repository.
5. From the CorVu Objects folder you can select one of the following wizards:
   - Analytical Query to create a new analytical query.
   - ER Diagram to create a new ER diagram.
   - Folder to add an ER Diagram folder to the data source.
   - Forecast to create a new forecast.
   - JavaScript Module to create a JavaScript module and open the JavaScript code editor.
   - JavaScript Table to create a JavaScript table.
   - OLAP Query to create a new OLAP Query.
   - Procedure to create a new procedure.
   - Prompt Hierarchy to create a new prompt hierarchy.
   - Visual Query to create a new SQL query.
   - Virtual Data Source Schedule to specify cache expiration settings for a virtual data source.
   - Quick Report to create a new quick report.
   - Drill-down Path to create a new drill-down path.
   - Visual Application to create a new visual application project. (CorVu Workstation Pro only)
   - Visual Application Template Category to create a new visual application template category. (CorVu Workstation Pro only)
   - Visual Dashboard to create a new dashboard visual project. (CorVu Workstation Pro only)
   - Visual Dashboard Template Category to create a new visual dashboard template category. (CorVu Workstation Pro only)
   - Visual Report to create a new visual report project. (CorVu Workstation Pro only)
   - Visual Report Template Category to create a new visual report template category. (CorVu Workstation Pro only)
6. From the **Environments** folder you can choose one of the following wizards:
   - **Data Source Mapping** to create a new data source mapping.
   - **Environment** to create a new environment.

7. 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.

8. 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.
   - **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.
   - **Multidimensional Data Source** to add a multidimensional data source to your repository.
   - **U2 Data Source** to add a U2 data source to your repository.

9. Click **Next**. The first page of the wizard that you selected will open. The first page will be different depending on the object or repository item that you are creating.

## 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 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.
   - **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 or drag the file from its current location to the editor area.

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

1. Select File → Open From → File.
2. Specify the file and its location in the Open window.
3. Click Open.

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 editor. 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.
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.

Using Content Assist for queries and visual queries

If you are using CorVu Workstation Pro, 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 CorVu Objects folder. Select Visual Query from the list of CorVu Objects.
2. Specify the name of the query in the Name field and where the query will be saved in the Data Source field. Click Finish.
3. Click the SQL tab.
4. At any time while typing the SQL statement, press Ctrl+Spacebar. A drop-down list opens underneath the cursor. The contents of the drop-down list depend on the location of the cursor in the SQL statement and the contents of the table being called. For example, if SELECT * FROM Q is displayed when Content Assist is selected, the drop-down list will display all tables in the data source with the owner name Q. If SELECT * FROM Q.STAFF WHERE is displayed, the drop-down list will display all columns of the Q.STAFF table.
5. If you select Content Assist while partially typing 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 SELECT * FROM Q.ST, 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.

Using Parameter Hints for queries and visual queries

If you are using CorVu Workstation Pro, 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 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 CorVu Objects folder. Select Visual Query from the list of CorVu Objects.
2. Specify the name of the query in the Name field and where the query will be saved in the Data Source field. Click Finish.
3. Click the SQL tab.
4. Type the first part of a parameterized function, which includes the name of the function and the
left parenthesis.
   For example, \texttt{SUM(}.
5. Press \texttt{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.

Creating visual queries using the Diagram Query editor

If you are using CorVu Workstation Pro, you can build simple to multi-variable queries visually using the \texttt{Diagram Query} editor.

When building queries using the \texttt{Diagram Query} editor, you supply table, join, column, sort, and row information and the \texttt{Diagram Query} editor constructs the SQL statements. The \texttt{Diagram Query} editor is not available in CorVu Web Pro.

To create a visual query using the \texttt{Diagram Query} editor, perform the following procedure:

1. Select \texttt{File \rightarrow New \rightarrow Other} and expand the \texttt{CorVu Objects}. Select \texttt{Visual Query} and click \texttt{Next}.

\begin{itemize}
\item \textbf{Note:} You can also click the New Visual Query (\texttt{ }) toolbar button.
\end{itemize}

\begin{itemize}
\item a. Specify the name of the visual query in the \texttt{Name} field and where the visual query will be
saved in the \texttt{Data Source} field.
\item b. Click \texttt{Finish}.
\item c. Click the \texttt{Diagram} tab
\end{itemize}

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 \texttt{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. You can also add a table to the query in one of the following ways:

\begin{itemize}
\item Click the \texttt{ER Diagrams} tab. Navigate to the ER diagram that contains the tables that you want
to query and double-click the table columns that you want to include in the query results. The
ER diagram displays relationships between different tables; columns as join lines between
table diagrams. This makes selecting columns for join conditions easier.
\item Drag tables from the \texttt{Workspaces} or \texttt{Repository Explorer} views 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.
\end{itemize}

4. To remove a table from the query, click the \texttt{Close} toolbar button, and the table is removed from
the editor window and the query.

5. 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.

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. 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.

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:
   ▪ 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.

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 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.

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, 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.

14. 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 toolbar 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 using the Tables window

You can use the Tables window to add a table to a query when you are building the query using the
Diagram Query editor. You can also use this window 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 Tables window:

1. Right-click anywhere in the editor and select Add Table from the context menu.

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

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 Add From List 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. Click **Add**. 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.

### Specifying substitution variables

Substitution variables are used to input changing values to a SQL query at runtime.

This feature enables you to substitute a part of an SQL statement and make it more generic. Substitution variables are active only while the query 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 visual 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 \texttt{SELECT} and \texttt{FROM} clauses. See what results your queries return.

**Setting fonts for SQL statements**

If you are using CorVu Workstation Pro, you can set unique font attributes for the SQL statement text. To set unique font attributes:

1. Display SQL statements in the workstation SQL query editor.
2. Select \texttt{Query $\rightarrow$ Set Font}
   
   The Font window opens.
3. Select the font for the SQL statements from the list of available fonts in the \texttt{Font} field.
4. In the \texttt{Font style} field, specify the style of the font.
5. In the \texttt{Size} field specify the size of the font.
6. Click \texttt{OK} to set the new font settings that you have specified. The Font window closes.

**Query menu**

The \texttt{Query} menu becomes available when the active object in the editor is a 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.

- **Manage Prompts**
  
  Opens the Manage Prompts window, where you can add, delete, and edit both simple and complex prompts.

- **View Prompt Values**
  
  Opens the Prompt Values window, where you can view the query prompt values that you have specified the last time you ran this query.

- **Transfer To**
  
  This command allows you to send the open query to one of several different output formats. Available formats are:
  
  - Quick Report
  - Forecast
  - Visual Report
  - Drill-down Path

- **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**
Chapter 3: Creating new objects or repository items

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.

Resize Column on Each Run
Select this menu item to calculate the necessary column width depending on the value of each cell automatically. With the help of this option the query results are displayed correctly in the Results tab or when they are exported to PDF file. The column width calculation takes place every time you run the query, when this option is selected.

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.

Previous Sub-Tab
Opens the previous Sub-Tab of the Query.

Next Sub-Tab
Opens the next Sub-Tab of 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 CorVu Workstation Pro and CorVu Web Pro 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. From 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 on 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 on 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 on 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 on 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 in 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 on 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 on 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 on 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 U2 queries

A U2 query is a request for information from a U2 data source. A U2 data source is a data source that represents a U2 database (UniVerse or UniData) in itself. The U2 query requests information from a U2 data source by UniVerse SQL or UniData SQL.

You can create a U2 query for retrieving data from a U2 file or table. Query results can be retrieved and manipulated based on the relationships established by the query’s association.

Creating U2 Queries with the Query Diagram editor

You can create a new U2 query using the Query Diagram editor.

To create a new U2 query using the Diagram editor:

1. Select File → New → Other → U2 Query and click Next. The Add New U2 Query dialog box is displayed.
   a. Specify the name of the query in the Query Name field and select the data source containing the file or table to be queried from the Data Source drop-down list.
   b. Optional: Click Advanced to edit the JDBC connection string keywords. Typically, changing the JDBC connection string keywords is reserved for scenarios where either the row length or the column length exceeds the buffer size.
      To increase the size of the buffer:
      1. Select the check box for the buffer that you want to increase (MAXFETCHBUFFS or MAXFETCHCOLS)
      2. Enter a new value in the Value field.
   c. Click the refresh button to the right of the File drop-down list and select the required file.
   d. Click Finish to go to the Diagram view.
      The Diagram Query editor window is divided into two sections. The top section contains all the columns or fields of the selected file.
      From the bottom section, specify the columns, sort conditions, and row conditions for the query results. See Working with query results for more information about query results.

2. Double-click the field that you want to include in the query.
   The field that you selected is highlighted. To deselect a field, double-click it again.

3. To add a table, perform the following steps:
   a. 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 U2 Query Diagram window.
   b. Select a table to include in the U2 query.

4. To remove a table from the query, click the Close button, and the table is removed from the editor window and the query.
To re-add a table to the editor area, right-click in the vacant window and select the *Add Table* option. Alternatively, you can right-click in the top section of the Diagram Editor window while the table is present, and select *Change Table*.

5. To edit a table, perform the following steps:
   a. Right-click in the top section of the Diagram window and select *Change Table* from the context menu.
      The Tables window opens where you can select a table to include in the U2 query.
      A visual representation of each table that you add appears in the upper portion of the U2 Query Diagram editor window.
   b. You can modify the U2 query by adding or removing columns. Right-click the column header and select *Add* or *Remove*.

6. The U2 Query Diagram editor contains the line that represents a field (column) on which the query UNNEST takes place. This line is completed automatically, depending on the selected fields, and contains its association name. In this editor only the fields with one and the same association or without any association at all can be selected.

In the *Diagram editor* the fields (columns) are enumerated with its associations if they have them.

The icons near the field (column) names show whether it is *singlevalued*, *multivalued* or *multisubvalued*.

### Creating U2 Queries with the SQL editor

For those with SQL experience, one way of creating a U2 query is to type their own SQL statements in the *SQL Query* editor.

To create a U2 query using the SQL editor, perform the following procedure:

1. Click the *New Query* icon on the toolbar and select the *Editor* tab.
2. Type the text of the SQL statement in the input text area.
   Make sure that SQL statements that you type are syntactically correct.
3. Click the *Run Query* icon.
   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*.

### Opening an existing U2 file

You can create a new U2 query by opening an existing file.

To create a new U2 query from an existing file:

1. In your workspace, find the U2 data source that contains the file that you want to query and expand the *Files* folder.
2. Double-click the file that you want to query.
   The file opens as a query in the *Diagram* query editor.
Retrieving data from a U2 query

You display specific query results data by selecting fields from existing associations.

You can specify which fields that you want to display by selecting specific fields in the Diagram Query editor. To obtain query results from a U2 query:

1. In your workspace, find the U2 data source that contains the file that you want to query and expand the Files folder.
2. Double-click the file that you want to query.
The file opens as a query in the Diagram Query editor.
3. You can select individual columns to be included in the query results in one of the following ways:
   ▪ Drag a field in the upper part of the Diagram Query editor window to a column in the query results column table.
   ▪ Right-click a field in the upper part of the Diagram Query editor window. A context menu opens. Select Add.
   ▪ Double-click a field in the upper part of the Diagram Query editor window.
The column name is added. Repeat this process for each field that you want to include 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 field 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.

   **Note:** You can only select fields that are included in the same association. If you try to select fields from different associations, an error message opens. There are three different types of fields that you can select from, and each type has certain restrictions on what types of fields it can be included with in the query results:
   ▪ A single-value field is displayed with a single column icon and can be included with any other field.
   ▪ A multi-value field that is not associated with other multi-value fields is displayed with a multiple column icon, and can only be included with single-value fields.
   ▪ A multi-value field that is associated with other multi-value fields is displayed with a multiple column icon, and the name of the association in parentheses. This type of field can be included with any of the other fields that have the same association next to their names, or any single-value field.

4. For each column that you include in the column table, you can apply sort and row conditions using the column table fields. Double-click in 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
Updating Rocket U2 database server programs

You can update programs installed on a U2 database server right from CorVu Workstation Pro.

When you access a U2 database by viewing objects, running queries, creating, or editing data sources you are prompted to update programs on a database server if they are older than those included with the application.

The automatic update check option must be enabled in CorVu Workstation Pro preferences. To enable this option, open the UniVerse/Unidata page in the Preferences window and then select the Automatically check for updates to server programs check box.

You can manually check for updates to server programs when you crate or edit a U2 data source by clicking Check XTOOLSUB.

To update U2 database server programs:
Chapter 3: Creating new objects or repository items

1. If a U2 database server program requires an update, the Update Server Programs dialog opens. In this dialog, select programs to update, and click **OK** to start update process. If you want to postpone the update click **Cancel**. If you do not want CorVu to check versions of programs installed on U2 database server, select the **Disable automatic check for updates to server programs** check box.

2. When updates are installed, click **OK** in the Progress window.

3. If an error occurs during the update process, the Server Program Update Error dialog opens. For more information on the error, click **Server update failure details**.

4. In case of an error, you can try to use another account to reconnect to the server and restart update process. For this, specify different authentication parameters and the account path on the server in the **Credentials for Server Reconnect** and **Account for Server Reconnect** areas. Click **OK**.

### Working with simple prompts

You use simple prompts to limit the data retrieved from a query. Simple prompts are closely related to a query and can be saved within it.

By determining simple prompt values you can filter query results and display only those that are based on the set values. For example, 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 input 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 input 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 prompts to a query to determine the data 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 **Visual Query** editor switch to the **Design** page.

   **Note:** Use the **Prompts** pane on the **Design** page to manage prompts. The table on the **Prompts** pane lists the columns of the query and prompts assigned to these columns.

2. To create a simple prompt, click **Add Prompt** on the **Prompts** pane toolbar.

3. Specify the name of the prompt in the **Prompt name** field of the Simple Prompt window.

4. In the **Display string** field, type the prompt text that will be displayed when you run the query.

5. Select the prompt variable type from the **Input type** list. The following options are supported:
   - Literal
   - Date
   - Time
   - Timestamp
Editing simple prompts

You can use prompts to determine the data retrieved in a query so that only the specified value or ranges are used.

Use the Simple Prompt window to edit a prompt that was added to a query.

To edit a simple prompt to a query:

1. In the Visual Query editor switch to the Design page.

   **Note:** Use the Prompts pane on the Design page to manage prompts. The table on the Prompts pane lists the columns of the query and prompts assigned to these columns.

2. Open the Manage Prompts window by clicking Manage Prompts on the Prompts pane toolbar.
3. Select a prompt that you want to edit from the list of available prompts on the Prompts page, and then click Edit Prompt.
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 will be displayed when you run the query.
6. Select the prompt variable type from the Input type list. The following options are supported:
   - Literal
   - Date
   - Time
   - Timestamp
   - As is
   - Enumeration
7. If you selected Time or Timestamp 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.
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.

---

6. If you selected Time or Timestamp 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.
7. 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 will be 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 being displayed.

8. If you selected Time or Timestamp as the type of the prompt variable, you can set the value in the Date and Time window. To open the window click Choose default date.
9. Assign the prompt to a query column by selecting one of available columns from the Assigned column list.
10. Click OK to close the Simple Prompt window. New prompt is added to the table on the Prompts pane.
**Chapter 3: Creating new objects or repository items**

---

**Note:** If the **Has default value** check box is cleared, you will be 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 being displayed.

9. If you selected *Time* or *Timestamp* as the type of the prompt variable, you can set the value in the Date and Time window. To open the window click **Choose default date**.
10. Click **OK** to close the Simple Prompt window. New prompt is added to the table on the **Prompts** pane.

---

**Saving prompt values with 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 prompt 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 the end user to select one or more values from a flat or hierarchical list of options.

Prompt hierarchies are distinct CorVu 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. End 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 using the Create New Prompt Hierarchy wizard.

To create a new prompt hierarchy:

1. Select **File ➔ New ➔ Other**. The **New** window opens.
2. Expand the **CorVu Objects** folder, select **Prompt Hierarchy**, and click **Next**. The Create New Prompt Hierarchy wizard opens.
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 is divided into the following three sections:

- **Source of data** where you will specify where the prompt hierarchy will draw its data from.
- **Hierarchy levels** where you will construct the prompt hierarchy levels and their properties.
- **Prompt hierarchy preview** where you will preview the hierarchy for the prompt hierarchy results.

4. 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.
5. For linked data source, click the ellipsis (...) at the end of the **Path** field. The Open wizard opens.
6. Select **Open from Repository**, and click **Next**. The Open from Repository wizard opens.
7. Drill down through the relational data sources to select the desired 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.
8. 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** - 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.
9. From the **Prompt hierarchy preview** section of the Prompt Hierarchy editor window, use the **Refresh** icon to preview the prompt hierarchy.

10. 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.

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

12. 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.
Analytical queries and associated query types

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.

To create an analytical query:

1. Select **File → New → Other**. The New window opens.
2. Expand the **CorVu 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 to the analytical query structure, select the query under which you want to add the new query, click **Add**, and then select the type of query that you want to add. The new query is displayed under the selected query.

**Note:**
The following objects can be used as queries:

- Charts
- Queries
- Analytical Queries
- Dynamarts
- Compound Dynamarts
- Forecasts
- Data files (CSV, DBF, IXF, TXT, and XML)

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.

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.

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.

**Related topics**
- Importing query results from data files, on page 104

### 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** 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.
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 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** 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.
Adding a normalize query

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>

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 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 grouping 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.
8. In the Conditional Grouping section, click Add Filter.
   The Filter Wizard opens.
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 results are displayed in the Results editor.

You will notice that there is a column for every group and value column that you selected. The value column headings display the column names that you specified. The contents of the value columns represent all of the values that are included in the conditional filters that you set and are aggregated according to the methods that you selected.

For more information about building filters, see:

▪ Building a compound condition filter, on page 113
▪ Building a free style condition filter, on page 114
Adding a column filter query

You can use column filter queries to add calculated columns and filters to the source query without modifying the source query itself. You can also choose to exclude columns from the result query.

For example, let's say you have a query that displays staff data for your company and that this query uses a calculated column to enumerate the rows. When you apply a filter to this query, some rows are excluded from the result set. The remaining rows (the rows not excluded) maintain their original position, meaning their position is preserved and is not recalculated based on the rows excluded. You can use a column filter query to add another calculated column that displays the row numbers of the result query and not the source.

To add a column filter query:

1. Create an analytical query and display it in the Structure query editor.
2. Right-click the Analytical query structure pane, and then select Add Column Filter.
3. Right-click the <none> node, and then select Add Query.
4. In the Open window navigate to and select the source query that you want to use. The source query’s columns are displayed in the Result Columns list.
5. Specify the query result columns that you want to be displayed by clicking Move to the Left or Move to the Right. You can also click Move All to the Left and Move All to the Right to move all the columns simultaneously between the Source Columns and the Result Columns lists.
6. Click the Design tab. On the Design page you can manage prompts, calculated columns, and filters.
7. In the Prompts pane, you can add and edit prompts.
8. Use the Calculated Column window to add calculated columns. Click in the Dynamart pane, and then select Add Calculated Column.
9. Use the arrow buttons in the Dynamart pane to change the order of query columns. This column order is applied only when the query is used as a data source in the other application parts like dashboards or analytical queries.
10. Use the Dynamart Filter wizard to create conditional expressions that control the contents of query results. In the Transformations pane right-click the Filter node and select Edit Filter.

Note: When you use the Transformations pane controls the filtering is applied in the application, it does not change SQL text executed in the database.

The following table shows a query with calculated column which returns row numbers:

<table>
<thead>
<tr>
<th>Deptname</th>
<th>Deptnumb</th>
<th>Division</th>
<th>Location</th>
<th>Manager</th>
<th>Row</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEAD OFFICE</td>
<td>10</td>
<td>CORPORATE</td>
<td>NEW YORK</td>
<td>160</td>
<td>0</td>
</tr>
<tr>
<td>NEW ENGLAND</td>
<td>15</td>
<td>EASTERN</td>
<td>BOSTON</td>
<td>50</td>
<td>1</td>
</tr>
<tr>
<td>MID ATLANTIC</td>
<td>20</td>
<td>EASTERN</td>
<td>WASHINGTON</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>SOUTH ATLANTIC</td>
<td>38</td>
<td>EASTERN</td>
<td>ATLANTA</td>
<td>30</td>
<td>3</td>
</tr>
<tr>
<td>GREAT LAKE</td>
<td>43</td>
<td>MIDWEST</td>
<td>CHICAGO</td>
<td>100</td>
<td>4</td>
</tr>
<tr>
<td>PLAINS</td>
<td>51</td>
<td>MIDWEST</td>
<td>DALLAS</td>
<td>140</td>
<td>5</td>
</tr>
</tbody>
</table>

If you apply a filter to this query, some rows will be dropped from the result set. But the calculated column will return the same row numbers as in the source query, as shown in the following table:
Chapter 3: Creating new objects or repository items

You can use column filter query to add another calculated column that will show row numbers of the result query, as shown in the following table:

<table>
<thead>
<tr>
<th>Deptname</th>
<th>Deptnumb</th>
<th>Division</th>
<th>Location</th>
<th>Manager</th>
<th>Row</th>
<th>Row2</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEAD OFFICE</td>
<td>10</td>
<td>CORPORATE</td>
<td>NEW YORK</td>
<td>160</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>GREAT LAKE</td>
<td>43</td>
<td>MIDWEST</td>
<td>CHICAGO</td>
<td>100</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>PLAINS</td>
<td>51</td>
<td>MIDWEST</td>
<td>DALLAS</td>
<td>140</td>
<td>5</td>
<td>2</td>
</tr>
</tbody>
</table>

For more information about adding calculated columns and building filters, see:
- Adding calculated columns to the query results
- Filtering query results

**Selecting a new data source**

You use the Set Data Source window to select a data source against which a query object will be run.

All query objects are associated with a default data source. When query objects are run, they are automatically run against the default data source. The default data source for a query object is the last used data source before the query was saved or in the case of new queries that have not yet been saved, the default is the data source last used in the CorVu session.

To select the data source where you want to run the current query object:

1. With a query in the active editor, select Query → Set Data Source
   The Set Data Source window opens.
2. Select the data source where you want to run the query object from the Data source list. The Data source list includes all the data sources that are available for your use based on your user ID.
3. In the User name field, type the user ID that you want CorVu to use to connect to the new data source.
4. In the Password field, specify the password associated with the user ID that you typed in the User name field.
5. To save the password that you have specified for this user ID and use it every time to connect to this data source, select the Remember the password check box.
6. Click OK. The object is run against the data source that you have specified in the Data Source field.

**Setting fonts for SQL statements**

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

To set unique font attributes:

1. Display SQL statements in the workstation SQL query editor.
2. Select Query → Set Font
   The Font window opens.
3. Select the font for the SQL statements 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.

**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 query is running. As a result, only one object can access the substitution variable. The variable will not exist after the object is executed.

A substitution variable is special text in a query that begins with an ampersand character (&). A substitution variable can contain up to 18 alphabetic, numeric or special characters.

A substitution variable can appear anywhere in a query. The value of the substitution variable can be anything used in a query (except a comment). For example, you can use a substitution variable in place of a column name, search condition, subquery, or any specific value.

In the following example, you will be prompted for a customer number each time you run the following query:

```
SELECT ORDERNO, SALESREPNO, PRODNO, QUANTITY, &CUSTNO AS CUSTOMER#
FROM Q.SALES
```

When you run the query and supply customer number at the prompt, the query will retrieve only those records that are associated with the specified customer number. Later you can launch the query and provide a different customer instead of writing a separate query.

To use a substitution variable:

1. Open a visual 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.

**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.
Chapter 3: Creating new objects or repository items

Matching patterns

When entering query 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 _

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 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 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.

**Profiling queries**

You use profiling to analyze the complexity of the queries.
You must have a query to analyze open.

You can profile the following query types containing only SELECT statements:

- visual queries
- analytical queries
- U2 queries

To profile a query:

1. Open the required query in the Query Editor.
2. Click the **Profile Query** toolbar button.
   The Query Profiler view opens. It displays the time spent on running the query and fetching its results if there are any.
3. Click the **SQL Text** toolbar button in the Query Profiler view to see the SQL text of the last profiled query.
   The SQL text is displayed in the Query Profiler window.

**Saving objects**

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

The save commands are **Save**, **Save As** and **Save All**. The status of the current active object determines which save commands are available. The active object status could 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:
Chapter 3: Creating new objects or repository items

- 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 As** command will save the current opened object to a particular directory or repository folder with a particular name. 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 queries that you create in a repository or in a particular repository workspace using the Save at Repository window.

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 that you create to a file.

To save an object to a file:

1. If you are using CorVu Workstation Pro:
   1. Select **Save At → File** from the **File** menu.
   2. Navigate to the directory where you will save the object.
   3. Specify the name of the object in the **File name** field.
   4. Click **Save**. The object is saved to the specified location.
2. If you are using CorVu Web Pro:
   1. Select **Save At → File** from the **File** menu.
   2. Specify the name of the object in the **Input file name** field.
   3. Click the **Link to file** hyperlink. A browser specific download dialog opens.
   4. Download the object to the desired location on your local machine.
Chapter 4: Working with query results

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

You can use different editors to work with query results.

**SQL editor**

Use it to write your own SQL statements or edit them. This editor supports syntax highlighting so that you can clearly see certain types of statements and relationships. The content-assist function displays a list of SQL keywords or object names that are appropriate for the current cursor position in the SQL statement. You can also display parameter hints for the SQL functions that you use in a query if you do not know the full syntax or use of the function. These SQL statements are issued to the database.

**Diagram editor**

Use it to wire the tables together visually and create visual or analytical queries. You can use drag-and-drop mechanism to add tables.

**Design editor**

This editor consists of the **Prompts**, **Dynamart** and **Transformation** panes.

- Use the **Prompts** pane to manage prompts and filter query results with the help of them. This kind of filtering works on the database level and adds filter statements to SQL instantly before sending it to the database when query is run.
- Use the **Dynamart** pane to specify the columns and their order in a query when this query is used as a separate data source for Visual Reports, Visual Dashboards, Visual Applications and other CorVu objects.
- Use **Transformations** pane to control the filtering which is applied in the application; it does not change SQL text executed in database.

**Display editor**

Use it to set up the representation of results. You can create charts or grids and choose default.

**Results editor**

Use the pop-up menu of this editor for grids:

- To format the columns and rows of the query results
- To aggregate columns or rows
- To add calculated columns to the query results
- To modify the order of the columns with the help of drag-and-drop action

Use the toolbar buttons and file menu:

- To save the query results to a file or a repository
- To export the results to a database or a file
- To transfer the query to a Drill-down Path, Forecast, Quick Report or Visual Report
- To switch between the representations of the results: result grid or charts

**Working with calculated columns**

Calculated columns are columns of data that you can add to the query result set.

Having access to the appropriate syntax reference information and function reference information before using the calculated column feature will help you to build expressions more quickly.

For information about syntax, see **Syntax for calculated column expressions** in the reference section of the User help.
For a list of all available functions, including the functions for calculated columns, see Functions for calculated columns in the reference section of the User help.

Adding calculated columns to a query result set

You use the Calculated Columns window to add a calculated column to the query result set. You can add a calculated column before or after existing columns.

The content of an added column is calculated by using an expression that you define. The expression can use data from other columns and constants. The expression used to calculate the value of the added column is saved with the query. The calculated column will be included in the query result set each time the query is run.

To add a calculated column to the query result set, perform the following steps:

1. Open the Calculated Column window in one of the following ways:
   - If you are using CorVu Workstation Pro, on the Results page of the editor, right-click a column’s header in the query results and select either Add Calculated Column Before . . . or Add Calculated Column After . . . from the context menu.
   - With query results in the active editor, select Results → Add Calculated Column.

2. In the Calculated Column window, specify the name of the calculated column as it should appear in the query results in the Name field.

3. Select the type of the calculated column from the Type drop-down menu. The type determines the format of the calculated column data. Valid choices are:
   - **Integer**: Data is expressed as whole numbers. For example, "5" or "100".
   - **Decimal**: Data is expressed as a real number. For example, "10.5" or "0.25".
   - **Character**: Data is expressed only as specific text strings. For example, "John Smith", or "10 Main Street".
   - **Date**: Data is expressed in a date format. For example, "12/03/2000" or "1/2/09".
   - **Boolean**: Data is expressed as a Boolean value. For example, "true or false", or "yes or no".
   - **Categorization**: Data is represented by category names that are defined for a range of data values. For example, "small, average and large" can be used to categorize sales quantities that range between 1,000 and 100,000 units.

4. In the area containing all available functions, navigate to and select the required one.

   **Note:** You can use the filter to quickly find the function you need.

5. Build the expression

   In the Expression: window, enter the syntax for the expression to be used to calculate the contents of the column.

   The expression that you build can use data from other columns, constants and functions. You must adhere to a specific set of syntax rules when building the expression for the calculated column.

   The expression language supports the following rules:
   - constant expressions defined as variables sourced on other columns (column 1, column 2, for example) in the result set
   - numeric operators (such as +, -, *, /), character operators (+, CONCAT)
   - numeric and character functions (such as MIN, MAX, AVG, SQRT, CONCAT, SUBSTR)
Adding user-defined functions

- 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.

6. Click **OK** to close the Calculated Column window.

The content of the column is calculated according to the specified expression and added to the query result set.

- If you selected **Add calculated column** from the context menu, the new column is added to the as the last column in the query results.
- If you selected **Add calculated column before** from the context menu, the new column is inserted in the query results before the selected column.
- If you selected **Add calculated column after** from the context menu, the new column is inserted in the query results after the selected column.

Adding user-defined functions

You can add JavaScript functions to CorVu Workstation Pro and CorVu Web Pro and use them for building different expressions, for example, when adding a calculated column or creating a dynamart filter.

To create and add a new user-defined function:

1. Create a file called `functions.js` with the description of JavaScript functions and their implementation.
2. Start each function with the keyword `function`. For details on syntax, see **Syntax for user-defined functions**.
3. Add a comment before each function.
   The application interprets this comment as the description of the function.
4. **Optional:** Define a category to which you add a new function.
   Use the `@category` tag in the comments to add it. If there is no such category, it will be created.

**Note:** The following categories already exist in the application:

- Columns
- Aggregation
- Conversion
- Date and Time
- Information
- Logical
- Math and Trigonometric
- Misc
- Text
- Objects

5. Place the `functions.js` file in the CorVu Workstation Pro or CorVu Web Pro user home directory.
   If you want to share `functions.js` with other users, they will have to copy the `functions.js` file to their home directories.

6. Launch or restart your application.
The function is added to the defined category and can be used for building expressions for calculated columns, dynamart filters, and scheduler.

This allows you to implement various features in other CorVu objects such as visual dashboards and application. For example, you can display data retrieved from a table on a map according to address information from that table. For more information, see Displaying data on a map according to address information retrieved from a query.

Related topics
Syntax for user-defined functions

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 active editor, select Results → Grouping and Aggregation.
   - If you are using CorVu Workstation Pro, on the Results page of the editor, right-click a column’s header in the query results and select Grouping and Aggregation from the context menu.
2. Select the grouping and aggregation option that you want to apply to the column.
3. Repeat the process for each column of data in the query results to which you want to add grouping and aggregation.

Grouping and aggregation fields

Grouping and aggregation options can be applied to query result columns to organize the result data into logical or summarized groupings.

No aggregation
Select this option to remove any aggregation or summary formatting from the selected column.

Top Group
Select this option to group data across the top of the query results by the selected column. Placing a column in a top group will display related values of that column on the horizontal axis, which is known as pivot or ACROSS functionality. You can define multiple columns of top groups. You can specify that Top Groups include summary information. With the column selected, click Show Total Summaries from the 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.

**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.

---

**Customizing how query results display**

You customize how query results are displayed in the editor window.

You can set formatting options for fonts, colors and text alignments for each column header and each cell of the column. You can also customize the column and cell formatting to be applied based on the results of a conditional expression.

To customize how query results are displayed:

1. Open the Layout Properties window in one of the following ways:
   - Right-click a column’s header and select **Font** or **Format** from the context menu.
   - Select **Results** → **Font**.
   - Select **Results** → **Format**.

The Layout Properties window uses a tree structure to represent the columns in the query results. 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.
You can apply formatting decisions to entire columns, individual cells, column headers, and summary cells. The options are divided into the following groups:

- General
- Font
- Format
- Conditional

2. 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.

3. 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 window changes. The formatting fields that can be used on the selected branch are made available.

4. 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 window. On this page you can specify a header name, column width, and summary row height.

5. 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 window.

6. 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.

7. 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 window. Click 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. Click the Format tab to specify how the data will be formatted in the detail cell of the column.

8. 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.

9. 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 window. Click 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. Click the Format tab to specify how the data will be formatted in the summary line cell of the column.

10. 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.

11. Click Apply. The General, Font, or Format options that you have specified are saved.

12. 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**: Click 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 click Apply to all columns, 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:** Click 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 click Apply to all levels, the font options that you have specified will be applied to the Grand Total Default branch for the column.

13. Click **OK**. The Layout Properties window 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 window 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. From the **Detail** branch, select the **Default** branch to specify the font attributes that will be used for all detail cells in the column. Select a **Condition** branch to specify the font attributes 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 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. From the **Grand Total** branch, select the **Default** branch, to specify the font attributes that will be used for the summary cell. 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 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 contents 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 on 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. Click the **Detail** branch to apply conditional formatting to the detail cells of the column. Click the **column name Total** branch to apply conditional formatting to the summary cell of the column. Click on 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. You must give each conditional expression that you create a unique name. Type the name of the expression in the **Condition name** field. A branch is automatically created in the tree using the condition name.

5. Each condition must have an expression. 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.

### 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 active 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 selecting one of the following options:
   - 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.

### 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. If you want to attach a comment to the data, type the comment in the **Comment** field.

5. 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 data in the current results grid will be saved with all the changes applied to it. For example, if you remove a column from the layout, it will not be saved in the export file.
   - **All - Result set only** to specify that all of the data in the original query result set not depending on the changes applied to results grid will be saved. For example, if you remove a column from the layout, it will be saved in the export file anyway.
   - **Selection** to specify that only the data that you have selected in the current query results will be saved. This can include calculated columns if they are selected. The query result data that you want to save must be selected before you open the Export Query Results window.

   **Note:** **All - Result set only** is the only available option when you export a query result set from a forecast.

8. 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.

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

10. 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.

11. 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,

```
SELECT 45.5 AS PRODPRICE, 30 AS AMOUNT, '=A2*B2' AS PRICE
FROM SYSIBM.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. If you are using CorVu Workstation Pro, specify a name for the file that you are creating in the **File name** field.
3. Specify the type of file that you are creating. From the **Export type** field, select one of the following file types:
   - **Text** (*.txt)
   - **IXF** (*.ixf)
   - **XLS** Microsoft Excel spreadsheet (*.xls)
   - **CSV** (*.csv)
   - **PDF** (*.pdf)
   - **dbase III** (*.dbf)
   - **HTML** (*.htm)
   - **XML** Extensible Markup Language file (*.xml)
4. 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.
5. Click **Finish**. If you are using CorVu Workstation Pro, a save window specific to your operating system opens. If you are using CorVu Web Pro, 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>▪ <strong>All</strong> to specify that all of the data in the current results grid will be saved with all the changes applied to it.</td>
</tr>
<tr>
<td></td>
<td>.csv</td>
<td>For example, if you remove a column from the layout, it will not be saved in the export file.</td>
</tr>
<tr>
<td></td>
<td>.ixf</td>
<td>▪ <strong>All - Result set only</strong> to specify that all of the data in the original query result set not depending on the changes applied to results grid will be saved.</td>
</tr>
<tr>
<td></td>
<td>.wqml</td>
<td>For example, if you remove a column from the layout, it will be saved in the export file anyway.</td>
</tr>
<tr>
<td></td>
<td>.xml</td>
<td>▪ <strong>Selection</strong> 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>.dbf</td>
<td><strong>Note:</strong> Calculated columns are saved when either <strong>All</strong> and <strong>All - Result set only</strong> are selected.</td>
</tr>
<tr>
<td></td>
<td>.pdf</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>
### Specifying file export options

<table>
<thead>
<tr>
<th>Field</th>
<th>File Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Date/Time Format</strong></td>
<td>.txt</td>
<td>Click 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 the time will be formatted as (HH.MM.SS).</td>
</tr>
<tr>
<td></td>
<td>.csv</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 the time will be formatted as (HH.MM 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 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 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><strong>CCSID</strong></td>
<td>.txt</td>
<td>Use this field to select the code page to use when saving this file.</td>
</tr>
<tr>
<td></td>
<td>.html</td>
<td></td>
</tr>
<tr>
<td></td>
<td>.csv</td>
<td></td>
</tr>
<tr>
<td></td>
<td>.ixf</td>
<td></td>
</tr>
<tr>
<td></td>
<td>.dbf</td>
<td></td>
</tr>
<tr>
<td><strong>Display null values</strong></td>
<td>.txt</td>
<td>Use this field to specify the string that will be used to display null values.</td>
</tr>
<tr>
<td></td>
<td>.csv</td>
<td></td>
</tr>
<tr>
<td><strong>Separate columns</strong></td>
<td>.txt</td>
<td>Use this field to specify the string that will be used to separate the columns.</td>
</tr>
<tr>
<td></td>
<td>.csv</td>
<td></td>
</tr>
<tr>
<td><strong>Delimit values</strong></td>
<td>.txt</td>
<td>Use this field to specify the string that will be used to delimit values.</td>
</tr>
<tr>
<td></td>
<td>.csv</td>
<td></td>
</tr>
<tr>
<td><strong>When should values be delimited</strong></td>
<td>.txt</td>
<td>Use this field to specify when values should be delimited. Your choices are:</td>
</tr>
<tr>
<td></td>
<td>.csv</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><strong>Which type of values should be delimited</strong></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></td>
<td>.csv</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>
</tbody>
</table>
### Chapter 4: Working with query results

#### Field | File Type | Description
--- | --- | ---
**Format** | .ixf | Click one of the following radio buttons to specify how to format the .ixf output file. Your choices are:
- PC/IXF (Version 1) to export data in PC/IXF Version 1 format.
- PC/IXF (Version 2) to export data in PC/IXF Version 2 format.
- System/370 IXF to export data in System/370 IXF format.

**Add Document Information** | .pdf | 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:
- **Title** specifies the title of the PDF document.
- **Author** specifies the author of the PDF document.
- **Subject** specifies the subject of the PDF document.
- **Keywords** Specifies the keywords for the PDF document. Keywords are used as search criteria.

**Use only ANSI Fonts** | .pdf | 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, check this check box to produce a smaller PDF in less time.

---

### Importing query results from data files

You can open CSV, DBF, IXF, TXT, and XML data files in CorVu. You use the Results page of the query editor to view and operate the data imported from these files.

You can manipulate the imported data like any query result set and perform different operations, including filtering, adding calculated columns, and saving the result set as a dynamart to a repository or file system.

The imported result sets can be used as subordinate queries in an analytical query and can be added to visual projects.

#### Related topics
- Opening data files, on page 107
- Adding calculated columns to a query result set, on page 92
- Applying grouping and aggregation to query result columns, on page 94
- Filtering query results, on page 112
- Customizing how query results display, on page 95
- Converting data files to dynamarts, on page 107
- Analytical queries and associated query types, on page 77
- Specifying queries for visual projects, on page 260
Importing query results from CSV and text files

You use the CSV File Import wizard to import CSV and text files into the repository.

CSV and TXT files are imported into the active editor where they can be saved as result sets to the repository. 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 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.
The content of the import file displays in the Preview text box.
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 CorVu 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 CorVu 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. Specify 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 the Separated by and then select 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. Specify 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 might 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.
Opening data files

You can open CSV, DBF, IXF, TXT, and XML data files in CorVu.

To open a data file:

1. Select **File → Open → From File**.
2. In the Open window, specify the name of the file and its location in file system.
3. Click **Open**. The result set opens on the **Results** page of the editor window.

Related topics

- **Adding calculated columns to a query result set**, on page 92
- **Applying grouping and aggregation to query result columns**, on page 94
- **Filtering query results**, on page 112
- **Customizing how query results display**, on page 95
- **Converting data files to dynamarts**, on page 107

Converting data files to dynamarts

You can save previously opened and modified data files (CSV, DBF, IXF, TXT, and XML) as dynamarts to a repository or file system.

When you save data files as dynamarts, they preserve all the modifications you apply to the result set, including added calculated columns and filters. When you open a dynamart, it reloads the data from the source file and updates its result set if the source file was modified or replaced.

To convert data files to dynamart:

1. Make sure that the data file you want to save as a dynamart is opened in the active window.
2. Select **File → Save** to open the Save window.
3. In the Save window, select one of the following wizards depending on where you want to save the dynamart:
   - Save to Repository
   - Save to File
4. In the Save to Repository window, specify the parent entry, file name and comment.
5. In the Save to File window, specify the directory where the file should be saved and the file name. You can click **Browse** to search the destination directory.
6. Select Dynamart from the **Type** drop-down list.
7. Click **Finish** to save the dynamart files.
Managing source data files in dynamarts

After saving the dynamart, you can change or completely remove the link to the source data object.

When you save data files as dynamarts, the link to this data file and the time of its last modification are saved within the dynamart. When you open this dynamart, CorVu reloads the data from the linked source file and updates the result set if the source file was modified or replaced.

You can change the source object or disable data update for a particular dynamart in the Dynamart Settings window.

To replace the source data file or disable data update:

1. Select **Results → Dynamart Settings**.
2. In the Dynamart Settings window, specify the name and location of a data file that you want to use as source file in the **Data file** field. You can click **Browse** to search the file.
3. Select the **Update dynamart when file is changed** check box if you want the result set to be updated on any changes in the linked data file.
4. Clear the **Update dynamart when file is changed** check box to unlink the source data file. In this case the dynamart always displays original data even if the data file is replaced or updated.
5. Click **OK** to close the Dynamart Settings window.

Related topics
- Setting root directory location for dynamart source files

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 CorVu administrator.

To retrieve or save LOB data:

1. Run a query 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.
   - 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. Select 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.

1. 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.

2. In 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.

3. 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 in 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 CorVu 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. Select 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 the contents of a cell in the query results**

If you are using CorVu Workstation Pro, you can use the **Zoom** option to view in hexadecimal or binary the contents of a cell in the query results.

To view the contents of a query results cell:

1. Right-click a cell in the query results. Select **Zoom** from the context menu. The Zoom window opens.
2. The **CCSID** field specifies the code page that was used to save the data in the selected query results cell.
3. The current value of the selected query results cell is displayed in the **Cell value** field.
4. Click Advanced to expand the window with additional fields to display the data in hexadecimal or binary format. This button is disabled for cells with numeric data.

5. 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.

6. Select Cancel to close the Zoom window.

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 in the query editor tool bar. 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 display the first value in the column.
▪ Select Last to display 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 Average to create a summary row showing the average of the values in the column.
▪ Select Standard Deviation to display the standard deviation of the values in the column.
- Select **Formula** to use the formula aggregation.

**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 Column window where you can specify the options for a new calculated column in the query result set.

**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 Excel Sheet**
Opens Microsoft Excel and displays the query results data in an Excel data sheet. (CorVu Workstation Pro only)

---

**Note:** For relational query results whose columns have both breaks and usage codes, an Excel PivotTable will be created automatically. Processing time for this functionality increases depending on the amount of data.

**Retrieve All LOB Values**
Retrieves all LOB values in the query results set.

**Export**
Opens the Export Query Results wizard where you can choose to save the query results to a file or a database.
Display Chart
- Allows you to display the query results in a specific chart type. Available chart types include:
  - Column Chart
  - Bar Chart
  - Line Chart
  - Area Chart
  - Pie Chart
  - Scatter Chart
  - Bubble Chart
  - Speedo Chart
  - Grid Chart

Filter Results
- Opens the Dynamart Filter Wizard where you can create new filters for the open query results.

Sorting query results

Use the Sort window 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 active window, open the Sort dialog 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.

The specified sorting order will be saved within the query and used the next time you run it if the GQW_QUERY_PRESERVE_SORT global variable is set preserve sorting order (1). If you need to rerun the query without applying the specified sorting order, set the GQW_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.

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 filter can contain any number of expressions that give you complete control over the content of the grid. These filters are applied in the application, they do not change SQL text executed in the database. To filter query results:

1. Open the query result set that you want to filter in the Results editor.
Building a compound condition filter

A compound condition filter allows you to filter query results using multiple conditions. Use a compound condition filter to set multiple conditions to filter your query results. You can add more than one condition from a table of options in the Dynamart Filter Wizard. To build a compound condition filter:

1. Open the Dynamart Filter Wizard, select the Enable filtering check box, select Compound condition filter and click Next. The Edit dynamart filter page opens.
2. Select the column that you want to filter from the Column drop-down list.
3. Select the operator of the filter from the Operator drop-down list. Valid options include:
   - Not Less Than - All values that are equal to or greater than the specified value are included.
   - Greater Than - All values that are greater than the specified value are included.
   - Not Greater Than - All values that are equal to or less than the specified value are included.
   - Less Than - All values that are less than the specified value are included.
   - Equal To - Only values that are equal to the specified value are included.
   - Not Equal To - All values that are not equal to the specified value are included.
   - Like - All values that match the specified pattern are included. The pattern can contain the following special characters:
     - % matches zero (0) or more characters.
     - _ matches only one (1) character.
     - ? matches only one (1) character.
   - 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.
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.
Chapter 4: Working with query results

- **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 CorVu 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 to use to filter the results in the **Expression** field.
3. Use the function categories on the right to help you construct your filter expressions.
   
   **Note:** You can use you own functions to construct the filter expression. For more information, see *Adding user-defined functions*.

4. Once you have finished building you filter expressions, click **Finish**.
   The Dynamart Filter Wizard closes and the query results are filtered.
Chapter 5: 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 Charts

You can create a chart with the help of the New chart wizard. You can use variety of fully customizable charts, or display modes 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. The fastest way to open the New chart wizard is through the query results grid.

With the help of the New chart wizard you can also add charts to the Header and Footer of your quick reports.
Creating Charts

You can create a chart with the help of the New chart wizard. You can use variety of fully customizable charts, or display modes 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. The fastest way to open the New chart wizard is through the query results grid.

With the help of the New chart wizard you can also add charts to the Header and Footer of your quick reports.

Specifying chart type options

You can specify how to display a chart on the Chart type page of the New chart wizard. Chart Type options determine the type of the chart and the visual format of the display mode.

To specify chart type options:

1. From the Chart type list, select the required display mode for the chart. Available options are the following:
   - **Column** – Data points are represented by vertical columns.
   - **Bar** – Data points are represented by horizontal columns.
   - **Line** – Data points are plotted on the X-Y axes using interconnected lines.
   - **Area** – Data points are plotted similar to the line chart but with the area between the data series and x-axis filled with a specified color.
   - **Pie** – Data points are displayed as wedges of a pie.
   - **Scatter** – Data points are plotted as discrete points, using a user-configurable data symbol.
   - **Bubble** – Data points are plotted using spherical data symbols, each of which can be sized according to an expression or column value.
   - **Speedo** – Data points are displayed as points on a dial.

   You can also select Grid mode to display data point in tabular grid.

2. From the Dimension list select the required dimension. Available variants are the following:
   - **2D** – The chart is displayed as a flat, two dimensional objects. It is available for all chart types.
   - **2D With Depth** – The chart is displayed as a two dimensional object, with data points displayed in three dimensions. It is available for Column, Bar, Area, and Pie chart types.

   **Note:** Stacked Area Charts do not support **2D With Depth** dimension.

3. If you select Column, Bar, or Line chart type, select the required subtype. Available options are the following:
   - **Side-by-Side**
   - **Stacked**
   - **Percent Stacked**
Specifying data series options

You can specify which columns of a result set are used to generate data points of a chart on the Data Source page of the New chart wizard.

To specify the data series options of a chart:

1. To specify the column that will populate the category axis, select a column from the Result set tree and click Add to category. The column is added to the Category Axis node of the Series tree.
2. To specify a column that will be displayed as a value, select a column from the Result set tree and click Add to value. The column is added to the Values node of the Series tree.
3. Select aggregation type for value column from the Aggregation drop-down list. Valid options are:
   - No aggregation
   - Count
   - First
   - Last
   - Maximum
   - Minimum
   - Sum
   - Average
   - Standard Deviation
4. Specify the way of aggregating a value column by selecting or clearing the Calculate as formula check box. The Calculate as formula check box is available only if you use a calculated column as a value. The expression you have specified for the calculated column is used as aggregation formula.
   - If you select the Calculate as formula check box, the fields of the columns that provide data for calculated column (and thus for value column) are aggregated at first. The formula uses already aggregated field values as arguments to calculate the result value.
   - If you clear the Calculate as formula check box, the defined formula calculates the value column fields separately. After the calculations the fields are aggregated to calculate the result value.
5. To remove a column from the series tree, select the column and click Remove. The column is added back to the Result set tree.

Specifying format options

Specify Chart Format options to determine the visual formatting of the chart object.

On the Format page of the New chart wizard you can specify Series options and Chart Area options.

Series options determine which category axis and value axis series are displayed in the chart, and how they are formatted.

Chart Area options determine the visual formatting of the chart object, such as the chart title, legend, and chart area color schemes.

Specifying Series chart format options

Series options determine which category axis and value axis series are displayed in the chart, and how they are formatted.

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 topics**

- Specifying Labels chart format options, on page 121

**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.
Chapter 5: Working with charts

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:

- YYYYxMMxDD
- MMxDDxYYYY
- DDxMMxYYYY
- YYxMMxDD
- MMxDDxYY
- DDxDDxYY

The following formats are available for time:

- HHxMMxSS
- HHxMM

The following formats are available for date:

- YYYYxMMxDD HHxMMxSS
Chapter 5: Working with charts

- DDxMMxYYYY HHxMMxSS
- YYYYxMMxDD HHxMM
- DDxMMxYYYY HHxMM
- YYxMMxDD HHxMMxSS
- DDxMMxYY HHxMMxSS
- YYxMMxDD HHxMM
- DDxMMxYY 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**.
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 5: Working with charts

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.
Exporting charts

You can export generated charts to a file system using the Export Chart wizard.

To export a chart:

1. Open the Export Chart wizard. With a chart opened in the editor select File > Export. The Select page of Export wizard opens. Select Chart from the list of available export wizards. The Export Chart wizard opens.
2. Select the source object from the Sources list.
3. From the Export Type drop-down list select the type of exported file. The possible variants are as follows:
   - PDF
   - PNG
4. Click Next. The second page of the Export Chart wizard opens. In the Select export parameters section, specify the size of the image. Click the Update Chart button to refresh the Chart export preview window.
5. Click Finish.

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.
Chapter 5: Working with charts

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. It is available for Column, Bar, Area, and Pie chart types.
   • 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.

Note: Stacked Area Charts do not support 2D With Depth dimension.

Editing the data series options of a display mode

You can edit the data series options of a display mode to specify which columns of a 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.
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 aggregation type for value column from the Aggregation drop-down list. Valid options are:
   - No aggregation
   - Count
   - First
   - Last
   - Maximum
   - Minimum
   - Sum
   - Average
   - Standard Deviation

6. Specify the way of aggregating a value column by selecting or clearing the **Calculate as formula** check box. The **Calculate as formula** check box is available only if you use a calculated column as a value. The expression you have specified for the calculated column is used as aggregation formula.

   If you select the **Calculate as formula** check box, the fields of the columns that provide data for calculated column (and thus for value column) are aggregated at first. The formula uses already aggregated field values as arguments to calculate the result value.

   If you clear the **Calculate as formula** check box, the defined formula calculates the value column fields separately. After the calculations the fields are aggregated to calculate the result value.

7. To remove a column from the series tree, select the column and click **Remove**. The column is added back to the **Result set** tree.

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

**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
Chapter 5: Working with charts

- 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.

**Editing the chart format options of a display mode**

Edit `Chart Format` options of a display mode to change the visual formatting of the 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.
Specifying print options for a chart

You can specify the page setup options that will be used when you print a chart opened in the editor window.

To control the appearance of printed pages for a chart:

1. With an active object in the editor window, open the Page Setup window by selecting **File → Page Setup**.
2. Select the printer that you want to use from the **Printer** list.
3. For detailed settings click **Properties**.
   The Page Setup window opens.
4. Select the size of the printable area from the **Size** list.
5. Select the paper tray to use from the **Source** list.
6. Select the orientation of the printout from the **Orientation** area.
7. Specify the size of the margins of the page in the **Margins** fields.
8. Click **OK** to close the Page Setup window.
9. In the **Scale Options** area, select one from the available printing options:
   a. **Fit to printable area** to fit the chart size to the printable area.
   b. **Scale print to %** to specify the required chart size in percent.
10. To make the specified values default, click **Set as Default**.
11. Click **OK** to close the Page Setup window.
The options that you specified are used to control the appearance of the printed object.
Chapter 6: Working with quick reports

You use quick reports to rapidly generate a printable report from a query result set.

You use the Create New Quick Report wizard to define the report ‘break’ sections, determine the columns over which data will be reported. For example, report sales by region, followed by country within each region. In the Create New Quick Report wizard, you can also define the aggregation function applied to each column value (ascending, descending, count, maximum, minimum, sum, average, and so on), as well to define the formatting (text, alignment, font, background color, and so on) of both the detail and summary rows of the report.

Creating quick reports

You can create a quick report from the data in a query by using the Create New Quick Report wizard.

To create a quick report:

1. To open the Create New Quick Report wizard, select File → New → Other, expand the CorVu Objects and select Quick Report.
2. On the first page of the Create New Quick Report wizard, specify the quick report name and select the required source query. Select the Copy settings from grid check box to apply such grid settings as grouping and detail columns distribution and formatting options to a quick report. Click Next.
   
   **Note:** You can create a quick report on the basis of an opened query result set. With an active query in the editor window, select Query → Transfer To → Quick Report. In this case, the first page of the Create New Quick Report wizard is skipped and the Columns page is opened automatically.

4. On the Format page of the wizard, set the following quick report options:
   - Report title and total line format options
   - Columns order
   - Column headers format options
   - Column groups
   - Total and detail columns sets
   - Conditional format for report columns
   - Grouping columns format options
   - Header and Footer of grouping columns
   - Detail columns format options
   - Custom data objects that can be assigned to report columns.
5. Click Finish. The quick report is displayed on the Design page of the editor where you can manually customize the look of the quick report. You can view the print version of the quick report on the Preview page of the editor.
Arranging grouping and detail columns

You can add and remove the grouping and detail columns from a quick report and convert a grouping column to a detail column and vice versa.

You must open the Create New Quick Report wizard.

To arrange the grouping and detail columns in a quick report:

1. Open the **Columns** page of the Create New Quick Report wizard.
2. To add or remove grouping columns, click **Move to grouping columns set** or **Remove from grouping columns set**.
3. To add or remove detail columns, click **Move to detail columns set** or **Remove from detail columns set**.

**Note:** You can add all available columns to the **Detail Columns** list by clicking **Move all to detail columns set**. To remove all columns from the **Detail Columns** list, click **Move all from detail columns set**.

4. To convert a grouping column to detail column and vice versa, select the columns and then click **Move to grouping columns set** or **Move to detail columns set** buttons.
5. To change the order of grouping columns, use **Move selected columns up** or **Move selected columns down** buttons to the right of the **Grouping Columns** list.
6. If you have specified all the required parameters for the quick report, click **Finish** to close the Create New Quick Report wizard and display the quick report on the **Design** page of the editor where you can manually customize the look of the report.

**Next:** Specifying how to display the report title and report total information, on page 132

**Related topics**

- Specifying how to display the report title and report total information, on page 132
- Specifying how to display column headers, on page 134
- Creating column groups, on page 134
- Creating total and detail column sets, on page 135
- Applying conditional formatting, on page 136
- Specifying format options for Grouping Columns, on page 138
- Specifying how to display Header and Footer information, on page 138
- Specifying format options for the Details Columns, on page 140
- Specifying quick report column format options, on page 141
- Assigning custom data objects to quick report columns, on page 142

Specifying how to display the report title and report total information

You can specify how to display the report title and report total information of a quick report.

**Prerequisite**

You must open the Create New Quick Report wizard.
To specify how to display the report title and report total information in the quick report:

1. Open the **Format** page of the Create New Quick Report wizard.
2. Select **Report Settings** from the tree.
3. Select the **Visible** check box in the **Report Title** section to display the title of the report. The **Report Title** options become available.
4. Specify the title of the report in the **Text** field.

   **Note:** You can view the quick report title only on the printed report.

5. From the **Alignment** menu, specify where on the page the report title must be displayed. Valid options are:
   - **Left** - The title is displayed on the left hand side of the header.
   - **Center** - The title is displayed in the center of the header.
   - **Right** - The title is displayed on the right hand side of the header.

6. Specify the font of the title in the **Font** field.
7. Specify the background color of the title in the **Background** field.
8. Select the **Visible** check box in the **Total Line** section to include a totals line at the bottom of the report. The **Total Line** options become available.
9. Specify the font of the totals line text in the **Font** field.
10. Specify the background color of the totals line in the **Background** field.
11. In the **Common settings** area, specify how to display the quick report grid:
    - To display row and column numbers, select the **Show row and column numbers** check box.
    - To display grid lines, select the **Show grid lines** check box.

    By default, both check boxes are cleared.
12. If you have specified all the required parameters for the quick report, click **Finish** to close the Create New Quick Report wizard and display the quick report on the **Design** page of the editor where you can manually customize the look of the report.

**Next:** Specifying how to display column headers, on page 134

**Related topics**

- Arranging grouping and detail columns, on page 132
- Specifying how to display column headers, on page 134
- Creating column groups, on page 134
- Creating total and detail column sets, on page 135
- Applying conditional formatting, on page 136
- Specifying format options for Grouping Columns, on page 138
- Specifying how to display Header and Footer information, on page 138
- Specifying format options for the Details Columns, on page 140
- Specifying quick report column format options, on page 141
- Assigning custom data objects to quick report columns, on page 142
Specifying how to display column headers

You can specify the look of column headers and the order in which they are displayed in a quick report on the **Columns Headers** page of the Create New Quick Report wizard.

You must open the Create New Quick Report wizard.

To specify how to display column headers:

1. Open the **Format** page of the Create New Quick Report wizard.
2. Expand **Report Settings** node and select **Column Headers** node from the tree.
3. To change the order of the columns, select a column and drag it to the required position.
4. To hide a column, select this column and then select the **Hide** check box.
5. To create hierarchical column headers or group existing column headers, click **Create Header**, and drag the required column headers to this header.

**Note:** If you create a header and do not populate it with at least one of existing headers, it will not be displayed in the report.

6. In the **Header** area, specify the name, font, background color, horizontal and vertical alignment of the column header.
7. If you have specified all the required parameters for the quick report, click **Finish** to close the Create New Quick Report wizard and display the quick report on the **Design** page of the editor where you can manually customize the look of the report.

**Next:** Creating column groups, on page 134

**Related topics**

- Arranging grouping and detail columns, on page 132
- Specifying how to display the report title and report total information, on page 132
- Creating column groups, on page 134
- Creating total and detail column sets, on page 135
- Applying conditional formatting, on page 136
- Specifying format options for Grouping Columns, on page 138
- Specifying how to display Header and Footer information, on page 138
- Specifying format options for the Details Columns, on page 140
- Specifying quick report column format options, on page 141
- Assigning custom data objects to quick report columns, on page 142

Creating column groups

To limit the data that is displayed in a quick report, you can organize the columns in mutually exclusive groups of columns.

You must open the Create New Quick Report wizard.

You can then toggle between the groups to display only those columns that are assigned to that group. Columns assigned to other groups are not displayed. Any columns not assigned to any group are always displayed in the quick report.
To create a column group:

1. Open the **Format** page of the Create New Quick Report wizard.
2. Expand **Report Settings** node and select **Column Groups** node from the tree.
3. On the **Column Groups** page, click **Add group**. A new group is displayed in the **Groups** list.
4. Add the required columns to the group by selecting them from the **Columns** list.
5. To hide the columns already assigned to a group from the **Columns** list, clear the **Show grouped columns** check box.
6. To change the order of the column groups, select a group and click **Move Up** or **Move Down**.

**Note:** The first group in the list is displayed by default when you open the quick report.

7. If you have specified all the required parameters for the quick report, click **Finish** to close the Create New Quick Report wizard and display the quick report on the **Design** page of the editor where you can manually customize the look of the report.

**Next:** Creating total and detail column sets, on page 135

**Related topics**

- Arranging grouping and detail columns, on page 132
- Specifying how to display the report title and report total information, on page 132
- Specifying how to display column headers, on page 134
- Creating total and detail column sets, on page 135
- Applying conditional formatting, on page 136
- Specifying format options for Grouping Columns, on page 138
- Specifying how to display Header and Footer information, on page 138
- Specifying format options for the Details Columns, on page 140
- Specifying quick report column format options, on page 141
- Assigning custom data objects to quick report columns, on page 142

**Creating total and detail column sets**

You can associate detail columns that you do not always want to be displayed in a quick report with a total column.

You must open the Create New Quick Report wizard.

For example, you might have a **Product Costs** column that is calculated from the **Item Cost** and **Number of Items Shipped**. In the quick report you might want only the **Product Costs** column to be displayed, but you still want the other columns available if needed. You can use this feature to make the **Product Costs** column the total column, and specify the columns used to calculate the total column as detail columns.

You can display or hide detail columns by clicking the arrow in the total column header.

To create total and detail column set:

1. Open the Format page of the Create New Quick Report wizard.
2. Expand **Report Settings** node and select **Total/Detail Sets** node from the tree.
3. On the **Total/Detail Sets** page, click **Add Total/Detail Set**. A new set is displayed in the table.
4. Specify a total column from the list of available columns. All of the non-selected columns are automatically displayed in the Detail Columns list.

5. From the Detail Columns list, select detail columns to associate them with the specified total column.

6. If you have specified all the required parameters for the quick report, click Finish to close the Create New Quick Report wizard and display the quick report on the Design page of the editor where you can manually customize the look of the report.

Next: Specifying how to display column headers, on page 134

Related topics
Arranging grouping and detail columns, on page 132
Specifying how to display the report title and report total information, on page 132
Specifying how to display column headers, on page 134
Creating column groups, on page 134
Applying conditional formatting, on page 136
Specifying format options for Grouping Columns, on page 138
Specifying how to display Header and Footer information, on page 138
Specifying format options for the Details Columns, on page 140
Specifying quick report column format options, on page 141
Assigning custom data objects to quick report columns, on page 142

Applying conditional formatting

You can use a conditional format in quick reports to help you visually explore and analyze data, discover critical issues, and identify patterns and trends.

You must open the Create New Quick Report wizard.

Conditional format allows you to specify conditional expressions that control both the display of the data and the behavior of a quick report based on the underlying data. For example, you can use conditional format to highlight in red any fields that indicate year-end sales that fall below a particular figure.

To apply conditional formatting:

1. Open the Format page of the Create New Quick Report wizard.
2. Expand Report Settings node and select Conditional Format node from the tree.
4. On the first page of the New Conditional Format wizard, specify the conditional format name and select the columns and levels of detail to which this format will be applied. Click Next.

Note: If the format for the selected columns or levels of detail has already been set, the new format will override it.

5. On the second page of the wizard, specify conditional expressions and the format options.
6. From the Base column list, select the column which values will be used as a part of condition for formatting. You can select the <Based on the same column> option, to use the values from the column to which you apply the conditional format. If you select this option, the format can only be applied to the columns of the same type (numeric, string, or date/time).
7. Click Add Condition and specify condition parameters in the Selected condition area.
a. From the **Operator** list, select the required operator. Available options are:

- **Not Less Than** - All values that are equal to or greater than the specified value are included.
- **Greater Than** - All values that are greater than the specified value are included.
- **Not Greater Than** - All values that are equal to or less than the specified value are included.
- **Less Than** - All values that are less than the specified value are included.
- **Equal To** - Only values that are equal to the specified value are included.
- **Not Equal To** - All values that are not equal to the specified value are included.
- **Like** - All values that match the specified pattern are included. The pattern can contain the following special characters:
  - % matches zero (0) or more characters.
  - _ matches only one (1) character.
  - ? matches only one (1) character.
- **Unlike** - All values that are not like the specified value are included.
- **Is Null** - Only null values are included.
- **Is Not Null** - All of the values that are not null are included.
- **In Range** - All of the values that fall within the two specified values, along with the values themselves, are included.
- **Outside Range** - All of the values that fall outside of the two values, along with the values themselves, are included.
- **Between** - All of the values that fall within the two specified values are included.
- **Not Between** - All of the values that fall outside of the two specified values are included.

b. Specify the types and values for the operands. You can set either a constant value or a name of the column that will provide required values.

8. In the editor, specify the conditional format options. You can preview the resulting format in the **Sample** area.

a. On the **Text** page of the editor, set the text font, data format, and alignment.

b. On the **Colors** page, set the foreground and background colors. You can use either a single color or gradient.

c. On the **Image** page, you can select an image to display in the report and set its position. Available position options are:

- Before text
- After text
- Replace text.

9. Click **Finish** to close the New Conditional Format wizard.

The created condition format is displayed on the **Conditional Format** page of the Create New Quick Report wizard.

10. If you have specified all the required parameters for the quick report, click **Finish** to close the Create New Quick Report wizard and display the quick report on the **Design** page of the editor where you can manually customize the look of the report.

You can edit created conditional formats by using the Edit Conditional Format wizard. You can also reassign created conditional formats to other columns or levels of detail on the **Assignments** page.

**Next:** Specifying format options for Grouping Columns, on page 138

**Related topics**

- Arranging grouping and detail columns, on page 132
- Specifying how to display the report title and report total information, on page 132
Specifying how to display column headers, on page 134
Creating total and detail column sets, on page 135
Specifying format options for Grouping Columns, on page 138
Specifying how to display Header and Footer information, on page 138
Specifying format options for the Details Columns, on page 140
Specifying quick report column format options, on page 141
Assigning custom data objects to quick report columns, on page 142

Specifying format options for Grouping Columns

You can specify format options for the Grouping Columns of a quick report.

You must open the Create New Quick Report wizard.

To specify Grouping Columns formatting options:

1. Open the Format page of the Create New Quick Report wizard.
2. Select Grouping Columns from the tree.
3. To change a grouping column's name, click the Header cell and specify the new text.
   The text that you specify will display as the grouping column name in the quick report. To set a multiline header you should use “_” symbol to separate lines. In the report it will be substituted by the line break character.
4. To change the sorting order of a grouping column, click the Sort cell and select Ascending or Descending.
5. To change the width of a grouping column, click the Width cell and specify the new width.
6. If you have specified all the required parameters for the quick report, click Finish to close the Create New Quick Report wizard and display the quick report on the Design page of the editor where you can manually customize the look of the report.

Next: Specifying how to display Header and Footer information, on page 138

Related topics
Arranging grouping and detail columns, on page 132
Specifying how to display the report title and report total information, on page 132
Specifying how to display column headers, on page 134
Applying conditional formatting, on page 136
Specifying how to display Header and Footer information, on page 138
Specifying format options for the Details Columns, on page 140
Specifying quick report column format options, on page 141
Assigning custom data objects to quick report columns, on page 142

Specifying how to display Header and Footer information

You can specify the how to display Header and Footer information for a grouping column of a quick report.

You must open the Create New Quick Report wizard.
To specify Header and Footer information for the **Grouping Columns** options:

1. Open the Format page of the Create New Quick Report wizard.
2. In the tree, expand **Grouping Columns**, then one of the column names under it, and select **Header**.
3. To display a header for every grouping column row, select the **Visible** check box.
4. Specify the text that you want to display in the grouping column row header in the **Text** field. The default text is `'column_name: '+@[column]` where 'column_name' is the column name of the grouping column and 'column' is the value of the given grouping column row. For example, if your grouping column is named `DEPT` and it contains department numbers of a given company, this text could generate sequential row headers of `DEPT: 10, DEPT: 20, and so on`. You can change this text to anything that you want, but it is recommended that you keep this overall structure.
5. Specify the font of the grouping column row headers in the **Font** field.
6. Specify the background color of the grouping column row headers in the **Background** field.
7. Specify the height of the header in the **Height** field.
8. You can display charts in the Header of a grouping column. To display a chart in the header, select **Enable chart** check box. In the New chart wizard specify parameters of the chart.

**Note:** If you clear the **Enable chart** check box, the specified chart will not be displayed but it is not removed from the quick report.

9. You can edit a created chart in the Edit chart properties window. To open the Edit chart properties window, click **Chart settings**.
10. To remove a chart, click **Delete chart**.
11. The **Footer** options are identical to the **Header** options. To open the Footer page, select **Footer** in the tree.
12. If you want to display the aggregate totals of the specified detail columns, select the **Visible** check box and specify the totals line text in the **Text** field. The default text is `'Summary of ' + @[column]` where 'column' is the name of a details column which values should be aggregated.
13. If you have specified all the required parameters for the quick report, click **Finish** to close the Create New Quick Report wizard and display the quick report on the **Design** page of the editor where you can manually customize the look of the report.

**Next:** Specifying format options for the Details Columns, on page 140

**Related topics**

- Creating Charts, on page 115
- Arranging grouping and detail columns, on page 132
- Specifying how to display the report title and report total information, on page 132
- Specifying how to display column headers, on page 134
- Creating column groups, on page 134
- Creating total and detail column sets, on page 135
- Applying conditional formatting, on page 136
- Specifying format options for Grouping Columns, on page 138
- Specifying format options for the Details Columns, on page 140
- Specifying quick report column format options, on page 141
- Assigning custom data objects to quick report columns, on page 142
Specifying format options for the Details Columns

You can specify how to display the Details Columns of a quick report.

You must open the Create New Quick Report wizard.

To specify Details Columns formatting options:

1. Open the Format page of the Create New Quick Report wizard.
2. Select Details Columns from the tree.
   The Details Columns page opens.
3. To change a detail column’s column name, click the Header cell and specify the new text.
   The text that you specify will display as the detail column name in the quick report. To set a multiline header you should use "_" symbol to separate lines. In the report it will be substituted by the line break character.
4. To change how a detail column is aggregated, click the Aggregate cell and select a new value.
   Valid options include:
   - Count
   - First
   - Last
   - Maximum
   - Minimum
   - Sum
   - Average
   - Standard Deviation
   Select None to remove aggregation from a column.
5. To change the width of a detail column, click the Width cell and specify the new width.
6. If you have specified all the required parameters for the quick report, click Finish to close the Create New Quick Report wizard and display the quick report on the Design page of the editor where you can manually customize the look of the report.

Next: Specifying quick report column format options, on page 141

Related topics
- Arranging grouping and detail columns, on page 132
- Specifying how to display the report title and report total information, on page 132
- Specifying how to display column headers, on page 134
- Creating column groups, on page 134
- Creating total and detail column sets, on page 135
- Applying conditional formatting, on page 136
- Specifying format options for Grouping Columns, on page 138
- Specifying how to display Header and Footer information, on page 138
- Specifying quick report column format options, on page 141
- Assigning custom data objects to quick report columns, on page 142
Specifying quick report column format options

You can specify the formatting options of individual columns of your quick report.

In addition to customizing the look and feel of a quick report's summary rows, you can specify the formatting options for each column that is displayed in your report.

To specify individual column formatting options:

1. Open the Format page of the Create New Quick Report wizard.
2. Select a column name from the tree.
   The formatting options for each row are identical, whether the column is a grouping column or a detail column.
3. Specify the width of the column in the Column Width field.
4. In the Details area, specify the font, data format, background color, horizontal and vertical alignment of the column. For the columns containing character data, you can apply automatic text wrapping. To do this, select the Wrap text check box.
5. If you have specified all the required parameters for the quick report, click Finish to close the Create New Quick Report wizard and display the quick report in the Preview window.

Related topics
- Arranging grouping and detail columns, on page 132
- Specifying how to display the report title and report total information, on page 132
- Specifying how to display column headers, on page 134
- Creating column groups, on page 134
- Creating total and detail column sets, on page 135
- Applying conditional formatting, on page 136
- Specifying format options for Grouping Columns, on page 138
- Specifying how to display Header and Footer information, on page 138
- Specifying format options for the Details Columns, on page 140
- Assigning custom data objects to quick report columns, on page 142

Specifying data format options for quick report columns

You can specify data values format options for individual columns in the Choose a format window.

To specify data format options:

1. Open the Format page of the Create New Quick Report wizard.
2. Select a column name from the tree. In the Details area, click the ellipsis (...) next to the Format field.
3. Select the format of the data values from the Format drop-down list. Valid options include:
   - As is - The exact text of the data point value is displayed without any formatting.
   - Currency - Data point values are displayed as currency. The locale settings of the current session determine the currency unit that is displayed.
   - Decimal - Data point values are displayed as decimals.
   - Scientific - Data point values are displayed in scientific notation.
   - Percent - Data point 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:

- YYYYYMMxDD
Chapter 6: Working with quick reports

- MMxDxYYYY
- DDxMMxYYYY
- YYxMMxDD
- MMxDDxYY
- DDxDDxYY

The following formats are available for time:
- HHxMMxSS
- HHxMM

The following formats are available for timestamp:
- YYYYxMMxDD HHxMMxSS
- DDxMMxYYYY HHxMMxSS
- YYYYxMMxDD HHxMM
- DDxDDxYYYY HHxMM
- YYxMMxDD HHxMMxSS
- DDxDDxYY HHxMMxSS
- YYxMMxDD HHxMM
- DDxDDxYY HHxMM

4. To include thousands separators with data point values, select the Thousands Separator check box.
5. Specify the number of decimal places that are displayed in the Decimal Places field. A value of '0' indicates that no decimal places are displayed.
6. Select the format of negative numbers in the Negative numbers list.
7. Click OK.
   The Choose a Format window closes and control returns to the Create New Quick Report wizard.

Assigning custom data objects to quick report columns

You can assign custom data objects that you want to be opened when you click a column in your quick report.

You must open the Create New Quick Report wizard.

To specify click events for a column:

1. Open the Format page of the Create New Quick Report wizard.
2. Expand either a grouping or detail column node and select Click Events node from the tree. The Click Events page of the Create New Quick Report wizard contains two separate pages:
   - Drill objects page on which you can assign objects to the selected column.
   - Column prompts page on which you can specify variable values that must be passed to the assigned object.
3. On the Drill objects page, click Add new drill object and browse to the required object in a repository or file system. If the selected object contains prompts, they are displayed in the Prompts Assignments list.

   **Note:** Several objects can be assigned to one and the same column. If you click the column, the first object in the list of assigned objects will be opened. To change the order of the objects by clicking Move Object Up or Move Object Down.
4. You can specify column parameters that will be passed to the assigned object on the Column prompts page.

5. To add a column prompt, click Add new column prompt and specify its parameters including column prompt name, type, and value. Available type options are:
   - Constant to use a constant value
   - Column to set the column that will provide the value
   - Input prompt to specify the prompt of the query on which the report is based that will pass the value to the drill object you assign to the selected column.

6. Assign the created column prompts with the required prompts of the data object in the Prompts Assignments area on the Drill objects page. You can try to assign prompts automatically by clicking Automatch columns by name. If the names of a column prompt and an object prompt do not coincide, you can assign them manually.

7. If you have specified all the required parameters for the quick report, click Finish to close the Create New Quick Report wizard and display the quick report on the Design page of the editor where you can manually customize the look of the report.

Related topics
- Arranging grouping and detail columns, on page 132
- Specifying how to display the report title and report total information, on page 132
- Specifying how to display column headers, on page 134
- Creating total and detail column sets, on page 135
- Applying conditional formatting, on page 136
- Specifying format options for Grouping Columns, on page 138
- Specifying how to display Header and Footer information, on page 138
- Specifying format options for the Details Columns, on page 140
- Specifying quick report column format options, on page 141

Editing quick reports

You can edit the layout of an existing Quick report by adding, removing, or reordering the grouping and detail columns.

You can add, remove or reorder the grouping and detail columns in the report on the Columns page of the Edit Quick Report wizard. To open the Columns page of the wizard, click the Edit Quick Report Columns toolbar button.

You can change the report title and totals line, as well as the formatting options of grouping and detail columns on the Format page of the Edit Quick Report wizard. To open the Format page of the wizard, click the Edit Quick Report Format toolbar button.

These pages are similar to the Column and Format pages of the Create New Quick Report wizard.

You can also modify and change the data source you use to create the quick report.

Changing data source

You can change data sources for already created quick report.

You must create a quick report.

To change data source:
1. With a quick report opened in Preview window, select Report → Change data source.
2. On the first page of Edit Quick Report wizard, specify a new data source. You can select either a previously saved data source or an opened one.
3. If columns in the new data source match the columns of the previously specified data source, you can click Finish to close the wizard and update the report. Otherwise, click Next to proceed.
4. On the second page of the wizard, match the data source columns with the columns of the report. If there is no required match to a report column, select the Remove unmatched columns check box to proceed. All such columns will be excluded from the report.
5. Click Finish to close the wizard and update the report.

Note: All report columns remain their initial properties. If after changing data source the column output format does not match the column data type, the default format for this data type is used.

Related topics
Editing data source, on page 144

Editing data source

You can edit related data sources information for already created quick report.

You must create a quick report.

To edit data source:

2. Edit and save the data source. When you save the modified data source, its columns are compared with the columns of the report. If there is no full compliance, you must match the columns manually in the Edit Quick Report wizard.
3. On the Columns page of the Edit Quick Report wizard, match the data source columns with the columns of the report. If there is no required match to a report column, select the Remove unmatched columns check box to proceed. All unmatched columns will be excluded from the report.

Note: If it is impossible to match the report columns and data source columns, close the Edit Quick Report wizard and edit the data source. If you do not match the columns, data source will not be saved within the report.

4. Click Finish to close the wizard and update the report.

Note: All report columns remain their initial properties. If after editing the column output format does not match the column data type, the default format for this data type is used.

Related topics
Changing data source, on page 143
Modifying your quick report design

Once you have created a quick report, you can make simple changes to the layout right in the **Preview** mode.

You must create a quick report.

To modify a quick report in **Preview** mode:

1. To expand or collapse the content of a group, click the plus or minus sign to the left of the name of this group.
2. To collapse all groups and show only aggregated information, click the minus sign in the header of the grouping column.
3. To view or hide the contents of detail columns assigned to a total column, click the arrow in the header of the total column.
4. To change the columns order, you can select a column and drag it to another position. The red vertical line indicates the possible column position.
5. To change the width of a column or the height of a row, you can drag the column or row border.
6. To move labels and charts within the header and footer of grouping columns, simply drag them to a new position.

**Note:** You can also make changes to the size and position of components in the **Properties** view.

7. To add an extra text field to the header or footer of grouping columns, right-click the header or footer and select **Add Text to Break**.
8. To modify the titles displayed in the header or footer and the text fields that you add, use the **Properties** view.
9. To add a chart to the header or footer of grouping columns, right-click the header or footer and select **Add Chart**.
10. To quickly change the sorting of grouping or detail columns, hide selected columns, or add calculated columns, use the column pop-up menu.
Chapter 7: Working with drill-down paths

A drill-down path is an interactive series of charts that allows you to display several levels of details of a given query.

About drill-down paths

A drill-down path is an interactive series of charts that allows you to display several levels of details of a given query.

Drill-down paths allow you to drill down into several different views of query results. Each level of detail is represented by a different display mode, and each display mode chart can be fully customized. The data that is passed from one level of detail to another is driven by user interaction with the data points of the display mode charts. The movement of data from one level to another is called an analysis path.

For example, you create a column chart that displays the sum of the salaries of employees in each department of a company. The column chart displays a different column for each department. Next, you create a pie chart that displays the sum of the salaries of all of the employees of a specific department, divided up by job type. The pie chart displays a different pie wedge for each job type. The analysis path passes data from the specified department column to the next chart. Now, when you click a specific column of the first chart, the specified department is displayed in the pie chart. You can go back to the original column chart and click a different department’s column. The pie chart now displays the new department’s data points. You can create any number of analysis paths to provide more specific views of the query data.

Creating drill-down paths

Drill-down paths allow you to drill down into several different views of query results.

To create a drill-down path, perform the following procedure:

1. Open the Create New Drill-down Path wizard in one of the following ways:
   - With an active query in the editor window, select Query → Transfer To → Drill-down Path
   - Select File → New → Other
     From the CorVu Objects folder, select the Drill-down Path wizard
   - Click the New Drill-down Path toolbar button.

   The Create New Drill-down Path wizard opens.

   If you opened the Create New Drill-down Path wizard by selecting Query → Transfer To → Drill-down Path, the Chart Type page of the wizard opens.

   Otherwise, the Create New Drill-down Path page opens and you must perform the following actions:
   - Specify a name for the drill-down path in the Name field
   - In the Data Source section, specify a query to use.
     To use a saved in a repository or file query click ellipsis (...) right to the Saved query.
     To use an opened query select one of the opened queries in the corresponding list box.
   - Click Next. The Chart Type page of the wizard opens.

2. Specify chart type options and click Next.
   The Data Series page opens.
Managing drill-down path recording display modes

You can use the **Drill-down structure** pane of the drill-down path **Design** editor to manage the display modes.

Each level of detail of a drill-down path recording is represented by a different display mode. Each drill-down path display mode is edited separately, allowing you to display data using different charts.
and formatting options. You use the **Drill-down structure** pane of the drill-down path **Design** editor to manage the available display modes and the analysis paths that they represent.

To manage drill-down path display modes:

1. Open a drill-down path in the drill-down path **Design** editor.
2. In the **Drill-down structure** pane, right-click a display mode, select **New**, and specify the type of display mode that you want to add.
3. From the **Data source** list select a query, dynamart, table or forecast, on the basis of which you want to create a drill-down level.
4. To add a new object click **Add data source object**. In the Set Data Source dialog select the necessary data source object type and specify a path to it in the **Path** field:
   a. **Embedded** - the object is saved in the drill-down path, so it can be transferred with it to a file or repository when you save this drill-down.
   b. **Linked** - the drill-down path saves a link to the data source object. The object is not saved to a file or repository with a drill-down path.
5. If you add a new data source object, in the **Relation between data sources** area select a column of the parent object from the **Master column** list and a column from the child object from **Detail column** list. These two columns are used to join the data from two objects.
6. If the specified columns have TIMESTAMP format you can select the necessary value from **Calendar function** list to specify the format settings of the columns. Calendar function joins columns with TIMESTAMP data from different data sources if they belong to one and the same time period, for example, one year, quarter or month.
7. Click **Next** to open Chart Type page of the New Drill-down Path Display Mode wizard.
8. Specify chart type options and click **Next**. The Data Series page options.
9. Specify data series options and click **Next**. The Format page opens.
10. Specify format options and click **Finish**. The Create New Drill-down Path wizard closes and the first chart is displayed in the **Drill-down structure** pane of the **Drill-down Path** editor.

**Note:** The new display mode is placed underneath the right-clicked display mode, and represents the next level of detail in the drill-down structure.

11. Right-click a display mode and select **Make Default**. The default display mode is the first chart that is displayed when you view query results
12. Right-click a display mode and select **Edit** from the context menu. The Edit display mode wizard opens.
13. Make the necessary changes to the chart type, data series, and chart format options and click **OK**. The Edit display mode wizard closes.
14. Right-click a display mode and select **Remove** from the context menu to delete it from the **Drill-down structure** tree.
15. Right-click a display mode and select **Move Up** or **Move Down** to change its location in the **Drill-down structure** tree.

**Note:** You can select the necessary level of detail to be displayed in the drill-down by clicking a proper chart area. The Drill-down dialog with a list of available sublevels appears. Select the necessary one and click **OK**. You cannot select a level if there is only one sublevel or the drill-down path is being recorded.
Editing data sources in drill-down paths

You can modify data sources in drill-down paths by applying changes to the source queries.

You must have a previously created drill-down path to edit its data sources.

To edit a necessary data source in a drill-down path:

1. You can open a data source for editing in one of the following ways:
   a. Open the drill-down path in the Design tab and click the Open in a separate editor button in the Result set pane. The source query opens in Diagram tab of the query editor.
   b. Select Drill-down Path → Open in a Separate Editor and select the data source you want to edit from the list.
2. Apply the required changes to the query.
3. Save the query. If you delete the query columns, the Columns Matching window opens. Only unmatched columns are displayed in it.
4. Select Remove unmatched columns check box to delete the unmatched columns automatically. When you match the columns the query will be saved.
5. Switch to the drill-down path editor and run it. The changes are applied to the drill-down path.

Exporting drill-down path data

You can export the data from a specific drill-down path level to a database or to a file.

You use the Export Query Results wizard to export drill-down path data. The procedure is similar to the query results export.

Note: You can also export the data contained on a certain level of a drill-down path as a chart. To do so, use the Export chart wizard.

To export the drill-down path data:

1. Run a drill-down path and navigate to the level that contains the data you need to export.
2. Open the Export Query Results window by selecting File → Export → Result set.
3. Select where you want to export the drill-down path data by selecting one of the following options:
   ▪ Click File system to export the data to a file.
   ▪ Click Database to export the data to a database.
4. Specify the required values in the fields that are available for the type of export you have selected.

Related topics

Exporting charts, on page 125

Exporting drill-down path data to a database

You can export the data from a specific drill-down path level to a database using the Export Query Results wizard.

To export the drill-down path data to a database:
1. Open the Export Query Results wizard. Select the Database destination option.

2. Specify the database where you want to save the 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. In the Export destination area, specify the name and the owner of the table where you want to save the data. You must use the correct format for identifiers when entering the owner or name of the table. If you want to attach a comment to the data, type the comment in the Comment field.

4. 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. Click Next.

5. You can specify how much of the data in the current drill-down path level will be exported by selecting one of the following options:
   - **All** to specify that all of the data will be retrieved and saved to the database.
   - **All - Result set only** to specify that all of the data that has been retrieved so far will be saved to the database. For example, if a query 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** option is not available for exporting the drill-down path data.

6. Specify how existing data is to be handled by selecting one of the following options:
   - **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.

7. 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.

8. Click Finish to export the drill-down path data.

**Exporting drill-down path data to a file**

You can export the data from a specific drill-down path level to a file system using the Export Query Results wizard.

To export the drill-down path data to a file:

1. Open the Export Query Results wizard. Select File system destination option.

2. Specify a name for the file that you are creating in the File name field.

3. Specify the type of file that you are creating. From the Export type field, select one of the following file types:
   - **dbase III** (*.dbf)
   - **Text** (*.txt)
   - **HTML** (*.htm)
   - **WQML Web Query Markup Language** (*.wqml)
   - **XML Extensible Markup Language file** (*.xml)
   - **IXF** (*.ixf)
   - **CSV** (*.csv)
   - **PDF** (*.pdf)

4. Click Next. On the second page of the Export Query Results wizard, you can select additional options that are available depending on the type of file that you are creating.
Specifying print options for a drill-down path

You can specify the page setup options that will be used when you print a drill-down path opened in the editor window.

To control the appearance of printed pages for a drill-down path:

1. With an active object in the editor window, open the Page Setup window by selecting File → Page Setup.
2. Select the printer that you want to use from the Printer list.
3. For detailed settings click Properties.
   The Page Setup window opens.
4. Select the size of the printable area from the Size list.
5. Select the paper tray to use from the Source list.
6. Select the orientation of the printout from the Orientation area.
7. Specify the size of the margins of the page in the Margins fields.
8. Click OK to close the Page Setup window.
9. In the Scale Options area, select one from the available printing options:
   a. Fit to printable area to fit the drill-down path size to the printable area.
   b. Scale print to % to specify the required drill-down path size in percent.
10. To make the specified values default, click Set as Default.
11. Click OK to close the Page Setup window.

The options that you specified are used to control the appearance of the printed object.

Related topics
Specifying file export options, on page 102

5. Click Finish. If you are using CorVu Workstation Pro, a save window specific to your operating system opens. If you are using CorVu Web Pro, a save window opens that lists the available files for download. Click a hyperlink to download the file to your workstation.
Chapter 8: 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.

CorVu 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 CorVu 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 CorVu Objects folder select the Forecasts wizard
   - Click the New Forecast toolbar button.

   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 hierarchal 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 can 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 can influence the choice of forecasting model for the entire forecast. For example, when marketing a new product it might 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 (...). 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 from the **Date column** list.

**Note:** Only date format columns are available in the **Date column** list.

The following formats are available for date:

- YYYYxMMxDD
- MMxDDxYYYY
- DDxDDxYYYY
- YYxDDxDD
- MMxDDxYY
- DDxDDxYY

For time:

- HHxDDxDD
- HHxDD

For timestamp:

- YYYYxDDxDD HHxDDxDD
- DDxDDxDD HHxDD
- YYYYxDDxDD HHxDD
- DDxDDxDD YYYY HHxDD
- YYxDDxDD HHxDDxDD

- YYxDDxDD HHxDDxDD
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.

- DDxMMxYY HHxMMxSS
- YYxMMxDD HHxMM
- DDxMMxYY HHxMM

3. Select the historical data values that you want to use to calculate the forecasted values from the **Historical data** list. The available options are:
   
   a. **All data**: Select this option to specify that all data values will be used to calculate the forecasted values.
   
   b. **From**: Select this option to specify the starting data values that will be used to calculate the forecasted values. Enter the start date in the **From** calendar field. You can also enter the end date in the **To** calendar field. In this case, the data values retrieved between the start and end dates will be used to calculate the forecasted values.
   
   c. **Last**: Select this option to specify that the data values of the last number of periods will be used to calculate the forecasted values. Specify the number of last periods in the **Periods** field.

4. Select the time period over which to aggregate data values from the **Period** list.
5. Specify the number of periods to calculate forecast values in the **Number of future periods** field.
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**.
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.
Chapter 8: Working with forecasts

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.
Chapter 8: Working with forecasts

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 button.
   The Create Seasonality window opens.
5. Select the seasonality time period from the Time period drop-down list.
6. Select the historical data averaging method from the Seasonality type drop-down list.
7. Click OK.
   The Create Seasonality window closes and a seasonality chart and seasonality graph are displayed in the Results editor. The seasonality table displays, for each period, the overall average for all seasons and the average for each seasonality period.

### 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 Cyclicity toolbar button.
   The Create Cyclicity window opens.
5. Select the cycle length from the Number of points drop-down list.
The cycle length is the number of periods expected in a complete cycle.

6. Click **OK**.
   The Create Cyclicity window closes and a cyclicity chart and cyclicity table are displayed in the **Results** editor. The cyclicity table displays, for each period in a cycle, the overall average for all cycles and the average for each individual cycle. For example, if a cycle has twelve periods and the data history is three years then average values for three cycles are displayed.

## 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** toolbar button.
   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.
Exporting forecasts

You can export charts and query result sets of generated forecasts to a file system or database using the Export Chart or Export Query Results wizard.

- To export a chart see the Exporting charts, on page 125 topic.
- To export query results to a database see the Exporting query results to a database, on page 99 topic.
- To export query results to a file see the Exporting query results to a file, on page 100 topic.

Specifying print options for a forecast

You can specify the page setup options that will be used when you print a forecast opened in the editor window.

To control the appearance of printed pages for a forecast:

1. With an active object in the editor window, open the Page Setup window by selecting File → Page Setup.
2. Select the printer that you want to use from the Printer list.
3. For detailed settings click Properties.
   The Page Setup window opens.
4. Select the size of the printable area from the Size list.
5. Select the paper tray to use from the Source list.
6. Select the orientation of the printout from the Orientation area.
7. Specify the size of the margins of the page in the Margins fields.
8. Click OK to close the Page Setup window.
9. In the Page area, specify the forecast objects that you want to print by selecting or clearing the Chart and Table check boxes.
10. In the same area, select one of the available page options:
   a. Print on separate pages to print the forecast objects, the chart and the table, on separate pages.
   b. Print on the same page to print both forecast objects on one page.
11. In the Scale Options area, select one from the available printing options:
   a. Fit to printable area to fit the forecast size to the printable area.
   b. Scale print to % to specify the required forecast size in percent.
12. To make the specified values default, click Set as Default.
13. Click OK to close the Page Setup window.
   The options that you specified are used to control the appearance of the printed object.
Chapter 9: Working with scheduled tasks

Scheduled tasks allow you to create automated actions such as running queries and exporting query results without taking direct action from the user interface.

Scheduled tasks are invaluable tools that help to automate business processes to run at specified dates and times. This is especially useful for actions that must be run on a regular basis, such as weekly sales reports that are supposed to be distributed to several different locations, or quarterly earnings reports.

To create a scheduled task:

1. Select View → Scheduled Tasks or File → New → Scheduled Tasks.
2. Click Add task to create a scheduled task.
3. Specify a name for the task in the Task name field.
4. Click Add Action.
   The Action Wizard opens.
5. Specify a name for the action in the Action name field.
6. Select what type of action you want to add and click Next.
7. Specify the options for the selected task type and click Finish.
   The Action Wizard closes and the new action is added to the Actions list.
8. When you have added all of the actions that you want the task to run, click OK.
   The Task window closes and control returns to the Task List window.
9. Click Import task to import a task.
10. Click Export task to export a task.
11. To schedule a task, select the task that you want to schedule and click Schedule task.
12. Click OK.
   The Task List window closes.

Adding an execute object action

The execute object action allows you to automatically execute a visual report, visual query, or quick report, and export, mail, or print it.

To add an execute object action to a scheduled task:

1. Open a task in the Task List window.
2. Click Add Action.
   The Action Wizard opens.
3. Specify a name for the action in the Action name field, click Execute object, and then click Next.
   The Specify object page opens.
4. Select the type of object that you want to execute.
5. Specify the location of the object.
   There are two options:
   a. To specify the static source location of the object, click Path and then click the ellipsis (...) at the bottom right of the Source area.
      The Open window opens.
   b. Navigate to and select the object and click Finish.
      The Open window closes.
c. To specify a conditional source location for the object, click Formula and then click the ellipsis (...). The Formula editor opens.

d. Specify the formula to use to determine the location of the object and click OK. The Formula editor closes.

6. To add an available data source, click Add Data Source. The Add Data Source window opens.

7. Select Set a new data source check box to specify a particular data source against which you want to execute the object. Then click the ellipsis (...), the Add Data Source window opens, select the required data source and click OK. The Add Data Source window closes.

8. Navigate to and select the data source that you want to add, specify a login name and password for the data source, and click OK. The Add Data Source window closes.

9. Click Next. The Specify operations page opens.

10. Click Add Operation. The Operation Wizard opens.

11. Select an operation and click Next.

12. Specify the options for the selected operation and click Finish. The Operation Wizard closes.

13. When you have finished adding all of the operations that you want the action to run, click Finish. The Action Wizard closes and control returns to the Task window.

14. Click OK. The Task windows closes and control returns to the Task List window.

15. Click OK. The Task List window closes.

Specifying export query operation options

You use the export query operation to export query results to a file.

To add an export query operation to an action:

1. Open the Specify operations page of the Action Wizard and click Add Operation. The Operation Wizard opens.

2. Click Export query and then click Next. The Specify export parameters page opens.

3. Select the file type that you want to export from the Export type drop-down list. Valid file types are:
   - Text (*.txt)
   - CSV (*.csv)
   - PDF (*.pdf)
   - HTML (*.htm)
   - IXF (*.ixf)
   - XML Extensible Markup Language file (*.xml)
   - XLS Microsoft Excel file (*.xls)

4. Specify the location to save the object. There are two options:
a. To specify a static save location for the object, click **Path** and then click the ellipsis (…) at the bottom right of the **Source** area. The Save As window opens.
b. Navigate to the directory location where you want to save the object and click **Finish**. The Save As window closes.
c. In case the **Root output directory** is specified on the **Server-Side File System** page of the Preferences window, you can just type a path and a file name or only a file name in the **Path** field. The file will be exported to this root directory automatically. In CorVu Web Pro 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.
d. To specify a conditional save location for the object, click **Formula** and then click the ellipses (…). The Formula editor opens.
e. Specify the formula to use to determine the location to save the object and click **OK**. The Formula editor closes.

5. Select the **Delete after task completion** check box if you want to remove the exported file from the specified location when the scheduled task is completed.
6. Click **Next**. The Specify [file type] export options page opens. On this page you can select additional options that are available depending on the type of file that you are creating.
7. Click **Finish**. The Operation Wizard closes.

**Related topics**
Specifying file export options, on page 102
Setting the directory location preference for objects generated by Export actions, on page 28

**Specifying export query to mail operation options**

You use the export query to mail operation to send query results in an email.
To add an export query to mail operation to an action:

1. Open the Specify operations page of the Action Wizard and click **Add Operation**. The Operation Wizard opens.
2. Click **Export query to mail** and then click **Next**. The Specify export parameters page opens.
3. Select the file type that you want to export from the **Export type** drop-down list. Valid file types are:
   - **Text** (*.txt)
   - **CSV** (*.csv)
   - **PDF** (*.pdf)
   - **HTML** (*.htm)
   - **IXF** (*.ixf)
   - **XML** Extensible Markup Language file (*.xml)
   - **XLS** Microsoft Excel file (*.xls)
4. Specify the name of the file attachment in the email in the **Attachment name** field.
5. If you want to specify a conditional file attachment name, click the button next to the field. The Formula editor opens.
6. Specify the formula to use to determine the name of the attachment and click **OK**.
The Formula editor closes.

7. Click **Next**.

The Setup [file type] export options page opens. On this page you can select additional options that are available depending on the type of file that you are creating.

Click **Next**.


**Note:** The values that you set for can be specified dynamically during Iterator work with the help of substitution variables. You can use an action group with Iterator to send separate email messages to several recipients. To do so you need to specify Iterator query that returns all necessary email information, define Iterator variables, and type the variable names in the corresponding fields on the Specify mail settings page using the equality sign (=Form, =SMTP).

When you run the task, Iterator goes over the Iterator query records, extracts a set of email parameters at each iteration, and passes it to the Export query to mail operation.

- From the **Format** list, select the format of the body text. The available formats are Text and HTML.
- Specify the recipient’s email address in the **To** field.
- Specify additional recipients in the **Cc** field.
- Specify the subject of the email in the **Subject** field.
- Specify the content of the email in the **Message** field.

**Note:**

If you want to include an embedded image in the text of the email, you must insert the following HTML snippet into the text:

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

where *cid:* is the standard abbreviated content identifier required when referencing an embedded image within a formatted email, and where *[image name]* is the name of the image file. For example, to embed an image file named `image1.png`, you would insert the following HTML snippet into the text of the email:

```
<img src="cid:image1.png"></img>
```

- Specify the email address from which the email is sent in the **From** field.

9. To specify SMTP server parameters, click **SMTP settings**. In the SMTP Settings window, specify whether to use a server predefined by administrator or a custom one.

   a. If you decide to use a predefined SMTP server, select the required server from the list of predefined servers.
   b. If you decide to use a custom server, specify server parameters including its address and port number. If the SMTP server requires authentication, select **Use SMTP authentication**, and then specify user name and password. If it is necessary to specify additional access preferences, click **Advanced** to open the Advanced SMTP Settings window where you can add required keywords supported by javamail. Click **OK**.

   Click **OK** to close the SMTP Settings window.

10. Click **Finish** to close the Operation wizard.

**Related topics**
Specifying print query operation options

You use the print query operation to print the specified query results.

To add a print query operation to an action:

1. Open the Specify operations page of the Action Wizard and click Add Operation.
   The Operation Wizard opens.
2. Click Print query and then click Next.
   The Specify printer page opens.
3. Select the printer that you want to use to print the query results from the Printer drop-down list.
4. Click Properties.
   The Page Setup window opens.
5. Select the size of the printable area from the Size drop-down list.
6. Select the paper tray to use from the Source drop-down list.
7. Select the orientation of the printout from the Orientation radio group.
8. Specify the size of the margins of the page in the Margins fields.
9. Click OK.
   The Page Setup window closes and control returns to the Operation Wizard.
10. Click Finish.
    The Operation Wizard closes.

Specifying export quick report to mail operation options

Use the export quick report to mail operation to send a quick report in an email.

To specify export quick report to mail operation options, perform the following procedure:

1. Open the Specify operations page of the Action Wizard and click Add Operation.
   The Operation Wizard opens.
2. Click Export quick report to mail and then click Next.
   The Specify export parameters page opens.
3. Select the file type that you want to export from the Export type drop-down list.
   Valid file types are:
   - PDF (*.pdf)
   - HTML (*.htm)
4. Specify the name of the file attachment in the email in the Attachment name field.
5. If you want to specify a conditional file attachment name, click the ellipsis (...).
   The Formula editor opens.
6. Specify the formula to use to determine the name of the attachment and click OK.
   The Formula editor closes.
7. Click Next.
   The Setup [file type] export options page opens. On this page you can select additional options that are available depending on the type of file that you are creating.
8. Click Next. On Specify email settings page, specify values for the following parameters:
- Select the format of the body text from the **Format** list. The available formats are Text and HTML.
- The address that the email will be sent to in the **To** field
- The address that a carbon copy of the email will be sent to in the **Cc** field
- The subject of the email in the **Subject** field
- The body text of the email in the **Message** field
- The address that will send the email in the **From** field

9. To specify SMTP server parameters, click **SMTP settings**. In the SMTP Settings window, specify whether to use a server predefined by administrator or a custom one.
   a. If you decide to use a predefined SMTP server, select the required server from the list of predefined servers.
   b. If you decide to use a custom server, specify server parameters including its address and port number. If the SMTP server requires authentication, select **Use SMTP authentication**, and then specify user name and password. If it is necessary to specify additional access preferences, click **Advanced** to open the Advanced SMTP Settings window where you can add required keywords supported by javamail. Click **OK**.

Click **OK** to close the SMTP Settings window.

10. Click **Finish**.
    The Operation Wizard closes.

**Related topics**
**Specifying file export options**, on page 102

**Specifying export quick report operation options**

Use the export quick report operation to export quick report results to a file.

To specify export quick report operation options, perform the following procedure:

1. Open the Specify operations page of the Action Wizard and click **Add Operation**. The Operation Wizard opens.
2. Click **Export quick report** and then click **Next**. The Specify export parameters page opens.
3. Select the file type that you want to export from the **Export type** drop-down list. Valid file types are:
   - **PDF** (*.pdf)
   - **HTML** (*.htm)
4. Specify the location to save the object.
   There are two options:
   a. To specify a static save location for the object, click **Path** and then click the ellipsis (...) at the bottom right of the **Source** area. The Save As window opens.
   b. Navigate to the directory location where you want to save the object and click **Finish**. The Save As window closes.
   c. In case the **Root output directory** is specified on the **Server-Side File System** page of the Preferences window, you can just type a path and a file name or only file name in **Path** field. The file will be exported to this root directory automatically. In CorVu Web Pro 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.
d. To specify a conditional save location for the object, click **Formula** and then click the ellipsis (...).
The Formula editor opens.
e. Specify the formula to use to determine the location to save the object and click **OK**.
The Formula editor closes.

5. Select the **Delete after task completion** check box if you want to remove the exported file from the specified location when the scheduled task is completed.

6. Click **Next**.
The Specify [file type] export options page opens. On this page you can select additional options that are available depending on the type of file that you are creating.

7. Click **Finish**.
The Operation Wizard closes.

**Related topics**
---
Specifying file export options, on page 102
Setting the directory location preference for objects generated by Export actions, on page 28
---

**Specifying export visual report to mail operation options**

You use the export visual report to mail operation to send a visual report in an email.

To specify export visual report to mail operation options, perform the following procedure:

1. Open the Specify operations page of the Action Wizard and click **Add Operation**.
The Operation Wizard opens.
2. Click **Export visual report to mail** and then click **Next**.
The Specify export parameters page opens.
3. Select the file type that you want to export from the **Export type** drop-down list.
Valid file types are:
   - PDF (*.pdf)
   - HTML (*.htm)
4. Specify the name of the file attachment in the email in the **Attachment name** field.
5. If you want to specify a conditional file attachment name, click the ellipsis (...).
The Formula editor opens.
6. Specify the formula to use to determine the name of the attachment and click **OK**.
The Formula editor closes.
7. Click **Next**.
The Setup [file type] export options page opens. On this page you can select additional options that are available depending on the type of file that you are creating.
8. Click **Next**. On Specify email settings page, specify values for the following parameters:
   - Select the format of the body text from the **Format** list. The available formats are Text and HTML
   - The address that the email will be sent to in the **To** field
   - The address that a carbon copy of the email will be sent to in the **Cc** field
   - The subject of the email in the **Subject** field
   - The body text of the email in the **Message** field
   - The address that will send the email in the **From** field
9. To specify SMTP server parameters, click **SMTP settings**. In the SMTP Settings window, specify whether to use a server predefined by administrator or a custom one.
a. If you decide to use a predefined SMTP server, select the required server from the list of predefined servers.
b. If you decide to use a custom server, specify server parameters including its address and port number. If the SMTP server requires authentication, select Use SMTP authentication, and then specify user name and password. If it is necessary to specify additional access preferences, click Advanced to open the Advanced SMTP Settings window where you can add required keywords supported by javamail. Click OK.

Click OK to close the SMTP Settings window.

10. Click Finish.
The Operation Wizard closes.

Related topics
Specifying file export options, on page 102

Specifying export visual report operation options

Use the export visual report operation to export a visual report to a file.

To specify export visual report operation options, perform the following procedure:

1. Open the Specify operations page of the Action Wizard and click Add Operation.
The Operation Wizard opens.
2. Click Export visual report and then click Next.
The Specify export parameters page opens.
3. Select the file type that you want to export from the Export type drop-down list.
Valid file types are:
   - PDF (*.pdf)
   - HTML (*.htm)
4. Specify the location to save the object.
   There are two options:
   a. To specify a static save location for the object, click Path and then click the ellipsis (...) at the bottom right of the Source area.
      The Save As window opens.
   b. Navigate to the directory location where you want to save the object and click Finish.
      The Save As window closes.
   c. In case the Root output directory is specified on the Server-Side File System page of the Preferences window, you can just type a path and a file name or only a file name in the Path field. The file will be exported to this root directory automatically. In CorVu Web Pro 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.
   d. To specify a conditional save location for the object, click Formula and then click the ellipsis (...).
      The Formula editor opens.
   e. Specify the formula to use to determine the location to save the object and click OK.
      The Formula editor closes.
5. Select the Delete after task completion check box if you want to remove the exported file from the specified location when the scheduled task is completed.
6. Click Next.
The Specify [file type] export options page opens. On this page you can select additional options that are available depending on the type of file that you are creating.
7. Click Finish.
Specifying export visual portfolio to mail operation options

You use the export visual portfolio to mail operation to send a visual portfolio in an email.

To specify export visual portfolio to mail operation options, perform the following procedure:

1. Open the Specify operations page of the Action Wizard and click Add Operation. The Operation Wizard opens.
2. Click Export visual portfolio to mail and then click Next.

   Note: The available export format is PDF (*.pdf) only.

3. Specify the name of the file attachment in the email in the Attachment name field.
4. If you want to specify a conditional file attachment name, click the ellipsis (...). The Formula editor opens.
5. Specify the formula to use to determine the name of the attachment and click OK. The Formula editor closes.
6. Click Next. The Specify email settings page opens. Specify values for the following parameters:
   - Select the format of the body text from the Format list. The available formats are Text and HTML
   - The address that the email will be sent to in the To field
   - The address that a carbon copy of the email will be sent to in the Cc field
   - The subject of the email in the Subject field
   - The body text of the email in the Message field
   - The address that will send the email in the From field
7. 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.
8. Click Finish. The Operation Wizard closes.
Specifying export visual portfolio operation options

Use the export visual portfolio operation to export a visual portfolio to a file.

To specify export visual portfolio operation options, perform the following procedure:

1. Open the Specify operations page of the Action Wizard and click Add Operation. The Operation Wizard opens.
2. Click Export visual portfolio and then click Next.

   **Note:** The available export format is PDF (*.pdf) only.

   The Specify export parameters page opens.
3. Specify the location to save the object.
   There are two options:
   a. To specify a static save location for the object, click Path and then click the ellipsis (...) at the bottom right of the Source area. The Save As window opens.
   b. Navigate to the directory location where you want to save the object and click Finish. The Save As window closes.
   c. In case the Root output directory is specified on the Server-Side File System page of the Preferences window, you can just type a path and a file name or only a file name in the Path field. The file will be exported to this root directory automatically. In CorVu Web Pro 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.
   d. To specify a conditional save location for the object, click Formula and then click the ellipsis (...).
   The Formula editor opens.
   e. Specify the formula to use to determine the location to save the object and click OK. The Formula editor closes.
4. Select the Delete after task completion check box if you want to remove the exported file from the specified location when the scheduled task is completed.
5. Click Next. The Specify [file type] export options page opens. On this page you can select additional options that are available depending on the type of file that you are creating.
6. Click Finish. The Operation Wizard closes.

Specifying export visual report operation options

Use the export visual report operation to export a visual report to a file.

To specify export visual report operation options, perform the following procedure:

1. Open the Specify operations page of the Action Wizard and click Add Operation. The Operation Wizard opens.
2. Click Export visual report and then click Next. The Specify export parameters page opens.
3. Select the file type that you want to export from the Export type drop-down list. Valid file types are:
Adding a resource manipulation task

The resource manipulation action allows you to automatically copy, paste, delete, and email resources that are generated by the scheduled task.

To add a resource manipulation action to a scheduled task:

1. Open a task from the Task List window.
2. Click Add Action.
   The Action Wizard opens.
3. Specify a name for the action in the Action name field, click Resource manipulation, and then click Next.
   The Specify resource manipulation type page opens.
4. Click Copy or Send mail and then click Next.
   The next page of the wizard opens.
5. Specify the Copy or Send email options and click Finish.
   The Action Wizard closes and control returns to the Task window.
6. Click OK.
   The Task window closes and control returns to the Task List window.
7. Click OK.

Related topics
Specifying file export options, on page 102
Setting the directory location preference for objects generated by Export actions, on page 28
Specifying copy resource parameters

The copy resource manipulation action allows you to automatically copy, paste, and delete resources that are generated by the scheduled task.

To specify copy resource parameters:

1. Open the Specify resource manipulation type page of the Action Wizard.
2. Click Copy and then click Next.
   The Specify copy resource parameters page opens.
3. Specify the resource that you want to copy.
   There are two options:
   a. To specify the static source location of the object, click Path and then click the ellipsis (…) at the bottom right of the Source area.
      The Open window opens.
   b. Navigate to and select the object and click Finish.
      The Open window closes.
   c. In case the Root output directory is specified on the Server-Side File System page of the Preferences window, you can just type a path and a file name or only a file name in the Path field. The file will be exported to this root directory automatically. In CorVu Web Pro 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.
   d. To specify a conditional source location for the object click Formula and then click ellipse (…).
      The Formula editor opens.
   e. Specify the formula to use to determine the location of the object and click OK.
      The Formula editor closes.
4. Specify where you want to save the resource.
   There are two options:
   a. To specify the static source destination of the resource, click Path and then click the ellipsis (…) at the bottom right of the Source area.
      The Save As window opens.
   b. Navigate to the location where you want to save the resource and its file name and click Finish.
      The Save As window closes.
   c. To specify a conditional source destination for the resource, click Formula and then click the ellipsis (…).
      The Formula editor opens.
   d. Specify the formula to use to determine where to save the resource and click OK.
      The Formula editor closes.
5. To cut and paste the resource from the source location to the destination location, select the Move check box.
6. To create a new folder to contain the moved resource, select the Create folders check box.
7. To delete the moved copy of the resource once the task has been completed, select the Delete after task completion check box.
8. Click Finish.
    The Action Wizard closes and control returns to the Task window.

Related topics
Specifying mail settings

The send mail resource manipulation action allows you to automatically send generated resources in an email.

To specify mail settings:

1. Open the Specify resource manipulation type page of the Action Wizard.
2. Click **Send mail** and then click **Next**.
   The Specify mail settings page opens.
3. Specify the format of the body text in the **Format** list. The available formats are Text and HTML.
4. Specify the recipient of the email in the **To:** field.
5. Specify any additional recipients in the **Cc:** field.
6. Specify the subject of the email in the **Subject** field.
7. Specify the text of the email in the **Message** field.
8. Specify the email address that will send the email in the **From:** field.
9. Specify the SMTP server of the email address in the **SMTP server** field.
10. If the SMTP server requires authentication, select **Use SMTP authentication**, and then specify user name and password in the **User** and **Password** fields.
11. Click **Add**.
   The Enter path as text or formula window opens.
12. Specify the path to the resource that you want the task to send in an email and click **OK**.
   You can attach images as well as files of other available types.
   If you want to include an embedded image in the text of the email, you must insert the following HTML snippet into the text:

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

where **cid:** is the standard abbreviated content identifier required when referencing an embedded image within a formatted email, and where **[image name]** is the name of the image file. For example, to embed an image file named **image1.png**, you would insert the following HTML snippet into the text of the email:

        <img src="cid:image1.png"></img>

The Enter path as text or formula window closes.
13. Click **Finish**.
   The Action Wizard closes and control returns to the Task window.

Adding an action group

You can use action groups to create a series of actions that should be performed successively.
Before creating an action group, you must first create a scheduled task.

To add an action group to a scheduled task:

1. Open the Task window in one of the following ways:
Double-click a task in the Tasks list;  
Select a task from the Tasks list, and then click Edit Task.

2. In the Task window, click Add Action to open the Action wizard.
3. Type a name in the Action name field.
4. Select Action Group, and then click Finish.

Action group properties

Each action group has properties that can be found and modified in the action group context.

To see the options available in the context, right-click Actions field.

The following operations are listed on the context menu:
1. Rename invokes Rename Action window and allows you to rename the current group.
2. Set Iterator invokes Iterator Parameters window. In the Query Path field select a query, analytical query, and so on for the Iterator.
3. Choose a Data Source.
4. Set Login / Password information to access a Data Source if necessary.
5. Set Prompt name to use names of the variables in the formulas.
6. Add Data Source allows adding a new Data Source.
7. Remove Data Source allows removing a Data Source.
8. Refresh Data Sources refreshes the list of data Sources.
9. Edit Iterator appears if the Iterator has already been set and allows editing its properties.
10. Remove Iterator removes the selected Iterator.
11. Manage prompts property helps to bind the prompts, already existing and repeating in the query, in the group during the iteration process.

Adding actions to an action group

You can use drag-and-drop method to add actions to an action group.

You must create an action group.

You must create an action.

To add an action to an action group:

1. Open the Task window in one of the following ways:
   - Double-click a task in the Tasks list;
   - Select a task from the Tasks list, and then click Edit Task.
2. In the Task window, select an action that you want to add to a group.
3. Drag the selected action onto an action group item.

Next: Setting Iterator parameters

Related topics

Adding an execute object action, on page 165
Adding a resource manipulation task, on page 175
Removing actions from an action group

You can use drag-and-drop method to remove actions from an action group. You must create an action group and populated it with actions.

To remove an action from an action group:

1. Open the Task window in one of the following ways:
   - Double-click a task in the Tasks list;
   - Select a task from the Tasks list, and then click Edit Task.
2. In the Task window, expand an action group that you want to edit.
3. Select an action that you want to remove from the action group.
4. Drag the selected action outside the action group.

Setting Iterator parameters

You can use Iterator to run an action group for several times in a cycle. You use the Iterator Parameters window to manage Iterator preferences.

You must create an action group.

To set Iterator parameters:

1. In the Actions list of the Task window, right-click the action group that you want to run in a cycle, and then select Set Iterator from the pop-up menu.
2. In the Iterator Parameters window, click the ellipsis (…) to specify a query for Iterator.
   The columns of this query define Iterator variables and the number of rows specifies the number of iterations.
3. In the Open window navigate to and select an object that you want to use as a query for Iterator.
   The following objects can be used as a query:
   - Charts
   - Queries
   - Analytical Queries
   - Dynamarts
   - Compound Dynamarts
   - Tables
4. Click Finish. The path to the selected object is displayed in the Query path field.
5. The data source that contains the selected object is added to the Data Sources list. You can specify your login information for a data source in the Login and Password fields.
6. To add another data source to the list, perform the following procedure:
   a. Click Add Data Source to open the Add Data Source window.
   b. From the Data Source list, select a data source that you want to add.
   c. Type your login information in the User name and Password fields.
   d. Select the Remember the password check box if you want CorVu to remember the password for the user ID that you entered in the User name field.
Chapter 9: Working with scheduled tasks

7. To remove a previously added data source from the list, select this data source, and then click **Remove Data Source**.

8. In the **Variables** list, link the query columns with the prompts used in the source object that you specify in Execute object actions. Click in the **Prompt** cell and select the prompt from the drop-down list.

9. Use the **Continue on failure** check box to specify whether to continue Iterator work when an error occurs. If the check box is selected and an error occurs within any iteration, Iterator skips this iteration and continues working. If an error occurs and the check box is cleared, Iterator stops working.

   **Note:** All error messages can be viewed in the **Output** view.

10. Click **OK** to close the Iterator Parameters window.

Next: **Managing prompts in an action group**

Related topics
   - Adding an execute object action, on page 165

Managing prompts in an action group

You can create a scheduled task that runs a variety of data objects such as queries, quick reports, visual reports, and procedures containing parameters.

You use the Manage Shared Prompts window to define how the parameter values must be passed to the source objects specified in the task.

   **Note:** Depending on the type of an object, parameters are represented by different elements. For queries, parameters are prompts defined for these queries. For quick reports and visual reports, parameters are prompts specified in queries on which these reports are based. For procedures, parameters are variables defined in these procedures.

You can group similar prompts used in the source objects so that all the prompts within one prompt group receive the value passed to this group. By default, there are as many groups as prompts in the source objects within the action group. Each group stores one particular prompt and borrows its name. You can move prompts between the groups. If you move the last prompt from the group, this group is removed automatically because empty groups are not allowed.

You must create an action group and populated it with execute object actions.

To manage shared prompts:

1. Open the Task window by selecting a task from the **Tasks** list and clicking **Edit Task**.
2. In the Task window, right-click an action group, and then select **Manage Prompts**.
3. From the **Available prompts** list in the Manage Shared Prompts window, select a prompt that you want to add to a group.
4. From the **Shared prompts** list, select a group that you want to add the prompt to.
5. Click **Add to Group**. The selected prompt is added to the prompt group.
6. To remove a prompt from a prompt group, select a prompt from the **Shared prompts** list, and then click **Detach Prompt**.
7. To rename a prompt group, select the prompt group that you want to rename, and then click **Rename prompt group**.
8. In the Prompt Group window, specify the name and string to display, and then click OK.
9. Add an Iterator query for the action group and link the query columns with the parameters used in the source objects that you specified in actions within the action group.

Related topics
Adding focus to queries by using prompts

Specifying parameter values for actions and action groups

You can create a scheduled task that runs a variety of data objects such as queries, quick reports, visual reports, and procedures containing parameters.

You use the Set Variables window to define how the parameter values must be passed to the source objects which are specified in the task. You can also use the specified parameters in actions themselves as substitution variables. For example, you can form the file system path by using parameters.

**Note:** Depending on the type of an object, parameters are represented by different elements. For queries, parameters are prompts defined for these queries. For quick reports and visual reports, parameters are prompts specified in queries on which these reports are based. For procedures, parameters are variables defined in these procedures.

You can set parameter values for both action groups and single actions. You can also set the iterator for the same group. In this case, the parameters provided by the iterator will take precedence.

To specify parameter values:

1. Open the Task window by selecting a task from the Tasks list and clicking Edit Task.
2. In the Task window, right-click an action or action group, and then select Set Variables. All the parameters specified for the selected data object are displayed in the list of parameters in the Set Variables window.

**Note:** If you decide to set parameter values to an action group to which you have already assigned shared prompts, the specified shared prompts are displayed instead of parameters defined in the source object.

3. To add a parameter, click Add Variable, and then specify its name and the value in the Add Variable window.
4. To edit or remove a variable, select the variable you want to edit or remove, and click Edit or Remove.
5. To set the parameter value, open the Add Variable or Edit Variable window and enter either the constant value or a formula in the Value field. To open the Formula editor window, click the ellipsis (...) next to the Value field.

**Note:** You can also set the parameter value directly in the Set Variables window by simply typing it in the required cell of parameters table.

6. You can change the order in which parameters are calculated by clicking Move Up and Move Down. Parameters are calculated sequentially as they appear in the list and those that have already been calculated can be used to calculate other parameters.
7. Click OK to close the Set Variables window.
Chapter 9: Working with scheduled tasks

Copying scheduled tasks

You can add a scheduled task by copying an existing one and modifying it.

To copy a scheduled task:

1. Select View → Scheduled Tasks. The Task List window opens.
2. From the Tasks list, select the task you want to copy and click Copy. The Task window opens.

   **Note:** You can select only one task from the Tasks list to copy.

3. In the Task name field, type a name for the copied task.
4. From the Repository connection list, select the repository connection that you want the task access to.
5. Click Set User Information to specify the login name and password for the repository in the Set User Information for Repository window.
6. In the Actions list, you can edit the actions you want to copy with the task, add, or remove them.

   **Note:** The time settings of scheduled tasks execution are not copied.

7. Click OK. The scheduled task will be copied and added to the Tasks list in the Task List window.

Scheduling tasks

You can schedule a task to run in the .

To schedule a task:

1. Select View → Scheduled Tasks. The Task List window opens.
2. Select the task that you want to schedule and click Schedule task.
3. Click OK to close the window.

Sending HTML tables in email

You can send query results as HTML tables in the email when you work with scheduled tasks.

Make sure that the selected format of a letter is HTML and the table you want to add is saved in the HTML format too.

The inserted tables appear in the body of the letter in HTML format.

Adding an HTML table is available only for the following tasks of the scheduler:

- Exporting query to mail task. As a result of this task you get query result set in HTML format, which can be sent in the email.
- Sending mail resource manipulation task. In this task you can insert the necessary HTML tables created when performing other tasks.

To insert an HTML table in the body of an HTML letter:
1. In the Action Wizard window, click an ellipsis button (...) in the **Message** field. The Formula editor opens.

2. From the list of the functions expand the **Scheduler** and select **insertHTMLTable** or type the name of this function in the **Filter** field.

3. Double-click the function and specify its parameters.

4. Click **OK** to add the function. The specified table is added to the letter.

You can add *grid.html* table to the letter with the following parameters:

- When you work with export task, just add the function without any parameters
  
  =insertHTMLTable()

- When you work with resource manipulation task, you specify only one parameter, in case the exported HTML file has UTF-8 encoding.

  =insertHTMLTable("grid.html")

- You must specify both parameters if the encoding of the exported HTML file differs from UTF-8.

  =insertHTMLTable("grid.html", "UTF-16")

**Working with remote scheduled tasks**

With the help of remote scheduled tasks you can perform different operations such as running queries and exporting query results using a remote server.

To work with remote tasks follow the prerequisites:

1. You need to establish a web service connection to CorVu Web Pro to use it as a remote server.

2. You can open the objects only from CorVu repository when you select them and add new tasks.

To open the remote scheduled tasks select **View > Scheduled Tasks** and click **Remote tasks** tab. The remote tasks dialog opens and you can run the specified tasks, create new or edit them.
Chapter 10: Working with visual projects

Visual projects include visual reports, dashboards, and applications.

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.

Content developers can compile their visual projects to be read-only, protecting the data from other users.

Ad hoc reports
Ad hoc reports provide a flexible and user-friendly approach to creating reports, while making the overall design process more efficient. They include a scene with a canvas on which you create a report. A canvas can contain data objects such as charts, queries, and dynamarts.

Visual dashboards
Visual dashboards present interactive or persistent data obtained from querying multiple heterogeneous data sources across the enterprise. Visual dashboards present data in a scene format using a wide variety of graphics including charts, graphs, maps and user interface widgets. Data driven graphical objects can be easily linked so that user selections will trigger unique data displays. Content developers can create a Visual dashboard that can be viewed by multiple users with either CorVu Workstation Pro or CorVu Web Pro.

Content developers can compile their visual dashboards to be read-only, protecting the data from other users.

Visual applications
An extension of visual dashboards, visual applications present interactive or persistent data obtained from querying multiple heterogeneous data sources across the enterprise. Similar to visual dashboards, visual applications present data in a scene format using a wide variety of graphics including charts, graphs, maps and user interface widgets. Unlike visual dashboards, visual applications support concurrent development, allowing multiple designers to work on a single application at the same time. Additionally, visual applications provide ad hoc design capabilities on Tree and Canvas control widgets for scene objects. These controls are not available for visual dashboards. Data driven graphical objects can be easily linked so that user selections will trigger unique data displays. Content developers can create a visual application that can be viewed by multiple users with either CorVu Workstation Pro or CorVu Web Pro.

Content developers can compile their visual applications to be read-only, protecting the data from other users.

Visual Designer environment
You can quickly design visual reports, visual dashboards, and visual applications 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, dashboard, and application; and the Palette, Properties, Events, Layers, and Output views which support the visual designer editor and assist in creating visual reports, dashboards, and applications.
Visual project templates

Templates provide standardized, pre-formatted examples of visual projects (visual reports, applications, and dashboards). They are reusable components that facilitate the design process.

Visual project designers might use templates for a visual project as the entry-point into their project development process.

Templates help developers avoid repetitive development and promote effective reuse strategies. For example, if an enterprise has multiple visual applications that connect to a particular data source, rather than defining the connection properties in multiple visual applications, you could create a single template with the required connection. The new visual applications that you create can be based on this one template.

There are three template types for visual projects:

- **Predefined**
- **Local**
- **Repository**

**Predefined templates**
Predefined templates are included with CorVu Workstation Pro.
The layout of predefined templates cannot be changed.

Currently there is a single predefined default template included with CorVu Workstation Pro. The name of the default predefined template is Blank. The Blank template contains no structural elements.

Predefined templates provide a base-level functionality for a yet to be built business function. For example, you can use a predefined template as the basis for a connecting to a data source, running a standard set of queries or a customized set of charts and tables, connected with lists or other data driven controls.

**Local templates**
Local templates are those that have been created for your organization and saved locally.

Each template contains predefined content (such as company logos, standardized text, frequent graphical layouts).

The templates are listed by category for organizational purposes.

Typically, local templates are used by individual developers for their own purposes. Visual projects saved as local templates can not be shared via repository services.

**Repository templates**
Repository templates are those that have been created for your organization and saved in a shared repository.

Each template contains predefined content (such as company logos, standardized text, frequent graphical layouts).

The templates are listed by category for organizational purposes.

Users can edit the structural content of repository templates to suit their needs.

Typically, repository templates are used by multiple developers. Visual projects saved as repository templates are made available to multiple developers through repository services.
Creating template categories for visual projects

Use the Create a New [objectname] Template Category wizard to create a template category for your visual projects.

This procedure describes how to create a template category for your visual projects (visual reports, applications, and dashboards).

1. Launch the Create New [objectname] Template Category wizard.
   From the Menu bar, select File → New → Other → CorVu Objects → [objectname] Template Category and click Next.
2. Specify a name for the template category and (optionally), enter a comment in the Comment field.
3. Click Finish.

You have created a template category for your visual project.

Editing visual project connection data for Rocket CorVu

You can edit visual project connection data for opened visual projects.

To edit an opened visual project perform the following steps:

1. Click Edit in the list of the available visual projects.
2. In the Edit window you can change the login and password of the data source connections.
   The changes are saved on your device automatically and are applied when accessing the data source from your mobile device only.

Working with visual reports

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

Using CorVu Workstation Pro, 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 CorVu Web Pro, but you cannot design visual reports. The Visual Designer perspective is not available.

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

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 by using the New Visual Report wizard.
   ▪ Open an existing 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
Creating reports using the Visual Report wizard

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

The Create New Visual Report wizard offers a step by step, in-depth method of creating a new visual report, and is most useful if you are just learning the authoring process. 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:

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
   - Globals
   - Queries
   - Fixed Pages
   - Main Page

4. Create data source connections.

Visual reports are intended to be shared across the enterprise. In order to facilitate sharing and distributing visual reports, CorVu Workstation Pro does not tie the queries that are included in a visual report to a specific data source. Instead it ties 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 window or by dragging the data source from the Workspaces view to the Connections folder for the visual report in the Project Explorer view.

See Specifying data source connections for visual reports for more information

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.

See Specifying queries for visual data objects for more information

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 189 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 190 for more information on designing fixed pages.

8. Each time that you click Preview 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.


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:

   - Click 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.
   - Click 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.
   - Click 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 how and where to save the report template.
   You have the following options:
   - 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 can click Browse to search for a location.
In the Path field, specify the path to a local directory into which to save the template. For example:

- For Windows XP or earlier, the path might be:
  
  C:\Documents and Settings\[user_name]\Application Data\Rocket Software\CorVu Workstation Pro\Templates\VR Templates\VisualReport1

- For Windows Vista or later, the path might be:
  
  C:\Users\[user_name]\Application Data\Rocket Software\CorVu Workstation Pro\Templates\VR Templates\VisualReport1

**Note:** You can replace VisualReport1 with any name.

It is recommended that the path be to the local directory as this is where CorVu Workstation Pro will look for all local templates in order to display them to users when they create a new visual report.

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.

6. Click the **New Visual Report Template Categories** icon to create a category.


7. Specify a name for the report in the **Name** field. Optionally you can specify a comment in the **Comment** field.

8. Click **Finish**.

   The new visual report template category has been created. It is displayed in the Select parent entry list.

9. Specify a name for the report in the **Name** field. Optionally you can specify a comment in the **Comment** field.

10. Click **Finish**.

    You have created and saved a new visual report template. It will be available to users when they create a new visual report.

---

### 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 Set** section describes what appears at the beginning of the report.
- **Page Header Set** section describes what appears at the top of each page of the report.
- **Detail Header Set** section describes what appears as column headers to each detail line.
- **Detail Set** section describes how each detail line of data will be presented.
- **Detail Footer Set** section describes what appears on the break line for a detail section.
- **Page Footer Set** section describes what appears at the bottom of each page of the report.
- **Report Footer Set** 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, click the Design tab.

5. You can save the report to a file 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 in 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 in 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, click 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:
   - **hourly**: Select this option to specify that the same data will be used for an hour period. If you select this option, the visual report will run, data will be retrieved, and every time the report is run within one hour it will use the same data. When the one hour period expires, the next time the report is run, new data will be retrieved.
   - **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 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**: Select this option to have the data 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 in 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.
4. To convert an offline report back to an online report, open the offline report and select **Edit → Convert to Online Report**.

**Working with Ad hoc reports**

Ad hoc reports are canvas-based reports that graphically represent information retrieved from a database. An Ad hoc report includes a scene with a canvas.

Ad hoc reports provide an extempore and user-friendly way to design reports. You can create Ad hoc reports quickly by using the intuitive runtime editor designed to make working with data objects more efficient.

When creating a report, you can place multiple objects on the Canvas and manage the Canvas space to suit to your purposes. Typically, the Canvas represents what you believe to be the most appropriate set of data objects.

The Canvas Internal Tree displays the objects of all available workspaces and allows you to add any of them to your report.

The Canvas provides the following objects:

- Visual Queries
- Analytical Queries
- Dynamarts
- Compound Dynamarts
- U2 Queries
- U2 Dynamarts
- Drill-down Paths
- Tables

The Canvas with all the objects and layouts is a part of the Ad hoc report solution and is saved as a Built-in Canvas type directly to that report.

You can also wire objects in the Canvas for Ad hoc reports.
Creating Ad hoc reports

You can create an Ad hoc report and populate it with objects in the Ad hoc Reports runtime editor. An Ad hoc report includes a scene with a canvas. When creating an Ad hoc report, you can place multiple objects on the Canvas and manage the Canvas space according to your purposes.

To create an Ad hoc report perform the following procedure:

1. Open the Ad hoc Report editor in one of the following ways:
   - Select File → New → Other.
     - The New window opens.
   - Expand CorVu Objects and select Ad hoc Report
     - Click Next.
   - Click the New Ad hoc Report toolbar button ( ).
     - The Ad hoc report opens in the editor.
2. Drag your cursor on to the Canvas object. The Canvas changes color from white to gray and a cross appears in the center of the Canvas indicating the position of the first display.
3. Left-click anywhere in the grey display area.
   - The Canvas's Internal Tree is displayed.
4. Expand the folders in the Internal Tree to locate the object that you want to add to the Canvas.
   - Canvas supports the following objects:
     - Visual Queries
     - Analytical Queries
     - Dynamarts
     - Compound Dynamarts
     - U2 Queries
     - U2 Dynamarts
     - Drill-down Paths
     - Tables
5. Double-click the object that you want to insert into the Ad hoc report.
   - The default Display Mode for the object you selected is rendered graphically.
6. Click any of the widget icons listed on an object’s Display Mode and the image is rendered accordingly on the Canvas.

   **Note:** The widget icons that are displayed on the toolbar vary depending on the type of object you added to the Canvas.

The complete set of widgets allows the following actions:

- Switching display mode to one of available display modes
- Changing chart type to one of available 2d charts
- Changing chart type to one of available 2d with depth charts
- Changing chart type to one of available 3d charts
- Changing the query columns and aggregation being charted
- Deleting the query result from the canvas
- Adding a new chart
- Navigating forward and backward in Drill-down Paths.
You have created an Ad hoc report.

Related topics
- Managing Canvas operations using the Canvas toolbar, on page 210
- Wiring objects on the Canvas, on page 212

Publishing visual projects for mobile devices

You can publish your visual project as a Rocket CorVu application that runs on a mobile device. Published visual projects are run using the Flash rendering mode.

To publish a visual project as a Rocket CorVu application, perform the following procedure:

1. On the toolbar, select the Publish Visual Project icon ( ) to open the Publish Visual Project wizard. Or in the Workspaces, Project Explorer or Repository Explorer view, right-click the required project and select Publish Visual Project item.
2. Enter the Name and specify the Publication type.
   - Select Shared to make the published visual project available to all users of the repository.
   - Select Private to make the published visual project available to the current user only.
     This value applies to secure repositories.
3. Click Next.
4. Specify the user logins and passwords for data sources and save the information by selecting the Save check box.
   Note: If there are no available data sources in the visual projects, this button is disabled.
5. Click Finish.

The wizard closes and the visual project is published.

Visual projects of the Shared publication type are saved in the repository root in Published Visual Projects folder.

Visual projects of the Private publication type are saved in the home workspace in Published Visual Projects folder.

Working with visual dashboards and applications

A Visual dashboard is a type of visual project that presents interactive or persistent data to a wide range of users.

Visible dashboards can simultaneously draw data from multiple, heterogeneous data sources and present the data using a variety of graphs, charts, and graphical widgets. Using CorVu Workstation Pro, content developers can create a visual dashboard that can be viewed by multiple users with either CorVu Workstation Pro or CorVu Web Pro.

Note: You can view previously designed visual dashboards in CorVu Web Pro, 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:
Chapter 10: Working with visual projects

1. Activate a visual dashboard in the workstation editor window. You can activate a visual dashboard in the workstation editor window in one of the following ways:
   - Create a new visual dashboard using the Create New Visual Dashboard wizard.
   - Open an existing visual dashboard.

2. Open the Visual Designer perspective. The Visual Designer perspective presents an editor window and several views and menus that you will use when you work with visual dashboards. You can open the Visual Designer perspective by using either the Open perspective toolbar button or by selecting Window → Open Perspective → Visual Designer.

3. View the key components of the visual dashboard as listed in the Project Explorer view. Each visual dashboard is listed under the Visual Dashboards node in the Project Explorer tree. Each visual dashboard will have the following folders:
   - Connections
   - Globals
   - Queries
   - Scenes

4. Create data source connections.

   Visible dashboards are intended to be shared across the enterprise. In order to facilitate sharing and distributing visual dashboards, CorVu Workstation Pro 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 used to display your data. After creating a new visual dashboard, the system provides one default scene. You can choose to have only one scene in your visual dashboard or you can add additional scenes.

7. Insert objects into a scene.

   The objects that you can insert are displayed in the Palette view and are organized on individual palettes based on their type. From the Palette view select the objects that you want to insert into your scenes.

8. Modify the object's properties in the Properties view.

   By modifying an object's properties you set the values that determine how the object looks and behaves. You can use the Expression Designer to help you specify an object's properties.

9. Insert the layout objects

   In a visual dashboard scene layout objects display query results.

   When you insert a layout object the system prompts you to associate the object with a query that is contained in the Queries folder. You can add queries to your Queries folder at any time during the process of creating your dashboard.

   Layout objects can present data in many ways. For most layout objects, you display the results of multiple queries in a single layout. For example, you might create a single XY chart that displays sales figures derived from one query and spending figures derived from another query. You
can add multiple layout objects. When you place layout objects, you can pass query result information from a higher level layout object to a lower level layout object. Because you have the ability to pass this information, you can use the placed layout objects to display more detailed information that relates to a specific data value.

10. Use the **Events** view to assign the different navigation options.

   By assigning navigation options to scenes and objects, you set a path through the dashboard. One form of navigation is to define dependencies between dashboard control objects (such as combo or listbox objects) and data-driven layout objects (such as a grid or bar chart object) using the **Connectivity** tool from the **Palette** view.

11. Run the visual dashboard.

   Each time that you click the **Runtime** tab, you are actually running the visual dashboard object. By performing this step you can review how the results will display at run time. Think of this step as a test-run of your dashboard that allows your to improve upon the design before saving and deploying the dashboard.

12. Save the visual dashboard.

   When you have completed designing or editing the visual dashboard, you can save the dashboard and copy it to one or more workspaces for distribution to other users.

### Planning visual dashboards and visual applications

Before developing a visual dashboard or a visual application, plan your development and design process. One of the most important planning activities for designing visual dashboards or visual applications is known as **storyboarding**.

#### Using storyboards

Storyboards are visual aids to a design and a design process. Acting as a graphical organizer, storyboards help all members of your team pre-visualize the application or dashboard you intend to design.

The information in this section is intended as a general guideline to using storyboards. Proper use of storyboards during the planning phase facilitates the design process by clearly illustrating how the dashboard or visual application will look and function.

Storyboards help take the guesswork out of the actual design phase and can minimize potential development missteps and help you deploy the dashboard or visual application more efficiently. Additionally, a well designed dashboard or application has a higher likelihood for reuse in other areas of the enterprise.

You use a storyboard to design the basic features, functionality and presentation flow of the scenes and objects to be included in your visual dashboard or visual application.

A storyboard for a visual dashboard or application should specify the following:

- Data for each scene
- Scenes, drawn roughly
- Layouts for each scene
- Sequence of presentation
- Points where users can access further information
- Actions that result from user events
- Jumps between scenes
There are many types of tools, with varying level of complexity, that you can use to create storyboards. You can use a pencil and paper to sketch the design and flow. You can use a presentation tool, such as Microsoft PowerPoint, to create sample layouts, with supporting detail on notes pages. You can also put together Web pages to show actual jumps and navigation flow.

Use the following list of general topics as a guide to elements you should include in your storyboard / design:

**Displaying query results data**

You use layout objects or the List 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.

**Working with visual applications**

Visual applications extend visual dashboards functionality with concurrent development support and additional Flash-only objects that are not available with visual dashboards. Visual applications are run in Flash and HTML5 modes.

With visual application’s support for concurrent development and design, almost all objects inside an application are saved as a separate entry in the repository. These objects include queries, scenes with tree and canvas controls, pictures and global parameters. Whenever you open one of these objects for editing, it gets locked. All other application objects that depend on this object are also locked to protect integrity. As soon as a user saves an updated object and closes the editor, the object is unlocked and can be edited by another user.

Consider using visual applications instead of visual dashboards if your design process would benefit from concurrent development support and ad hoc design capabilities.

**Ad hoc design functionality**

CorVu Workstation Pro visual applications provide objects that support ad hoc design functionality. Ad hoc design promotes a more flexible and user-friendly approach to visual design, while making the overall design process more efficient.

The following visual application objects support ad hoc design:

- Canvas
- Tree
Trees are optional objects which provide a different path to ad hoc design.

**Canvas objects**

A Canvas provides the backdrop on which a designer creates his or her visual application. A Canvas contains the visual application data objects such as grids, charts and queries that are HTML and Flash-enabled.

The objects on a Canvas support *ad hoc design*. This allows the designer utilize the objects to address very specific or immediate design requirements.

The Canvas object has an *Internal Tree* comprised of two sources:

- Repository source
- Application source

Initially, the Canvas is blank. As part of the design process, the application designer:

- Sets properties for the Canvas
- Optionally populates the Canvas with objects

Canvas provides the following objects, all of which support Ad hoc design:

- Charts
- Visual Query
- Analytical Query
- Dynamart
- Compound Dynamart
- Optionally tests the appearance and usability of Canvas objects on the visual application in Runtime mode.

This might involve testing the options available from the object’s display mode.

- Saves the visual application and makes it available to users.

**Tree objects**

Tree objects provide an additional design mechanism for populating your canvas.

In Runtime mode, end-users or designers might use a Tree to locate objects such as queries, charts and dynamarts to drop onto the Canvas.

You can also set Canvas configuration parameters from the Tree object.

The Tree object has three sources:

- Repository source
- Application source
- Query source

The Query source allows the developer to specify a Query whose result set will populate the Tree with a TreeChart layout, where data in displays as tree structure that shows a cascading set of parent-child relationships. The TreeChart layout fetches all of the data for the nodes and branches at once, where as the Application Object Tree fetches the data for the expanded current node / branch only.

*Note:* The Repository source and Application source in a Tree serve the same purpose as they do in the Canvas.
The Tree object provides some functionality not available in the Canvas object Internal Tree. For example, the Tree object includes the Tree.Selection feature and provides additional properties (such as Label.text used in a title) that can be referenced by other visual application objects.

Concurrent development

Concurrent development is a process model and methodology that can be applied to a variety of disciplines, including project management, product management and software design.

Concurrent development is based on the concept of parallelization of tasks. When applied to software design, concurrent development allows members of a design team to perform development tasks concurrently (as opposed to sequentially), which enables your team to develop and deploy applications in less time.

CorVu Workstation Pro's visual application design model supports concurrent development.

Unlike visual dashboards, visual applications can be edited by several designers simultaneously because the product automatically provides for the integrity of the visual application.

Canvas types and layouts

Each Canvas type maintains a specific layout that is dependent upon the objects that it contains. The layout of a Canvas is determined by the data objects placed on it and the size and location of the Canvas on the application.

There are three types of Canvases:

- **Built-in**
  A Built-in Canvas is stored inside the application and is created while designing the application with Canvas

- **Shared**
  A shared Canvas is stored in unified shared space in repository

- **Personal**
  A shared Canvas is stored in the user's home workspace.

Each object on the Canvas includes the following attributes:

- **key**
- **container location and size**
- **object layout (each object has to be able to save it's current layout).**

**Built-in Canvas**

A Built-in Canvas type can be saved to the visual application directly when that visual application is open in the editor and is in Design mode.

A Built-in Canvas contains the objects and layouts that are part of the visual application solution, as envisioned and authored by the application designer. Typically, a Built-in Canvas represents what the application designer believes to be the most user-appropriate set of data objects, as well as the layout of those objects.

The designer of the visual application can also set the default type to Built-in at design time. This default type is loaded for the end users. If the designer of the visual application does not set the default type, the end users will see the last canvas state that was used by application designer.

Users can adopt a Built-in Canvas, using it as a building block for their visual application, tailoring it to suit the needs of their visual application and then saving it under a new name. The tailored Canvas would be accessible, alongside those that come as part of the application solution (i.e., Built-in Canvases).
Shared Canvas

A Shared Canvas is a Canvas type that is saved to a shared repository when the visual application on which the canvas resides is open in **Runtime mode**.

A Shared Canvas is accessible and usable by all users of the repository.

Personal Canvas

A Personal Canvas is a Canvas that is accessed by users while the application is in **Runtime** mode and saved to an existing personal workspace.

If a personal workspace does not exist, users will not be able to save the Canvas as a Personal canvas. Each user can also set default type for current canvas that will be loaded for this user in the future.

Creating visual applications

You can create visual applications by using the Create New Visual Application wizard, or you can create visual applications from existing visual dashboards.

Creating visual applications from visual dashboards

You can create visual applications from existing visual dashboards. A visual application created in this manner assumes all features and functions of the dashboard from which it was created, and will also support those features and functions available with visual applications only (for example, concurrent development and additional flash-only based widgets). The visual dashboard used as the source for creating the visual application will remain on your system for use.

Understand how visual applications extend visual dashboards.

This procedure describes how to create visual applications from existing visual dashboards.

To create a visual application from an existing visual dashboard, perform the following steps:

1. Open the **Visual Designer** perspective in one of the following ways:
   - From the menu bar, select Window → Open Perspective → Visual Designer
   - Click the Open Perspective toolbar button ( ) and select Visual Designer.

2. In the **Workspaces** view, navigate to the visual dashboard from which you want to create a visual application. Right-click the dashboard, and select **Edit**.
   
   The dashboard opens in **Design** mode in the viewer.

3. From the **Project Explorer** view, expand Visual Dashboards and right-click the dashboard and select Convert to Visual Application.

   **Note:** The system prompts you to save any changes made to the visual dashboard from which you are creating the visual application.

   **Note:** The Convert to Visual Application operation *does not remove or overwrite the existing dashboard object*. The visual dashboard remains as its own separate object after you complete the conversion.

4. In the **Project Explorer** view, expand the Applications object to display the new visual application created from the existing Visual Dashboards object.

5. Save the visual application.

   You can save the visual application at a repository or a file.

   To save the visual application to a repository, perform the following steps:

   1. In the **Project Explorer** view, expand visual applications.
2. Right-click the visual application created from the visual dashboard and select **Save At. Repository**
3. Set values in the Save at Repository window and click **OK**

To save the visual application to a file, perform the following steps:
1. In the **Project Explorer** view, expand visual applications.
2. Right-click the visual application created from the visual dashboard and select **Save At. File**
3. Set the workspace directory path in the Save at File window and click **OK**.

If saved to a repository, the application object (query, scene, picture, global parameter) is locked as soon as it is opened in the editor. If another user tries to open the same object, he is presented with a locked application error.

You have successfully created a visual application from an existing visual dashboard.

---

**Note:** The visual application is different in structure than the dashboard from which it was created. Dashboards have a flat structure; that is, all queries and scenes are located at the same level. Visual applications have a tree-like structure; that is, users can create folders in Scene and Query nodes and place their objects in them. You are permitted to have the multiple instances of a scene in a visual application, as long as the scenes are located in different folders.

As an application designer, you can continue to design functionality into your visual application. For example, you might add canvas or tree objects to your application.

As a runtime consumer of the visual application, you can also perform "design-on-the-fly" operations to the visual application.

---

**Creating visual applications using the New Visual Application wizard**

You can use the Create New Visual Application wizard to help you create new visual applications.

You understand how the visual application you want to create maps to one of the template types (Predefined, Local and Repository).

This task describes how to use the Create New Visual Application to help you create new visual applications.

The visual applications that you create are modeled on template types. Each template type (predefined, local or repository), has a specific usage scenario.

1. **Launch the Create New Visual Application wizard.**
   - You can launch the wizard from the **Visual Designer** perspective by using any of the following methods:
     - Right-click the **Visual Applications** folder in the **Project Explorer** view and select **New Visual Application** from the context menu.
     - Select **File → New → Other → CorVu Objects → Visual Application** and click **Next**.
     - Double-click the **Visual Applications** folder in the **Project Explorer** view.
2. **On the Create New Visual Application page,** type a unique name for your visual application in the **Application name** field.
3. **From the Select template section,** click the tab that best describes the type of template on which you will model your visual application and select the template.

   Template types include the following:
   - **Predefined** templates.
Predefined templates are packaged with the application.

- **Local** templates.
  
  Typically, you would select **Local** templates if the application you are creating will be for personal use (not shared in a repository)

- **Repository** templates.
  
  Typically, you would select **Repository** if the application will be shared in a repository.

4. **Click Finish.**

A folder with the application name is added to the **Applications** node in the **Project Explorer** view.

For each new visual application a node is created for the **Connections**, **Queries**, and **Globals** that will be used for this application. In addition, a **Scenes** node is added with a default first scene.

The editor window opens with the first scene.

You have successfully created a visual application based on one of the template types.

Add features and functions to your visual application.

**Working in Design mode**

Design mode refers to a specific state of the visual application editor. When the editor is in Design mode, the user who is designing the visual application can perform functions that are specific to that mode.

While working in Design mode, the visual application designer can perform the following operations:

- Place a Canvas on the visual application
- Set Canvas properties

**Working in Runtime mode**

Runtime mode refers to a specific state of the visual application editor. When the editor is in Runtime mode, the visual application designer can perform functions that are specific to that mode.

Most of the design-based tasks associated with the Canvas are performed while the visual application editor is in Runtime mode.

**Note:** The way by which users open the visual application has an affect on the types of operations the user can perform.

While working in Runtime mode, the visual application designer can perform the following design operations:

- Add objects that are supported by Canvas, to the Canvas
- Delete objects from the Canvas
- Resize objects on the Canvas
- Move objects within the boundaries of the Canvas
- Save the Canvas layout
- Switch between Canvas layouts
- Rename and delete Canvas layouts
- Set a Canvas type as the default layout
- Perform actions supported by, and specific to the objects on the Canvas

**Designing your visual application**

Use the Visual Designer to design your Visual Application. At a high-level, designing a Visual Application involves adding a Canvas to the application, setting the properties for the Canvas, setting
the layout of the Canvas, adding objects to the Canvas and then managing the visual design of the objects that you added to the Canvas.

The typical workflow for designing your visual application is illustrated in the following diagram:

**Adding a Tree to your visual application**

Use the workbench in CorVu Workstation Pro to add a Tree object to your visual application. The Tree object provides an additional design mechanism that you can use to locate and add objects to your Canvas.

You have created a visual application, either from an existing visual dashboard, or by using the New Visual Application wizard.

The steps in this procedure describe how to add a Tree to your visual application.

Adding a Tree object to your visual application is an optional design procedure.

The Canvas object has an internal Tree that provides the same functionality provided by the Tree object.

1. Make sure the visual application is displaying in the editor area of the workbench and that the application is in **Design** mode.

2. Click the **Show Palette** icon to show the list of folders that hold the design elements of your visual application.

   **Note:** The **Show Palette** icon is located in the upper-right corner of the editor area:

   ![Show Palette icon](image)

   From the Palette, locate the folder named **Visual Application Objects** and expand it.

   The expanded view of the **Visual Application Objects** folder displays a list of objects that you can incorporate into your visual application.

4. Double-click **Tree** to add it to your visual application.

   By default, the Tree object is named **Tree1** and appears as *grey box with a dotted line perimeter* in the editor area:
You have successfully added a Tree object to your visual application.
The Tree object provides a view into the hierarchical structure of data visualization objects and data transformation objects.
You can view the Tree’s hierarchical structure by switching the editor to Runtime mode.

While in Design mode, position the Tree object adjacent to the Canvas object.
Switch to **Runtime** mode and drag the visual application objects from the Tree onto the Canvas.

---

**Adding a Canvas to your visual application**

Use the workbench in CorVu Workstation Pro to add a Canvas to your visual application.

You have created a visual application, either from an existing visual dashboard, or by using the New Visual Application wizard.

The steps in this procedure describe how to add a Canvas to your visual application.

1. Make sure the visual application is displaying in the editor area of the workbench and that the application is in **Design** mode.
2. Click the **Show Palette** icon to show the list of nodes that hold the design elements of your visual application.

---

**Note:** The **Show Palette** icon is located in the upper-right corner of the editor area:
3. From the Palette, locate the **Visual Application Objects** node and expand it.

   The expanded view of the **Visual Application Objects** node shows a list of objects that you can incorporate into your visual application.

4. Double-click **Canvas** to add it to your visual application.

   By default, the Canvas object is named **Canvas1** and appears as **grey box with a dotted line perimeter** in the editor area:

   ![Canvas1](image)

   You have successfully added a Canvas object to your visual application.

   The Canvas object has an internal tree. The repository subtree or queries subtree of data visualization objects and data transformation objects located in visual application depends on the **Source** property of the internal tree.

   The Canvas object has an internal tree that represents (in its entirety) the hierarchical structure of data visualization objects and data transformation objects.

   You can view the Canvas's internal tree by switching the editor to **Runtime** mode and clicking the green plus (+) sign.

   Keep in mind that Canvas supports only a subset of data visualization objects and data transformation objects.

   Position and size the Canvas as needed and then set the configuration properties.

### Placing and sizing the Canvas

While in Design mode you decide where to locate the Canvas (within the boundaries of your visual application) and you can size the area of the Canvas to meet any spatial design requirements.

You have added a Canvas to your visual application.

The visual application editor is in **Design** mode.

The following procedure describes how to position and size a Canvas object in your visual application.

1. Place your cursor inside the Canvas object and drag it to the desired location on your visual application.
2. Place your cursor on the perimeter of the Canvas object and drag outward (to increase the size of the Canvas) or inward (to decrease the size of the Canvas).
3. To review the location and size of the Canvas object in relation to the visual application boundaries, toggle to Runtime mode by clicking the **Runtime** tab in the visual application editor.

You have placed the Canvas on your visual application and you have sized it to the desired size.
You are now ready to set properties for your Canvas object.

Setting properties for the Canvas

You can set the properties for a Canvas by using the Properties view.

You have created a visual application and you have added a Canvas to that application.

You have placed the Canvas to the desired location on the visual application and you have sized the Canvas object appropriately.

The procedure in this task describes how to set the properties for your Canvas object.

1. With the visual application in Design mode, select the Canvas.
2. In the Properties view, set the values for the Canvas.

Common properties, (those properties that are standard for all visual objects) include the following:

Name
   The name of the Canvas object

Location
   The value you enter determines the location and placement of the Canvas object on the visual application.

ToolTipText
   The value you enter is the "hover text" to display when the user places his or her cursor on the Canvas object at runtime.

Frame
   Set properties to determine how the Canvas frame will display at Runtime.

ScrollBars
   Specify scroll bars visibility on the Canvas object. Valid options are:
   ▪ None (0)
   ▪ Horizontal (1)
   ▪ Vertical (2)
   ▪ Both (3)
   ▪ Auto (4)

Width
   The value you enter determines in inches the width of the Canvas object on the visual application.

Height
   The value you enter determines in inches the width of the Canvas object on the visual application.

CurrentLayout
   The value you enter determines the layout that is active on the Canvas. If the specified layout cannot be found, a previously specified layout remains active.

Properties specific to Canvas objects include the following:

InternalTree
   Includes the following sub-properties:
   ▪ Font
     Use the font property settings to configure font type and appearance.
   ▪ Icon Style
Select **Triangular** or **Rectangular**

- **Source**

Select either **Repository** or **Application**.

If you select **Repository**, all objects that exist under the RootPath populate the internal tree.

If you select **Application** all objects under the Query node of the visual application to populate the tree.

- **RootPath**

  Specifies the starting node in the explorer tree if you selected **Repository** as the value for the Source property.

**Note:** If you set the Source property to **Repository**, all objects in the folders under the RootPath will be displayed in the tree. However, you can only use those object types that are supported by Canvas. These object types include the following:

  - Charts
  - Visual Query
  - Analytical Query
  - Dynamart
  - Compound Dynamart

To suppress unsupported document types from displaying in the tree, you can create a folder in the Workspace and populate it with links to the supported object types by using Copy/Paste Link feature. Alternately you can include all the query documents of interest in the visual application and create folder/directory structure with the Visual Application to populate the Tree if you choose **Application** as the Source.

You have set the Canvas properties.

You are now ready to add objects to the Canvas.

**Adding objects to the Canvas in the Flash mode**

You can add objects to your Canvas of your visual application in the Flash mode.

To add objects to the Canvas in the Flash mode.

1. Click the **Runtime** tab in the editor area to place the visual application in **Runtime** mode.
2. Drag your cursor on to the Canvas object.

The Canvas changes color from white to gray and there is a cross in the center of the Canvas indicating the position of the first display.
3. Click the grey display area to open the Canvas Internal Tree.
4. Select an object and click **OK** to run that object and display the result in the default format.

Currently, Canvas supports the following objects:

- Drill-down Paths
- Visual Query
- Analytical Query
- Dynamart
- Compound Dynamart

**Note:** You can add queries with prompts to the Canvas. Use local or global parameters to manage prompts automatically. Insert a parameter to the visual project, make sure that it has the same name as one of the prompted variables used in the query, and then set the default value. This value is set for the prompt variable when you add the query to the Canvas.

You have added objects to the Canvas.

After adding an object to the Canvas, the default *Display Mode* for that object is rendered graphically. For example:

**Display Mode for chart object**

![Display Mode for chart object](image)

**Display Mode for query object**

![Display Mode for query object](image)
Click any of the widgets listed on an object's Display Mode and the image is rendered accordingly on the Canvas.

The buttons on the toolbar vary depending on the type of query. The complete set of widgets allow for the following actions:

- Switching display mode from chart to grid.
- Changing chart type to 2D:
  - Area
  - Bar
  - Bubble
  - Column
  - Grid
  - Line
  - Pie
  - Scatter
  - Speedo
- Changing chart type to 2D With Depth:
  - Area (except stacked Area charts)
  - Bar
  - Column
  - Pie
- Changing chart type to 3D:
  - Area
  - Column
  - Line
- Changing the query columns and aggregation being charted.

Numeric columns can have an aggregation value of:

- None
- Count
- Maximum
- Minimum
- Sum
- Average
- First value
- Last value
- Standard deviation

Character columns can have an aggregation value of:

- Count
- Deleting the query result from the canvas.

Managing Canvas operations using the Canvas toolbar
The Canvas toolbar provides actions for managing the Canvas.

Canvas toolbar options
A toolbar displays whenever you add an object to the Canvas:

- In the Flash mode:
Managing Canvas operations using the Canvas toolbar

- In the HTML5 mode:

Each option on the toolbar is described in the following table:

**Table 14: Toolbar options**

<table>
<thead>
<tr>
<th>Toolbar widget</th>
<th>Flash mode</th>
<th>HTML5 mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="select.png" alt="Select" /></td>
<td><img src="select.png" alt="Select" /></td>
<td>Select <strong>New</strong> button to create an empty layout.</td>
<td></td>
</tr>
<tr>
<td><img src="newlayout.png" alt="New Layout" /></td>
<td><img src="select.png" alt="New Layout" /></td>
<td>Select <strong>New Layout</strong> options from the list for managing existing Canvas layout.</td>
<td></td>
</tr>
<tr>
<td><img src="saveas.png" alt="Save As" /></td>
<td><img src="select.png" alt="Save As" /></td>
<td>Click the <strong>Save As</strong> button to save the Canvas as a Built-in, Shared or Personal Canvas type.</td>
<td></td>
</tr>
<tr>
<td><img src="save.png" alt="Save" /></td>
<td><img src="select.png" alt="Save" /></td>
<td>Click the <strong>Save</strong> button to save the Canvas.</td>
<td></td>
</tr>
<tr>
<td><img src="opendefault.png" alt="Open default" /></td>
<td><img src="select.png" alt="Open default" /></td>
<td>There is no such option in the HTML5 mode.</td>
<td></td>
</tr>
<tr>
<td><img src="clear.png" alt="Clear" /></td>
<td><img src="select.png" alt="Clear" /></td>
<td>Click the <strong>Open default</strong> button to switch to default Canvas layout.</td>
<td></td>
</tr>
<tr>
<td><img src="saveasdefault.png" alt="Save as Default" /></td>
<td><img src="select.png" alt="Save as Default" /></td>
<td>Clear current Canvas layout. Objects will be removed.</td>
<td></td>
</tr>
<tr>
<td><img src="maximize.png" alt="Maximize" /></td>
<td><img src="select.png" alt="Maximize" /></td>
<td>Click the <strong>Save as Default</strong> button to make the current layout.</td>
<td></td>
</tr>
<tr>
<td><img src="pinunpin.png" alt="Pin/Unpin" /></td>
<td><img src="select.png" alt="Pin/Unpin" /></td>
<td>Click the <strong>Maximize</strong> button to maximize the Canvas.</td>
<td></td>
</tr>
<tr>
<td><img src="pinunpin.png" alt="Pin/Unpin" /></td>
<td><img src="select.png" alt="Pin/Unpin" /></td>
<td>There is no such option in the HTML5 mode.</td>
<td></td>
</tr>
<tr>
<td><img src="wiring.png" alt="Wiring" /></td>
<td><img src="select.png" alt="Wiring" /></td>
<td>Click <strong>Pin/Unpin</strong> button to pin or unpin the Canvas toolbar.</td>
<td></td>
</tr>
<tr>
<td><img src="wiring.png" alt="Wiring" /></td>
<td><img src="select.png" alt="Wiring" /></td>
<td>Click the <strong>Wiring</strong> or <strong>Toggle Wiring Lines</strong> button to activate the wiring mode.</td>
<td></td>
</tr>
</tbody>
</table>
Chapter 10: Working with visual projects

<table>
<thead>
<tr>
<th>Toolbar widget</th>
<th>Flash mode</th>
<th>HTML5 mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Toolbar widget" /></td>
<td></td>
<td>There is no such option in the HTML5 mode.</td>
<td>Click the Export button to open the export pane and select the available export options:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><img src="image" alt="PDF" /> - Export the current content of the Canvas to PDF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><img src="image" alt="PNG" /> - Export the current content of the Canvas to PNG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><img src="image" alt="Print" /> - Print the current content of the Canvas</td>
</tr>
</tbody>
</table>

Wiring objects on the Canvas

Use the object Wiring Mode to connect objects on the Canvas.

Wiring objects defines the relationship and dependencies between objects.

The steps in this procedure describe how to wire objects on the Canvas. Object wiring is supported for both visual applications and ad hoc reports.

To wire objects, perform the following procedure:

1. Click the **Wiring** icon (or ) on the Canvas Toolbar to switch the Canvas to the **Wiring Mode**.

2. On the Canvas, select the source object and the target object.

   The source object is highlighted with blue and the target object is highlighted with green.

   When you wire objects, the system checks for possibility of creating a new link. You can select any data object that supports wiring on your Canvas as the source object. However, once the source object has been set, only those objects that are valid targets to the source are available (highlighted) for wiring.

   **Note:** Any data object that supports wiring can be the source object and the target object at the same time. The number of links is limited only by the number of available columns. Each column can be the source in several links and the target in the only one.

   **Note:** You can also wire objects on the Canvas by using drag-and-drop method. When you drag an object the **Wiring** icon is displayed over the canvas.

3. In the Create new link window, specify the source and destination columns.

   From the **Source** drop-down list, select a column of the source object.

   From the **Destination** drop-down list, select the column of the target object

   If the queries already contain specified output and input columns, these columns are displayed and automatically selected. Otherwise, all the columns from the result set are displayed.

4. Click **Create** to wire the objects to one another and to create the link.
You have wired objects on the Canvas.

In objects that are linked, the target object displays the data according to the selected element of the source object.

When you select another element, the data that displays changes. The selected element is highlighted. The second click on the same element deselects it. In this case, the target object displays the result as if all the elements of the source object were selected.

Editing linked objects

Use the **Edit Link** function to make changes to the source and target objects or to remove the link relationship between the objects.

To edit linked objects, perform the following procedure:

1. From the Canvas, select the source object and the target object by clicking them.
   The Edit link window opens.
2. In the Edit link window you can edit the source columns of the source object and destination columns of the target object in the corresponding fields.
3. To remove the linkage between the two objects, click **Delete**.
   The link between the objects is removed and the Edit link window closes.
4. Click **OK** to save the changes and close the Edit link window.
5. Click **Cancel** to discard the changes and close the Edit link window.

Managing input and output columns for the query

You can specify available input and output columns for the current query. These columns will be used for wiring objects on a canvas when creating visual applications or ad hoc reports.

The steps in this procedure describe how to specify input and output columns for a query.

1. Select **Query > Wiring Settings** to open the Wiring Settings window.
2. In the Wiring Settings window the list of available columns is displayed. To specify input and output columns select the corresponding check boxes.
3. After you have specified the input and output columns, click **OK** to apply the changes and close the Wiring Settings window.

Creating visual dashboards using the Visual Dashboard wizard

You can create a visual dashboard by using the Create New Visual Dashboard wizard.

To create a visual dashboard, perform the following steps:

1. Open the Create New Visual Dashboard wizard on one of the following ways:
   - From the **Visual Designer** perspective, right-click the **Visual Dashboards** folder in the **Project Explorer** view. Select **New Visual Dashboard** from the context menu. The Create New Visual Dashboard wizard opens.
   - From the **Visual Designer** perspective, double-click the **Visual Dashboards** folder in the **Project Explorer** view. The Create New Visual Dashboard wizard opens.
   - Click the **New Visual Dashboard** toolbar button ( )
2. Type a unique name for your dashboard in the **Dashboard name** field.
3. Select the template that will be used for the new visual dashboard in one of the following ways:
   - Click 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.
   - Click 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 page displays only if there are locally saved templates in your Templates directory.
   
   - Click **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 visual dashboard template, perform the following steps:

1. **Determine the visual dashboard to be used as the basis for the template.**
   
   You can create a visual dashboard and include the common elements or open an existing visual dashboard that will serve as the basis of the template.

2. **Select File → Save As.** The Select a wizard window opens.

   On the Select a wizard window, select which wizard to use, check 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 3.](#)
   - **Save to Repository**
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.

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 source 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, click Create a new query.
   b. If you are creating a new analytical query, click Create a new analytical query.
   c. If the query exists, click Attach to an existing query. The Select query field becomes active. Specify the name of the query and where it resides in the Select Query field. You can click Browse to search for the query.
   d. If you would like to insert a static query, click From file.

      - Click 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.
      - Click 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 click Browse to search for the query.

      Note: You can open CSV, DBF, IXF, TXT, XML, SQY, and TAB files and use them as queries. You can edit the imported result set by filtering, adding calculated columns, applying grouping and aggregation. When you save the visual project, these files can be converted to dynamarts and saved with the visual project. All modifications will be preserved.

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 report in the Project Explorer view. When you drop the query, the Edit Query window opens where you select the data source connection information for the query.
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, CorVu Workstation Pro 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 on the Connections node. Select Insert Connection from the context menu. The Insert Connection window opens.
   - From the Project Explorer view, double-click on the Connections node. The Insert Connection window opens.
   - Click on a query from the Workspaces or Repository Explorer view. With the mouse button depressed, drag the query to the Queries node in the Project Explorer view. A new data source connection is automatically added to the Connections folder of the Project Explorer view.

2. From the list of available data sources, select the data source that will be associated with this connection information alias.

3. Specify a unique name for this connection information in the Connection name field.

4. Click Finish. The Project Explorer view lists the new connection information alias under the Connections node in the tree.

Note: You can also set up a connection information alias for a data source by dragging the data source from the Workspaces view to the Connections folder for the visual dashboard in the Project Explorer view. The connection information alias is added with the name ConnectionN. You can edit the name of the connection information alias by right-clicking 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.

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 source** 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:
   - If you are creating a new query, click **Create a new query**.
   - If you are creating a new analytical query, click **Create a new analytical query**.
   - If the query exists, click **Attach to an existing query**. The Select query field becomes active. Specify the name of the query and where it resides in the Select Query field. You can click **Browse** to search for the query.
   - If you would like to insert a static query, click **From file**.
     - Click **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.
     - Click **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 click **Browse** to search for the query.

   **Note:** You can open CSV, DBF, IXF, TXT, XML, SQY, and TAB files and use them as queries. You can edit the imported result set by filtering, adding calculated columns, applying grouping and aggregation. When you save the visual project, these files can be converted to dynamarts and saved with the visual project. All modifications will be preserved.

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 report in the **Project Explorer** view. When you drop the query, the Edit Query window opens where you select the data source connection information for the query.

---

**Working with scenes**

A visual dashboard consists of one or more scenes that display data to users. You can add, delete, or rename scenes in your visual dashboard.

Upon creating a new visual dashboard, you are given one default scene. You can add more scenes to the visual dashboard.
Within each scene of your dashboard you add objects to display data and information to your users and you 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

To add, delete, or rename a scene, perform the following steps:

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, perform 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 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.

You have worked with the scenes in your visual dashboard, adding scenes, renaming scenes or deleting scenes.
Creating scene parameters

Using parameters, you can alter the value of an object's properties in the scene. Scene parameters are available for the scene in which they are defined.

Each scene that you create has a Locals folder that is added to the Project Explorer view 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 and select Insert Parameter from the context menu. The Insert Parameter window opens.
   - From the Project Explorer view, double-click the Locals node for the scene to which you want to add a parameter and select Insert 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 → 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 default data type is Integer.
   - The data type describes the type of data to be included in the parameter. The following Data type options are available:
     - 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. If you want to add a description, type 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

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

Use the Drilldown wizard to create the click event, identify the destination scene, and map 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).

The Drilldown wizard provides an automated method for creating the click event, identifying the destination scene, and mapping the required scene parameters.

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. Click Next.

3. In the first window, select the query result data columns whose values you would like to pass through to the 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 to display second page of the Drilldown wizard

   Select from the Location radio group to determine whether the drill down 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.

   Click 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 to display 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:
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 window, 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 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 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, see Units of Measure.

**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, see the User’s Reference documentation in the product help.

5. Specify the zoom level of the viewpoint in the **Zoom level** field.

   The default value is 100%.

6. Click **Finish**.

   The New Viewpoint window closes.

The specified viewpoint appears in the **Viewpoints** folder of the **Project Explorer** view.

---

**Modifying viewpoints**

You can modify the viewpoints.

To modify a viewpoint, perform the following steps:

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** text boxes.

3. Specify a new zoom level in the **Zoom level** field.

4. Click **OK**.

   You have modified the viewpoint.

**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.

---

**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.
- **BorderStyle.** Sets the style of the border around the embedded scene object. You can choose a border style, or choose no border.

**Inserting a simple embedded scene object**

A simple embedded scene connects two scenes.

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 a visual dashboard or visual application scene, perform the following step:

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 specify a project.
   - If you want to use a scene from the current project make sure that the Another Visual Project check box is clear.
5. Select the name of the scene that you will embed from the **Embedded scene** list box.
6. Select the **Enter 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 **Enter embedded scene when clicked with the mouse** check box should be cleared if you want to specify a different event action for the embedded scene object.
7. Specify the location and zoom level of the embedded scene in the **X location**, **Y location**, and **Zoom** fields.
8. Click **Finish**. The EmbeddedScene Wizard closes and the new embedded scene is displayed in the editor window.
\textbf{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.

\textbf{Note:} You can use a scene from another Visual Project. If you want to open a scene from another Visual Project select 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. Select Insert $\rightarrow$ 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 want 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 Reset to Default 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 $\rightarrow$ 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 Enter 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 Enter 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 \textit{X location}, \textit{Y location}, and \textit{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 in the Value field to open the Expression Designer. The Expression Designer will help you specify the value for the parameter. If you used a combo box
to accept user input then you must select a property value of the combo box to include in the parameter value.

**Setting up event action navigation for your visual application or visual dashboard**

Setting up event action navigation involves assigning an event to a scene object or to any graphic object that can be inserted in a scene. The *event-to-object* relationship expresses how the object responds to a user’s action as the user navigates the visual application or visual dashboard.

Events are actions that users of the visual project undertake. 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 to be triggered when the event occurs. Additional actions might include jumping to a new scene, setting a parameter value or issuing a message. As a user navigates the visual dashboard or visual application, events that are assigned to an object take place (such as clicking a button object), and any additional actions defined for the event are triggered.

The following tasks describe process of assigning events to objects and how to specify the actions triggered when the event takes place:

**Assigning events and setting up event actions**

Use the **Events** view to assign an event to an object and to define the actions to be associated with the event.

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.

The event actions that you assign to objects when building a visual dashboard or visual application provide the scene interactivity and navigation at run time. As a user navigates your visual dashboard or visual application, the mouse actions that you define using this procedure, 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, perform the following steps:

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 New Action** toolbar button. 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
   - Navigate to Embedded Scene
   - Execute SQL statement
   - Execute procedure
   - Set values
   - Navigate back
   - Print scene
   - Execute shell command
Assigning jump events to objects in your visual dashboard or visual application

- Show message
- Execute JavaScript
- Export to Excel
- Set focus
- Reload scene
- End the session
- Export to PDF
- Send an email
- Send Event to Embedded Scene
- Invalidate Query Cache
- Refresh Component
- Add Launch LOB event
- Open URL
- Save Cookie
- Load Cookie

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 in 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 click **Insert after selected item** and **Insert before selected item** to position new actions in the list. You can click **Move Action Up in List** and **Move Action Down in List** toolbar buttons to move the actions in the list to new positions. You can click the **Copy Action** and **Paste Action** toolbar buttons 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** toolbar button. 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** toolbar button. 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 is shown in bold font 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 or visual application

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

You must create both the source and the destination dashboard or application 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.

To assign and define a jump event for an object in your visual dashboard or visual application, 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 application or to another visual project. Select This project to specify that the destination for the jump will be to a scene or viewpoint within the current dashboard or application. Select Another visual project to specify that the destination for the jump will be another visual dashboard, application, or visual report.
6. If you selected This project, the Scene and Viewpoint fields become available. From the Scene list box, select the destination scene name. From the Viewpoint list box, select a viewpoint. If there are no viewpoints in the scene, the default is used.
7. If you selected Another visual project, the Visual project field becomes available. Specify the name of the destination visual dashboard, application, or report in the Visual project field. Click the ellipses (...) to search for and open a dashboard, application, or report from a repository or 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, application, or report that you select has parameters, click Retrieve Parameters.

   Note: The parameters that must be set when the event occurs are listed.

9. Specify a value for each of the parameters in the Value cell for each parameter.
10. From the Open in area select the required option:
    a. Same tab to open another visual project in the current tab.
    b. New tab to open another visual project in a new tab in CorVu Workstation Pro or in a new browser tab in CorVu Web Pro.
    c. New window to open another visual project in a new tab in CorVu Workstation Pro or in a new browser window in CorVu Web Pro.
12. In the [objectname] Behavior window you can add conditional expressions to the jump to a new location action. You can define multiple jump to new location actions each with a unique condition. Once a jump to a new location does occur, any subsequent actions defined for the object will not occur.
13. In 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 Assigned the jump events to the objects in your visual dashboard or visual application.
Setting global and scene parameter values

Use the **Set values** action to set global and scene parameter values.

You can set global and scene parameters values to coincide with an event action. The parameter 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, 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 **Set values** from the list of available actions.
4. Click **Next**.
The Set values action parameters page opens.
5. Click 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 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 **Add New Entry**. 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 **Apply** 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 different event actions. For more information about event actions, see **Assigning events and setting up event actions**, on page 228 topic.

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. This **Click** event is set up automatically when you create the **EmbeddedScene** object.

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, perform the following steps:

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.

You have added a Navigate to embedded scene event action to the object.

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, perform the following steps:

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 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 and the Add New Action window closes.
5. In 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 the Navigate back action to the object.

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 to an object, perform the following steps:

   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:

   \[
   \text{Update MyTable Set Shipped = 1 Where CustomerID = [value of g\_nCustID global variable]}
   \]

   You would enter the following text in the SQL Statement field:

   \[\text{"Update MyTable Set Shipped = 1 Where CustomerID = " + g\_nCustID}\]

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 or 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. In 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 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 Print scene event action in an object, perform the following steps:

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 window 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. In 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.

You have added a Print scene event action to an object.

### 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, perform the following steps:

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 window 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 from the Take file path from parameter drop-down lists.

13. Click Finish. The Send Email action is added. The Add New Action window 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.

You have added a Send Email event action to an object.

**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.

To add a Send Event to Embedded Scene event action to an object, perform the following steps:

1. From the Events view, select the object in the editor window. Double-click the event that you want to trigger. 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 window requesting the Send Event to Embedded Scene parameters opens.

4. Specify embedded scenes from the Target scenes list.

   **Note:** You must select at least one embedded scene.

5. Specify a string in the Type filter text field to filter the available visual project scenes.

6. In the Result options section specify the global parameter to be used as an external event identifier in the target scene.

   The ExternalEventID global parameter is the default.

7. Click Add New Parameter to add a new parameter.

   The Insert Parameter window opens.

8. Specify the parameter options and click Finish.

   The Insert Parameter window closes and control returns to the Add New Action window.

9. Select the parameter that you want to use from the Result parameter drop-down lists.

10. Specify a unique event identifier in the Event ID field and click OK.

    The Add New Action window closes.

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

12. Click OK to close the [objectname] Behavior window.

13. From the Project Explorer view select a scene connected with the target embedded scene.

14. From the Events view select the ExternalEvent item and double-click it.

15. In the [objectname] Behavior window set up the event actions

   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>SendEmail</td>
<td>=ExternalEventID='Btn2SendEmail'</td>
</tr>
</tbody>
</table>

These actions will be applied to the content of the **Scene2**.

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

You have added a **Send Event to Embedded Scene** event action.

*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** to an object, perform the following steps:

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 procedure command.

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

3. In the **Run Procedure** section specify whether to run a procedure from a file or repository.

4. Type the path to the procedure in the text field or click the ellipsis (...).
   - If you chose to run procedure from a file, the Open window (specific to your operating system) opens. Search and select the necessary file.
   - If you chose to run procedure from a repository, the Open window (specific to your operating system) opens. Search and select the repository.

5. In the **Login configuration** section specify whether to use the Data Source defined in the procedure or select from available connections in the drop-down list.

6. Specify variables and their values in the **Variables list** if you want to pass information from the current visual project to a procedure.

   You can use both global and local parameters as a value.
   - Click **Add variable** to create a new variable.
   - Click **Remove variable** to delete the selected variable
   - Click the ellipsis (...) for selected variable to set the value using the Expression designer.

7. To hide procedure result windows, select the corresponding check box.

8. **Optional**: If the procedure command affects the content of a scene viewed by your user, be sure to select the **Refresh the display on completion** check box.

9. **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 procedure command.

   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.
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 window.

To add an **Execute JavaScript** to an object, perform the following steps:

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 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 email program.

To add an **Execute shell command** event action, perform the following steps:

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 **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 **Apply** to apply all actions that have been added for the event.
8. Click **OK** to close the [objectname] Behavior window.

You have added an **Execute shell command** event action to an object.

**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, perform the following steps:

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 **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 CorVu Workstation Pro.
6. Select the severity level of the message from the **Logging level** drop-down list.
   Your choices are:
   - error
   - warning
   - info
7. Click **Finish**. The **Show message** 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.
9. Click **OK** to close the [objectname] Behavior window.
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, perform the following steps:

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 (...).
   
   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 on 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 options:
   - **Open**. When the event occurs, the **Save As** dialog window opens.
   - **Save to a temp file**. When the event occurs, the generated file is automatically saved in a temporary directory.
   - **Save to a file**. When the event occurs, the generated file is automatically saved in a specified root directory.
8. Specify an export path to the directory where you want to save the generated file in the **Path** field. You can use either a constant string or an expression to specify the path.
9. If you select **Save to a file** check box, there are several ways of building an export path, depending on the directory which can be specified in the **Root output directory** field on the **Server-Side File System** page of the Preferences window:
   - When the output root directory is specified, the export directory of the file will be combined of the root output directory and the path typed in the **Path** field.
   - In Cor Vu Workstation Pro when the output root directory is not specified, the export directory of the file will be combined of the application working directory and the path typed in the **Path** field.
   - In Cor Vu Web Pro when the output root directory is not specified, the export directory of the file will be combined of the temporal directory and the file name, the typed path is ignored.
10. Click **Finish**. The **Export to Excel** action is added.
    The [objectname] Behavior window is updated.
11. Click **OK**.
    The [objectname] Behavior window closes.

You have added an **Export to Excel** event action to an object.
Related topics

Setting the directory location preference for objects generated by Export actions, on page 28

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, perform the following steps:

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 window opens. Select the component from the current level and click OK to close the Set component window.
6. Click Finish. The Set focus 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.
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, perform the following steps:

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 Reload scene from the list of available actions.
5. From the [objectname] Behavior window, click Apply to apply all actions that have been added for the event.
6. Click OK to close the [objectname] Behavior window.

You have added a Reload scene event action to an object.
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 visual project, the user will be able to close the application without having access to the application’s user interface. This is useful when you are accessing a visual dashboard or visual application from an embedded window.

To add an **End session** event action, perform the following steps:

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 **End Session** from the list of available actions.
4. Click **Finish**. The **End Session** action is added. The Add New Action window closes.
5. In the [objectname] Behavior window, click **Apply** to apply all actions that have been added for the event.
6. Click **OK** to close the [objectname] Behavior window.

You have added an End session event action.

Adding an Export to PDF event action

Use the **Export to PDF** event action to export specific components of a visual dashboard or visual application to a PDF file.

The **Export to PDF** event action allows you to select specific components of a visual dashboard or application 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 or application 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, perform the following steps:

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 PDF** from the list of available actions.
4. Click **Next**. An Export to PDF specific window opens.
5. In the **Export Type** area, select one of three options:
   - **Open**. It opens **Save As** dialog. Enter the name of the file in the **File name** field to save it.
   - **Save to a temp file**. It saves the generated file in a temporary directory. The path to this file is recorded in a global or local parameter. You specify the parameter in the bottom of the window.
     - Select either **Global parameters** or **Local parameters** from the **Result parameter** list.
     - Click **Add New Parameter**. The Insert Parameter window opens.
     - Enter a name for the parameter and its description in the **Path** and **Description** fields.
     - Select a parameter type. The valid types are FilePath, Literal, and Text.
     - Select the **Has default value** check box and specify the full file path in the text field; for example, C:\DOCUME~1\<user>\LOCALS~1\Temp\<software_product_name>\reporter-<user>\pdf\..
• **Save to a file.** It automatically saves the generated file in a specified root directory when the event occurs.

6. If you select **Save to a file** check box there are several ways of building an export path, depending on the directory which can be specified in the **Root output directory** field on the **Server-Side File System** page of the Preferences window:

   ▪ When the output root directory is specified, the export directory of the file will be combined of the root output directory and the path typed in the **Path** field.
   ▪ In CorVu Workstation Pro when the output root directory is not specified, the export directory of the file will be combined of the application working directory and the path typed in the **Path** field.
   ▪ In CorVu Web Pro when the output root directory is not specified, the export directory of the file will be combined 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 the **Save to a temp file** or **Save to a file**, the **Result options** section is active.

8. If you did not previously specify the directory parameter, select either **Global parameters** or **Local parameters** from the **Result parameter** list and click the **Add New Parameter button**.

   The Insert Parameter window opens.

9. Specify a necessary parameter and click **Finish**.

   The Insert Parameter window closes and control returns to the Add New Action window.

10. Select the parameter that you want to use from the **Result parameter** lists and click **OK**.

    The Add New Action window closes.

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

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

    You have added an Export to PDF event action to an object.

**Related topics**

- Setting the directory location preference for objects generated by Export actions, on page 28

**Adding a Create File event action**

Use the **Create File** to create a specific content while running a visual project and then use this content for different purposes, for example, sent it to mail.

To add an **Create File** 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 **Create File** from the list of available actions and click **Next**.

   An Create File specific window opens.

4. In the **Output Type** area, select one of three options:

   ▪ **Open.** It opens **Save As** dialog. Enter the name of the file in the **File name** field to save it.
   ▪ **Save to a temp file.** It saves the generated file in a temporary directory. The path to this file is recorded in a global or local parameter. You specify the parameter in the bottom of the window.

      □ Select either **Global parameters** or **Local parameters** from the **Result parameter** list.
Adding a Create File event action

▫ Click **Add New Parameter**. The Insert Parameter window opens.
▫ Enter a name for the parameter and its description in the **Name** and **Description** fields.
▫ Select a parameter type. The valid types are FilePath, Literal, and Text.
  • **Save to a file**. It automatically saves the generated file in a specified root directory when the event occurs.

5. If you select **Save to a file** check box you can build an export path in different ways, depending on the directory which can be specified in the **Root output directory** field on the **Server-Side File System** page of the Preferences window:
  ▪ When the output root directory is specified, the export directory of the file will be combined of the root output directory and the path typed in the **Name** field.
  ▪ In CorVu Workstation Pro when the output root directory is not specified, the export directory of the file will be a combination of the application working directory and the path typed in the **Name** field.
  ▪ In CorVu Web Pro 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.

6. Specify a file name for the file you want to create in the **Name** field. You can also specify the file extension in this field. If you do not specify it, the file type is selected from the list of available ones:
  ▪ **application/octet-stream** - *.bin
  ▪ **image/bmp** - *.bmp
  ▪ **image/png** - *.png
  ▪ **text/calendar** - *.ics
  ▪ **text/plain** - *.txt

  **Note:** Specify a file extension if you want to create a file different from the listed ones.

7. In the **File Content** area, enter the content of the file you want to create in the **Content** field. For example, you want to add the text from **TextBox1**, enter the value: = TextBox1.Text.

8. Select the type from the **Content type** list or enter the any necessary value in the list. The available types for the content selection are:
  ▪ **application/octet-stream** specifies a binary file that is associated with an application that opens this file.
  ▪ **image/bmp** creates an image in *.bmp format.
  ▪ **image/png** creates an image in *.png format.
  ▪ **text/calendar** creates a document in iCalendar data format for representing and exchanging, calendaring and scheduling calendar information.
  ▪ **text/plain** creates an ordinary sequential file readable as textual material without much processing.

9. If you did not previously specify the directory parameter, select either **Global parameters** or **Local parameters** from the **Result parameter** drop-down list and click the **Add New Parameter button**.

   The Insert Parameter window opens.

10. Specify a necessary parameter and click **Finish**.
    The Insert Parameter window closes and control returns to the Add New Action window.

11. Select the parameter that you want to use from the **Result parameter** drop-down lists and click **OK**.
    The Add New Action dialog closes.

12. From the [objectname] Behavior window, click **Apply** to apply all actions that have been added for the event.
13. Click **OK** to close the [objectname] Behavior window.

**Adding a Refresh Component event action**

Use the **Refresh Component** event action to refresh a component in a visual dashboard by invalidating a query cache. This event action reruns the query and the results are displayed in the refreshed component.

The **Refresh Component** event action is appropriate when querying related components only.

To add a **Refresh Component** 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 **Refresh Component** 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 is a scenario where you have changed the target query table, thereby invalidating any previous results. 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 will be rerun and the valid results are displayed in the component.

To add an **Invalidate Query Cache** 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 **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 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.

**Adding a Launch LOB event action**

Use the Launch LOB event action to open large object (LOB) data from an object of a visual project (dashboard or application).

Large Object Data is binary data of any type (for example, PDF files and the content of an image file 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 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 LOB column from the list or type the necessary expression.
   **Note:** The result of the expression must be the name of the column containing LOB data.
   b. Specify the file name by typing the necessary expression or entering it in the File name field. If the field is not specified the name will be formed automatically.
   c. Specify the file extension by typing the necessary expression in the File extension field.
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 dashboard or application runs you can open the LOB data by applying the specified event to the object, which handles this action. In CorVu Workstation Pro, the data opens in the associated system application. In CorVu Web Pro, 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 and applications.

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 CorVu Web Pro.

5. In the URL field, specify the link or calculated expression. To open the Expression Designer window, double-click the URL field.

   **Note:** You can specify either an absolute or relative URL. The relative URL starts with a slash (/) and contains the part of a link that succeeds the base URL. In this case, the absolute URL will be formed by concatenating the base URL (http://host:port/application_path) and relative URL.

6. Click Finish. The Open URL action is added to the list of actions and the Add New Action window closes.
7. In the [objectname] Behavior window, click Apply to apply all actions that have been added for the event.
8. Click OK to close the [objectname] Behavior window.

*Adding a Save Cookie event action*

Use the Save Cookie event action to store the required data values under specific keys in the application storage for the current user.

In this event action, a user is an operating system user who works with the application or a user who runs the application in a web browser. Each user has a particular storage to keep the created key values. So the key values created by different users do not overwrite each other, even if the keys have the same name.

To add a Save Cookie event action to an object, perform the following steps:

1. From the Events view, select the object in the editor window and double-click the event that you want to trigger. The [objectname] Behavior window opens.
2. Click the Add New Action icon. The Add New Action window opens.
3. Select Save Cookie from the list of available actions and click Next. The Save Cookie action parameters window opens.
4. Specify the name of the key in the Key field.
5. Specify the value of the key in the Value field.
6. Click Finish to close the Save Cookie action parameters window and return to the [objectname] Behavior window.
7. Click Apply to apply all actions that have been added for the event.
8. Click OK to close the [objectname] Behavior window.

The value of the key is stored in the application storage for the current user and can be retrieved by the name of the key any time.
Adding a Load Cookie event action

Use the **Load Cookie** event action to retrieve data previously saved in the **Save Cookie** event action from the application storage for the current user.

To add a **Load Cookie** event action to an object, perform the following steps:

1. From the Events view, select the object in the editor window and double-click the event that you want to trigger.
   The [objectname] Behavior window opens.
2. Click the **Add New Action** icon.
   The Add New Action window opens.
3. Select **Load Cookie** from the list of available actions and click **Next**.
   The Load Cookie action parameters window opens.
4. Type the name of the key which value you want to retrieve.
5. In the **Result option area**, click **Add New Parameter** to create a parameter for saving the value of the key.
   
   **Note:** You can save the value of the key to the existing parameter by selecting parameter type and parameter name from the **Result parameter** lists.

6. Click **Finish** to close the Load Cookie action parameters window and return to the [objectname] Behavior window.
7. Click **Apply** to apply all actions that have been added for the event.
8. Click **OK** to close the [objectname] Behavior window.

The value of the key is loaded to the parameter by the name of the key from the application storage for the current user.

Displaying data on a map according to address information retrieved from a query

You can display data from a query result set on a map in your visual applications and dashboards.

You use calculated columns and user-defined functions for calculated columns to convert addresses to coordinates in order to use them in your visual dashboards and applications.

You must have created the *functions.js* file in the CorVu Workstation Pro or CorVu Web Pro home directory.

To display data on a map:

1. Add the following function to the *functions.js* file:

   ```javascript
   /**
    * Returns address' geo coordinates
    * @param address address string
    * @category geo
    * @return lan/lng
    */
   function getLocation(address)
   {
     var request = new RSBIHttpRequest();
     request.open('GET','http://maps.googleapis.com/maps/api/geocode/json?address='
         + address + '&sensor=false', false);
     var response = '';
     request.onreadystatechange = function()
     {
       if(request.readyState === 4)
       {
   ```
response = request.responseText;
}
request.send(null);
var x = JSON.parse(response);
return x.results[0].geometry.location;
}

2. Create a visual query based on the required data table.
3. Add the Coords calculated column to the visual query with the following expression:

   var loc = getLocation(@[address_column]);
   '(' + loc.lat + ',' + loc.lng + ')'

   Save the visual query to a repository.
4. Create a procedure with the following text (replace the values in square brackets with the valid
   for your case):

   RUN QUERY "[Path to the visual query]"
   SAVE DATA AS [Table name] (ACTION=REPLACE, CONFIRM=NO

   Save the procedure to the repository
5. Create new scheduled task that will run the saved procedure. Data will be updated on the defined
   scheduled task.
6. Create a visual dashboard or application displaying the data from the table on a map.

   Publishing visual projects for mobile devices
   You can publish your visual project as a Rocket CorVu application that runs on a mobile device.
   Published visual projects are run using the Flash rendering mode.

   To publish a visual project as a Rocket CorVu application, perform the following procedure

   1. On the toolbar, select the Publish Visual Project icon ( ) to open the Publish Visual Project
      wizard. Or in the Workspaces, Project Explorer or Repository Explorer view, right-click the
      required project and select Publish Visual Project item.
   2. Enter the Name and specify the Publication type.
      • Select Shared to make the published visual project available to all users of the repository
      • Select Private to make the published visual project available to the current user only.

         This value applies to secure repositories.
   3. Click Next .
   4. Specify the user logins and passwords for data sources and save the information by selecting the
      Save check box.

         Note: If there are no available data sources in the visual projects, this button is disabled.
   5. Click Finish.

   The wizard closes and the visual project is published.

   Visual projects of the Shared publication type are saved in the repository root in Published Visual
   Projects folder.

   Visual projects of the Private publication type are saved in the home workspace in Published Visual
   Projects folder.
Downloading Rocket CorVu to your mobile device

You can download Rocket CorVu from online distribution platforms such as the Android Market Business site for Android devices and the App Store for iPads.

For mobile devices running the iOS platform, download Rocket CorVu from Apple's Apps for Business site.

For mobile devices running the Android platform, download Rocket CorVu from the Android Market Business site.

Profiling visual dashboards and applications

You use profiling to analyze the complexity of visual dashboards or applications scenes and objects they contain dynamically; to measure the data on query running and objects creating and also the time spent on compiling the whole visual project.

With the help of profiling you can monitor the work of scenes and objects and find the problematic ones. Using this information you can optimize your visual dashboard or application.

Analyzing the complexity of a scene

You can analyze the complexity of a particular dashboard or application scene with the help of the Analyzer view.

You must have a visual project to analyze and open the Analyzer view.

Analyzer counts the weight of each object lying on a scene. Each object has a number of properties that contain values and expression of different complexity. All default properties are basic units and cannot be subdivided into simpler ones. Each default property has the weight that equals 1. The weight of non-default properties is counted as an expression according to their complexity. In the analysis process you can count the weight of each object and get the overall weight of a scene.

The objects that can run queries or the DataTemplates have incomputable weight. They are highlighted in red in the Analyzer view.

The analysis data is represented in a table with the following columns:

- **Elements** column displays a tree structure of the analyzed scene.
- **Weight** column displays a weight of a particular element.
- **Cumulative Weight** column displays a weight of a particular parent and the child elements.

To analyze a scene:

1. Open the scene you want to analyze in Design or Runtime mode.
2. In the Analyzer view click Analyze.
3. The analysis of the scene is displayed in the Analyzer view as a tree listing all the components and objects of the scene.
4. You can select an object in the view and it will be displayed as selected on the scene and in the Project Explorer view.
5. You can specify the values for objects with incomputable weight by clicking the ellipsis (…) in the Cumulative Weight column or Set Row Count button on the view toolbar. This value will be used as an average row amount in the incomputable object.
Profiling a visual project

You can profile a visual dashboard or application with the help of the **Profiler** view.

You must have a visual project to analyze and open the **Profiler** view.

The **Profiler** view displays the data on query running, the amount of created objects, and the time spent on creating them and calculating expressions on the visual project. The profiler data is represented in a table with the following columns:

- **Elements** column lists all the scenes with the objects, expressions and queries of a visual project.
- **Count** column shows the number of times the object was created.
- **Time** column displays the time spent on creating an element.
- **Cumulative Time** column displays the time spent on creating a parent and its child elements.
- **Percentage** column displays the percentage ratio of the time spent on creating a particular element to the time spent on creating the entire project.

To start the profiling process:

1. Click the **Profile** button on the **Profiler** view toolbar.
2. Open the required visual dashboard, application, or a separate scene in the **Runtime** mode.
3. When all scenes are created and all queries are run click the **Profile** once again. The profiler results are displayed in the tree table.
4. You can modify the way the results are displayed by clicking **Tree View** or **Flat View** buttons on the **Profiler** view toolbar.
5. You can export or import the profiler data by clicking the **Import Data** or **Export Data** toolbar buttons.
6. You can select an object in the view and it will be displayed as selected in the **Project Explorer** view.
7. You can sort the values in the **Profiler** view columns by clicking their headers.

Working with visual dashboards or applications using JavaScript

The **Scene Script** perspective provides the interface from which you can create and work with visual dashboards or applications by using JavaScript.

From the **Scene Script** perspective you can create scripts for adding objects, setting their properties and managing them on the dashboard or application dynamically.

Opening the Scene Script perspective

To open **Scene Script** perspective, select **Window > Open perspective > Other > Scene Script**.

Scene Script perspective views

The **Scene Script** perspective consists of the editor area and several views, which are defined below:

- **Object Explorer**
  
  This view contains the objects added to a visual project from the **Palette** view.

  You can expand each element to view a list of its properties and select the necessary ones to add to the script.
Adding objects to a visual project in the Scene Script perspective

You can add new objects to the scenes of your visual dashboard or application in the Scene Script perspective.

You must have an open visual dashboard or application.

You can add new objects to your visual projects in 3 ways:

- Create JavaScript code to add new objects or edit the existent ones.
- Select an object from the Code Library view to add it by using the [objectname] wizard.
- Use the Palette view in Design tab of the visual project editor. For more information on this option see Working with the Palette view help section

Note: All objects added from the Palette view are available in the Object list of the SceneNScript editor.

Adding a new object using JavaScript code

You can dynamically add an object on you visual dashboard or application using JavaScript code.

Use SceneNScript editor to add JavaScript code for a particular scene. The script created in this editor manages only this scene.

Example 1

```javascript
var btn = new Button();
project.scene.Level1.add(btn);
```

Example 2

```javascript
var l = project.scene.Level1,
b = new Button();
```
1.add(b);

**Example 3**
If necessary you can call a parent component:
```javascript
var prnt = object.parent();
```

### Adding a new object using the Code Library view

You can open the [objectname] wizard and see prompts when creating objects on a particular scene of your visual project by using the **Code Library** view.

To add a new object to your visual project from the Code Library view:

1. Double-click the required **SceneNScript** item in the **Project Explorer** view. The **SceneNScript** editor opens.
2. Select an object you want to add from the **Code Library** view and double-click it. The [objectname] wizard opens.
3. Specify the object name in the **Name** field.
4. Select **Display only properties with modified values** check box if you want to display only the properties that you modify in this wizard. All the properties that you left with default values will be excluded from the **SceneNScript** editor.
5. Select **Add to level** to add the object to the current level. Clear the check box if you want to add an object just to the code. It will be invisible on the scene.
   
   For detailed information on adding objects see the following topics **Working with the Palette view** help section.
6. Click **Finish** to add the object code to the **SceneNScript** editor.

### Specifying the object properties using JavaScript

You can modify properties of all objects added by using JavaScript to your visual project in the script.

You must have an open visual dashboard or application with objects on it.

To modify objects or component properties:

1. Specify an object real name and the properties you want to edit. For more detailed information on the available object properties see the Object properties help section.
2. Set the necessary values for these objects or components.

   **Note:** Each property can be a bindable object. To make it bindable, add an underscore (‘_’) prefix to the required property, for example,
   ```javascript
   object.bind(function() { /*function body*/ }, [label._Text]);
   ```

**Modifying the button properties:**
```javascript
var l = project.scene.Level1,
    b = new Button();
```
Specifying the object properties added from the Palette view

You can modify properties of all objects previously added to your visual project from the Palette view by editing the scene script. All available object properties are listed in the Object Explorer view.

You must have an open visual dashboard or application with objects on it.

To modify objects properties:

1. With an open SceneNScript editor in Scene Script perspective, select the necessary object from the necessary visual project tree. All added elements are displayed in the Object Explorer tree as folders.
2. Expand the necessary object properties and select the required one and double-click it. The selected property adds to the SceneNScript editor and you can modify its values.

For more information on object properties see Object properties help section.

Accessing different components of a visual project

You can refer to different components of a visual project in your JavaScript code.

You can refer to the following components:

- Globals
- Scene

You can use the listed methods to call to an object of the visual project:

- on()
- bind()
- unbind()
- field()
- getActions()
- parent()

Referring to global variables in JavaScript

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 or application.

For detailed information on global parameters see Working with global parameters topic.

To refer a particular parameter of the global variables use the following code:

```
project.Globals.Parameters.<paramName>;
```

To refer to a color sequence use the following code:

```
project.Globals.ColorSequences.<sequenceName>;
```

For more information on color sequences see Working with color sequences topic.
To assign an image to a property use the following code:

```javascript
project.stockImage("<imageName>");
```

**Note:** `<imageName>` must be saved in Globals/Images directory. For more information on images see Storing images with visual projects, on page 325 topic.

**Referring to scene properties and objects in JavaScript**

You can refer to scene local parameters or its objects from the JavaScript code.

For detailed information on scene parameters see Creating scene parameters topic.

To refer to a particular local parameter of the scene use the following code:

```javascript
project.scene.Locals.Parameters.<paramName>;
```

To make a parameter bindable and use it in bind() method use an underscore ("_") before the parameter:

```javascript
project.scene.Locals.Parameters._<paramName>;
```

To refer to a scene object use an object name:

```javascript
project.scene.Level1.Label1;
```

**on() method**

You can handle the events on your visual dashboard or application using the on method.

The event type and callback() function are passed to the on() method. The keyword this in the callback function points to the event source object.

**Syntax**

```javascript
object.on(type, function());
```

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>The event type is set by a string constant. All events names are written in lowercase and underscore ('_') is used instead of a space. For more information on the available event types see the Events help section.</td>
</tr>
<tr>
<td>function()</td>
<td>You can use any JavaScript functions in this parameter.</td>
</tr>
</tbody>
</table>

**Examples**

**Example 1. Handling "Click" event of a button label:**

```javascript
var l = project.scene.Level1;
var l1 = new Label();
var l2 = new Label();
l1.Text='One';
function callback(){this.Text = this.Text+' clicked'};
l1.on('click', callback);
l2.on('click', callback);
```
Example 2. Handling "Click" event of a button:

```javascript
var l = project.scene.Level1,
    b = new Button();
l.add(b);
b.Location = [0,0];
b.Caption = "Button 1";
var f = function(){
    this.Caption = "Clicked!";
}
b.on("click",f);
```

**bind() method**

You can bind a function to the changes of bindable objects values. You can also use this method to bind a function to an object that will be called only when an object is created.

Use this function to initialize the generated objects within the DataTemplate.

**Syntax**

```
object.bind(function(), [bindable_objects]);
```

or

```
object.bind(function());
```

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>function()</td>
<td>The keyword <code>this</code> of the <code>function()</code> defines the current object for which the <code>bind()</code> method is called.</td>
</tr>
<tr>
<td>[bindable_objects]</td>
<td>The current bindable objects are passed as the function parameters.</td>
</tr>
</tbody>
</table>

**Examples**

**Example 1**

```javascript
var level = project.scene.Level1,
    l = new Label(),
    t1 = new TextBox(),
    t2 = new TextBox();

l.Location = [0,1];
t1.Location = [-1,0];
t1.Text = "Hello";
t2.Location = [1,0];
t2.Text = "world!";

var callback = function(text1, text2) {
    this.Text = text1 + " " + text2;
}

l.bind(callback, [t1._Text, t2._Text]);
level.add([l,t1,t2]);
```
Example 2. Create a visual project with the following query:

```
SELECT ID from Q.STAFF
```

Add this function in the DataTemplate, and the label in each generated string will have the proper ID column value from the ResultSet.

```javascript
var tabl = new Table();
tabl.DataTemplate1 = new DataTemplate(); tabl.DataTemplate1.Query = "STAFF";
tabl.DataTemplate1.Level1 = new Level();
tabl.DataTemplate1.Level1.TableRow1 = new TableRow();
tabl.DataTemplate1.Level1.TableRow1.TableCell1 = new TableCell();
tabl.DataTemplate1.Level1.TableRow1.TableCell1.Label1 = new Label();

tabl.DataTemplate1.Level1.TableRow1.TableCell1.Label1.bind(function()
{
  this.Text = this.field("ID");
});
tabl.DataTemplate1.Level1.TableRow1.add(tabl.DataTemplate1.
  Level1.TableRow1;
  TableCell.Label1);
tabl.DataTemplate1.Level1.TableRow1.add(tabl.DataTemplate1.Level1.
  TableRow1.TableCell1);  
tabl.DataTemplate1.Level1.add(tabl.DataTemplate1.Level1.TableRow1);
tabl.add(tabl.DataTemplate1);
project.scene.Level1.add(tabl);
```

unbind() method

You can unbind bindable objects from all the functions they were used in.

**Syntax**

```javascript
object.unbind([bindable_objects]);
```

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[bindable_objects]</td>
<td>The current bindable objects that must be unbound.</td>
</tr>
</tbody>
</table>

The example below requires two Param parameters, specified in Globals, Parameters and in Locals, Parameters.

**Example**

```javascript
var label = new Label(), gBtn = new Button(), lBtn = new Button(),
  uBtn = new Button(),
  level = project.scene.Level1;

gBtn.Caption = "Increase Global";
gBtn.Location = [-1.5, 1.5];
gBtn.on("click", function() {project.Globals.Parameters.Param++});

lBtn.Caption = "Increase Local";
lBtn.Location = [1.5, 1.5];
lBtn.on("click", function() {project.scene.Locals.Parameters.Param++});

label.bind(function(global, local) {this.Text = global + " " + local;},
  [project.Globals.Parameters._Param,project.scene.Locals.Parameters._Param]);
label.Location = [0.5, -0.5];
```
uBtn.Caption = "Unbind Local";
uBtn.Location = [0, -1];
uBtn.on("click", function()
{
    label.unbind([project.scene.Locals.Parameters._Param]);
});
level.add([gBtn, lBtn, uBtn, label]);

As a result the Increase Global and Increase Local buttons increase the parameters values with every click. The Unbind Local button unbinds the local parameter value and it stops increasing when you click the Increase Local.

field() method
You can use this method to get the value of a column with the specified name.

Use this method in bind() to access the row data of the object ResultSet lying in the DataTemplate.

Syntax

object.field("name");

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Use an object name as its identifier for the field() method.</td>
</tr>
</tbody>
</table>

Example

See the example in the bind() method, on page 255 topic.

Adding event actions

You can add various event actions to the objects lying on the visual project by using JavaScript.

To add an event action:

1. Double-click the required event action in the Code Library view. The [action] wizard opens.
2. Specify the event action parameters. For more information see Setting up event action navigation for your visual application or visual dashboard, on page 228.
3. Click Finish. The action will be added to the SceneNScript code.

You can also add new actions to the existent object:

1. Select the required object from the Object list.
2. Select the required action from the Events list.
3. Click Add. The specified event is added to the code.

Working with visual dashboard or application scene scripts

You can create, view or edit scripts for a particular scene or for the entire project.

Use the SceneScript editor to add, view or edit a script.

To open the SceneScript editor for a particular scene:

1. Open the Project Explorer view and select the required dashboard or application.
Chapter 10: Working with visual projects

To create a script for the entire project:

1. Open the Scripts view and select the required dashboard or application.
2. Right-click the dashboard or application name and select Insert Scene Script. The New Scene Script wizard opens.

Use scripts for the entire project to define the general functions for your visual project and all the data that you want to use in different scenes.

Generating JavaScript code for the scene objects

You can generate JavaScript code for the objects of visual dashboards and applications. You can add this code to any scene of any visual dashboard or application.

You must have an open visual dashboard or application with objects on it.

You can generate JavaScript code for all scene objects and their connected items:
- Object properties
- Events
- Actions (except Execute JavaScript and Launch Lob)
- Query parameters
- All child components

1. In the Design mode or the Project Explorer view, select one or several objects you want to generate JavaScript code.
2. Right-click the selected objects and select Generate Code from the menu. Read-only JavaScript code of the selected objects is displayed in the Scene Script editor of the current object.
3. Select the generated JavaScript code, right-click it, and select Copy from the menu.
4. Open the SceneNScript editor for the required scene, right-click anywhere in the editor, and select Paste from the menu.
5. Open the scene to which you copied JavaScript code in the Runtime mode. The objects are displayed on the required scene.

Data source connections and visual projects

Visual projects (visual reports, dashboards, and applications) are intended to be shared across the enterprise. In order to facilitate sharing and the distribution of these visual projects, CorVu Workstation Pro ties the query to a connection information alias (rather than associating the object to a specific data source).

You must set up a connection information alias for each data source that will provide data for these objects.
Specifying data source connections for visual projects

You set up a connection information alias for each data source that will provide data for the visual project (visual report, dashboards, or applications).

When you add the queries that will be used in a visual project, you must specify what connection information alias will be associated with each query.

To change the data source that will be used for a query, you edit the query information and select a connection information alias entry that points to the data source that you want to use. This feature allows users to easily share visual reports and use their own data sources. In addition, it facilitates moving between test and production data sources.

To specify a connection information alias for a data source, perform the following steps:

1. Open the Insert Connection window.
   You can open the Insert Connection window in one of the following ways:
   • From the Project Explorer view, right-click the Connections node for the specific visual project. Select Insert Connection from the context menu.
   • From the Project Explorer view, double-click the Connections node. The Insert Connection window opens.
   • Click a query from the Workspaces or Repository Explorer view. With the mouse button pressed, drag the query to the Queries node in the Project Explorer view. A new data source connection is automatically added to the Connections folder of the Project Explorer view.
2. From the list of available data sources, select the data source that will be associated with this connection information alias.
3. Specify a unique name for this connection information alias in the Connection name field.
4. Click Finish. The Project Explorer view lists the new connection information alias under the Connections node in the tree.

Note: You can also set up a connection information alias for a data source by dragging the data source from the Workspaces view to the Connections folder for the visual project in the Project Explorer view. The connection information alias is added with the name ConnectionN. You can edit the name of the connection information alias by right-clicking the entry and selecting Rename.

You have setup the data source connection information for the visual data object.

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.

Related topics
Specifying queries for visual projects, on page 260
Editing queries specified for visual projects, on page 261
Specifying queries for visual projects

Visual project (visual reports, dashboards, or applications) can include one or more queries to run to obtain the result data.

You can create a new query that will be included in the visual project or you can use existing queries. To specify a query that will be used in the visual projects, perform the following steps:

1. From the **Project Explorer** view, right-click the **Queries** node and select **Insert Query** from the context menu.
2. Specify a unique name for the query in the **Query name** field.
3. Identify the query by clicking one of the following radio buttons:
   a. Click **Create a new visual query** to insert a new visual query.
   b. Click **Create a new forecast** to insert a new forecast.
   c. Click **Create a new analytical query** to insert a new analytical query.
   d. Click **Attach to an existing query** to insert an existing query. Specify the name of the query and where it resides in the **Select query** field. You can click **Browse** to search for the query.
   e. Click **From file** to insert a static query and select one of the following options.
      - Click **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.
      - Click **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 click **Browse** to search for the query.

**Note:** You can open CSV, DBF, IXF, TXT, XML, SQY, and TAB files and use them as queries. You can edit the imported result set by filtering, adding calculated columns, applying grouping and aggregation. When you save the visual project, these files can be converted to dynamarts and saved with the visual project. All modifications will be preserved.

4. If you are inserting a visual query, select the connection that points to the data source that you want to use for this query from the list of available connections. To add a new connection information alias, click **Add Connection**.
5. If you are inserting an existing forecast or analytical query into a visual project, you can edit connection settings for subordinate queries in the Edit Connections window.
6. If you are inserting a new forecast or analytical query you cannot set connections for subordinate queries because these queries are not specified yet.
7. 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.

Depending on your selection, the **Visual Query** editor or the **Forecasts editor** opens in a new tab. The query or forecast is saved automatically 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 project in the Project Explorer view.

Related topics
- Specifying data source connections for visual projects, on page 259
- Editing queries specified for visual projects, on page 261
- Editing query connections, on page 262
- Importing query results from data files, on page 104
- Managing source data files in dynamarts, on page 108

Editing queries specified for visual projects

You can edit the queries specified for visual projects in the Edit Query window.

To edit a query that is used in the visual project:

1. From the Project Explorer view, right-click the query that you want to edit, and then select Edit Query from the context menu.
2. You can change the name of the query in the Query name field.
3. If you are editing a visual query, select the connection that points to the data source that you want to use for this query from the list of available connections. To add a new connection information alias, click Add Connection.
4. If you are editing a forecast or an analytical query into a visual project, you can edit connection settings for subordinate queries in the Edit Connections window.
5. Click Attach to an existing query to replace the selected query with another existing query. Specify the name of the query and its location in the Select query field. You can click Browse to search for the query.
6. Click From file to insert a static query and select one of the following options.
   - Click 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.
   - Click 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 click Browse to search for the query.
7. Click Finish to close the Edit Query window.

Related topics
- Specifying queries for visual projects, on page 260
- Specifying data source connections for visual projects, on page 259
- Editing query connections, on page 262
Editing query connections

You can edit the connection information for existing analytical queries or forecasts that you are inserting into a visual project.

This task describes how to edit the connection information for existing analytical queries or forecasts that you are inserting into a visual project. Using the Edit Connections window as a way to change connection information saves you from having to edit source queries every time you want to modify the connection settings for an analytical query.

**Note:** When inserting a new analytical query or forecast objects you cannot set connection information because you have not specified the source queries for these objects yet.

To edit query connections:

1. In the Insert Query window, click **Edit Connections** to open the Edit Connections window.
   - The queries embedded into the analytical query or forecast that you have selected are listed in the **Query** column. You can expand or collapse the list of queries.
   - The connections for the subordinate queries are displayed in the **Connections** column. The default value is an empty string which means that a query uses default data source stored within this query.
2. Click in the field of the subordinate query connection to which you want to apply changes, and then select appropriate connection from the drop-down list.
   - **Note:** Select the empty string value to set the default data source connection saved in the query itself.
3. If the required connection is not in the list of available connections, use the Insert Connection window to add a connection to the visual project. To open the Insert Connection window click **Add Connection**.
4. Click in the field of the query connection which you want change, and then select the added connection from the list of available connections.
5. Click **OK** to close the Edit Connections window.

**Related topics**

- Specifying data source connections for visual projects, on page 259

Assigning data retrieval schedules to visual project queries

To control visual project data retrieval, you must assign the data retrieval schedules to the queries in the visual project.

To assign data retrieval schedules to visual project queries:

1. Right-click the **Queries** node for the visual project 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 project.
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 project 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 project 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.

### Adding custom JavaScript functions to visual projects

You can use your own JavaScript functions in visual reports, dashboards and applications with the help of JavaScript Modules.

You use the JavaScript Module editor with syntax highlighting to write and edit your own functions. To open the editor, select **File → New → Other**. In the New wizard, select **CorVu Objects → JavaScript Module**.

After you have created a JavaScript Module, you can save it in a repository or file system, and then add it to a visual project.

To add custom JavaScript functions to a visual project:

1. Open the Insert JavaScript Modules window. In the **Project Explorer** view, right-click the **Globals** node, and then select **New → Insert JavaScript Module** option.

2. In the Insert JavaScript Modules window, decide whether you need to create a JavaScript Module or attach an existing one.

3. If you decide to create a JavaScript Module, specify the required name in the **JavaScript Module name** field, select **Create new**, and then click **Finish**. The empty JavaScript Module opens in the separate editor window. Write the required JavaScript functions and close the editor.

4. If you decide to attach an existing JavaScript Module, select **Attach to an existing**, and then specify the required module form a repository or file system. Click **Finish**.

**Note:** You can view comments added in the JSDoc format to the script. These comments are visible to a user in the **Expression Designer**.

5. You can view the attached JavaScript Modules in the Project Explorer view under the **Globals → JavaScript Modules** node.

All the functions defined in the created or attached JavaScript Module can be found in the **Expression Designer** under the **User** node. You can use them as other expressions in your visual project.

### Working with the Visual Designer

In CorVu Workstation Pro, 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.
Chapter 10: Working with visual projects

- **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 show 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 or visual application 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.
Viewing object references

You can view all references to an object used in the visual project in the Show References window. You must have a visual project opened in the Visual Designer editor.

To view object references:

1. In the Project Explorer view, right-click the required object and select Show References from the context menu.
   The number of objects that have references to the selected object is displayed in the Show References window. If the selected object is referenced by other objects, the Advanced button becomes available.
2. To view all references, click Advanced, and then navigate to the object references in the tree.
3. You can use the arrows to navigate to the previous or to the following object that references the selected one.
4. To expand or collapse the tree of objects, click Expand or Collapse.
5. To view the selected referenced object in the Project Explorer view, click Show in Project Explorer.

Working with the editor

The Visual Designer editor window displays a canvas on which you will 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 will display your project as a user will see it.

Multiple visual project elements can be open at the same time in the Visual Designer editor window. Each element that is open 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 the 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 or visual application. 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 or application as a user will see it.

- **Print Preview mode**
  You use Print Preview mode to test the content in your visual report. 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 report as it will be seen by a user.
You can control how your visual project is rendered in the Print Preview and Runtime modes by specifying a mode in the drop-down list in the workstation menu bar. Depending on the type of the visual project the following rendering mode options can be available:

- **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.
- **HTML5** optimizes the content for better performance in modern web browsers and on mobile devices.
- **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, editor rulers, guides, and grids help you position visual objects precisely.

**Rulers**

When visible, rulers appear along the top and left side of the active 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, perform the following steps:

1. Select View → Preferences
2. Select Visual Designer from the tree.
3. Select one of the Visual Designer Ruler Unit radio buttons 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.
Working with the Events view

Use the Events view to set properties for the event actions in your visual project.

To launch the Events view, perform the following steps:

1. From the menu bar, click Window → Show View → Show View → Other and expand the Visual Designer Category
2. Select Events from the list of options and click OK.

Events are actions that users perform when navigating through the visual project (dashboard or application). These actions are what provide the dashboard or application with the necessary navigation and interactive functionality. For information about Events, see Setting up event action navigation for your Visual Application or Visual Dashboard.

For information about how to assign events and event actions to Visual Dashboards and Visual Applications, see Assigning events and setting up event actions.

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 object properties

Properties and their corresponding values for each element or object in a visual project are displayed in the Properties view.

The properties presented in the Properties view vary depending on the element or object that is selected in the editor window or in the Project Explorer view. Properties are listed with their current or default values.

There are many ways for you to modify the property values. You can modify the property values using constants, mathematical expressions, function expressions, or property expressions. For many properties, you can select from the property’s list box of accepted values. Property values that you specify must match the data type required for the property.

To specify or modify individual object properties from the Properties view:

1. The Properties view opens automatically when you are in the Visual Designer perspective. If the Properties view is not open, you can open it by selecting Window → Show View → Properties. To open the Visual Designer perspective, select Window → Open perspective → Visual Designer.
2. Select from the Project Explorer view, or the editor window, the element or object whose properties you want to view or modify. Once selected, the properties for that element or object are displayed in the Properties view.
3. Click the property field for the property that you want to modify and enter the desired value. Depending on the property, the values that you enter could be a constant, a global or local parameter, a column name that is associated with the results of a query, a selection from a list of available values or an expression. The value that you enter must match the data type required for the property.

4. You can use the Expression Designer to help you specify property values. Click the **Edit with Expression Designer** toolbar button in the **Properties** view to open the Expression Designer.

5. You can use the **Variables** bar to help you specify property values that will use global or local parameters or column names from a set of query results.

### Specifying properties for groups of objects

You can specify properties for groups of objects that you have selected in the editor window.

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 object has a different value, the field will be 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 cell. The column or parameter name is inserted as the value for the property that you selected.
Using the Expression Designer to specify property or field values

You can use the Expression Designer to specify any field or property value that is associated with a selected object or event action.

The Expression Designer is an alternative way of specifying object properties or field values in dialogs. Rather than type object properties or field values directly in the property cell of the Properties view or any dialog window, you use various features in the Expression Designer to help you create and format the value. The advantages of using the Expression Designer are:

- The given templates can help you format each value.
- In addition, you have more space available to work and view your property or field value (which can grow quite large if expressions are used).

To specify a property or field value in the Expression Designer:

1. To modify a property value, in the Properties view, select the property you want to modify and click the Edit with Expression Designer toolbar button. The Expression Designer window opens.
2. To modify a field value, double-click the required field in a dialog, the Expression Designer window opens.
3. 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.
4. Each of the built-in functions that can be included in 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.
5. The property or field that is being edited is displayed above the value field. You can use both properties and functions to build your property or field value in the value field. Double-click a function to insert it into the value field. Double-click a property to insert a reference to it in the property value field. In both cases, the function or property is appended to what is already in the value field. To replace the content, highlight what you want to replace in the property field and then double-click the property or function. To insert content, position the cursor in the field and then double-click the property or function. Functions are placed in the field in a template form. Depending on your logic, you will replace the parameters with your own values.
6. The following buttons, located below the value field, provide additional help when creating a property expression:
   - **Undo**: Click this button to undo the last edit applied to the expression that you are building in the value field.
   - **Redo**: Click this button to redo the last edit applied to the expression that you are building in the value field.
   - **Edit**: Click this button to build an expression that will define a new value. To edit the value, select the new property or value from the Expression Designer tree in the left pane of the window, and click Edit. The new value is displayed above the value field. You can set or edit multiple values by clicking Edit without closing the Expression Designer window. Each property that you modify is marked in the Expression Designer tree with an asterisk (*).
   - **Open FormatNumber window**: Click 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 value field. The expression will be reviewed and simplified where possible.
   - **Select color**: Click 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 value field.
7. Click **Apply**. The property or field value is updated.
8. Click **OK**. The property or field value is updated. 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;&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>
</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:

```
Color = if(S_terrain="water", Blue, Yellow)
```

In this example, the condition of the object is based on a parameter, `S_terrain`. The object might be a polygon in a map layout or a data symbol shown on a chart. If the value of the scene parameter is water, the color of the object is blue. Otherwise, the color is yellow.

You can specify different conditions for the `if()` function, such as a column value or a numerical threshold. For example, you can make your visual project be responsive to changes and show their significance in the appearance of the objects in the project. Profit or loss, for example, can be displayed visually using color: black for profit, red for loss. The following property expression can be used to distinguish profit and loss in the color property of the object:

```
Color = if (sales>=expenses, Black, Red)
```

#### Displaying column data in a text string

You can enhance the data points for a layout object by adding a `Label` object that reveals information about the record returned. For example, you might want each (data point) in a tree chart to show the name of a department in a store, preceded by the word Department. You could insert a `Label` object into the data template for the layout object and then specify a property expression which combines the column name and a literal string.

<table>
<thead>
<tr>
<th>Task</th>
<th>Rule</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<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>
In this expression, dept_desc is a column returned by the query to your data template.

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.

**Visual Application Objects**

The visual Application Objects palette contains objects specific to visual applications including Canvas and Tree.

**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 or applications.

**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 visual dashboards or applications.

**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 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 or visual applications 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 or visual applications 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 application, and nearly all of them (except for Canvas and Tree) can be inserted in a visual dashboard.

The objects that do not have interactive properties can be inserted in a visual report. There is no need to insert interactive objects in visual reports because visual reports are not interactive.

To insert a graphical object into a visual project:

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:

1. Select Window → Open perspective → Visual Designer
2. With the Visual Designer perspective open, select Window → Show View → Other
3. 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, visual dashboard scene, or visual application 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.
Chapter 10: Working with visual projects

Modifying the Palette view display settings

You can alter the way objects are displayed in the Palette view.

To alter the way objects are displayed in the Palette view perform the following steps.

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:

1. Select Window → Open perspective → Visual Designer
2. With the Visual Designer perspective open, select Window → Show View → Other
3. From the Show View window, select General → Palette and click OK

The Palette view opens.

**Note:** If there is no active visual project, the palette will be unavailable. To make the Palette available, you must first select a visual project. For example, a visual dashboard, visual application, visual report or visual query object.

2. Each group of objects is displayed in a separate drawer. Clicking the drawer will alternately expand and collapse the drawer, displaying or hiding the object icons.

3. To alter layout of the objects in the drawers, 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. To alter the general appearance and behavior of objects on the Palette, 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 window.

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 active 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 visual dashboard or visual application 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 or visual applications.

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, you will select the source object 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 visual project.
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 in 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.


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
   ▪ Execute SQL statement

   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.
Note: The **Execute SQL statement** action is only available to objects if one or more repository connections have been defined for the visual project in the **Connections** folder.

- Execute procedure
- Set values
- Navigate back
- Print scene
- Execute shell command
- Show message
- Execute JavaScript
- Export to Excel
- Set focus
- Reload scene
- End session
- Export to PDF
- Send Email
- Send Event to Embedded Scene
- Save Cookie
- Load Cookie

Although you can select any of the actions, the most common use of the **Timer** object is to control the behavior of another object. To control the behavior of another object at each tick of the **EventInterval**, select the **Set values** action. If you intend to control the behavior of another object, then you must have created the object before you start the process of assigning an action to the **Tick** event.

9. Click **Next**. The next window that opens depends on the action that you selected. If you selected the **Set values** action, the **Set values** action parameters window opens.

10. Double-click in 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)
Chapter 10: Working with visual projects

- 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 page or scene.

To insert text into a visual project:

1. In the editor window, select the visual project to which you want to add the object. If the page or scene is not active 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 in the visual project page or 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 and applications, 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 the **LineStyle** property cell.
You can copy and paste multiple lines of text in the **Properties** view. The line breaks will automatically be displayed.

---

**Inserting line objects**

You can use the line objects to draw lines, arrows, double arrows, polylines or multipolylines in your visual project.

To insert a line, arrow, double arrow, polyline or multipolyline object into your visual project page or scene:

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, which represents 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, which represents 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 in the visual report page or visual project page or 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 and applications, 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 project page or 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**.
Chapter 10: Working with visual projects

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, which represents 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, which represents 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, which represents 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, which represents 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, which represents 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 in 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 page or scene.

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 page or scene:

1. In the editor window, select the visual project page or 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 in the ImageSource property cell. Click the ellipsis (...) that appears in the cell. The Picture Source window opens. Select one of the following options:

- **Column** to specify that the image will reside in a query result column. This option is available when a query is available to the visual project for selection. From the drop-down list of available queries, select the one that will return the result data. From the second drop-down list, select the column that will contain the image. The Queries folder in the Project Explorer view lists all the queries that are available to the visual project.

- **Linked** to specify that the image will reside in a local or network file. Select Linked and specify the path to the image file.

- **Embedded** to specify that the image is embedded in the visual project. In order for this option to be available you must have images available to the visual project for selection. All images that are available are listed in the Images folder that is under the Globals folder in the Project Explorer. To specify that the image is embedded, select Embedded. Select the image from the drop-down list of available images.

4. From the Properties view, you can modify any of the additional properties to alter the appearance or behavior of the picture object in the visual project page or 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 project page or 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 visual dashboard or application user and based on the response to the control, you can trigger subsequent events. **Control** objects are only available for placement in visual dashboards or applications.

The **Controls** palette offers the following control objects:

**Slicer Tool**
- An object that allows filtering for layout objects such as grid and charts.

**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 (option) 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 can enter additional choices using the editable text field of the **Combo** object.
**List Box**
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 dashboard or application user and based on the response to the control, you can trigger subsequent events. **Control** objects can only be inserted in visual dashboards and applications.

To insert a **Control** object into a visual project:

1. In the editor window, select the visual dashboard scene to which you want to add the object. If the scene is not opened in the editor, then double-click the page or scene in the **Project Explorer**.
2. From the **Palette** view, select the **Control** object and click in the editor window or double-click the **Control** object. The **Control** object is drawn based on default properties.

   **Note:** You can also select the **Control** 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 **Control** object is drawn.

3. From the **Properties** view, modify the object’s properties to make the control object appear and behave as you want. There are several ways to specify property values for the control objects. You can select values that are listed in the **Properties** view, type values directly in **Properties** view, use the **Expression Designer**, or use the **Variables** toolbar.
4. You can connect the control object to a target object, in most cases that is a layout object. This allows you to change the behavior of a target layout object based on the user’s input to the control object.

   For example, you can connect a **List Box** control object with the department names of your company to a **Grid** layout object containing all the department information. Use the Connectivity tool to connect the two objects. Your user will select a department from the list box, and only the information for that department will be displayed in the grid.
5. You can define an event action for the control object. This means that when a user performs a certain action on the object such as a mouse click, an event such as a jump to a scene will occur.

**Inserting 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 project 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 opened in the editor, then double-click the page or scene in the **Project Explorer**.
2. From the **Palette** view, select the **DateTime** object and click in the editor window or double-click the **DateTime** object. The **DateTime** object is drawn based on default properties.
Note: You can also select the DateTime 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 DateTime object is drawn.

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.

6. 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.

7. Insert the object whose behavior will depend on 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.

8. 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_TEXTURE property will equal the DateTimeObject.Value.

9. To use the DateTime object return value as a parameter:
   a. Create a visual query object (Query1) that requests 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 date parameter to control what rows are included in the grid. In this example, the date parameter value is supplied from the DateTime object Value property.

e. You can also use the Connectivity tool to set up the connection between the DateTime object and the Grid object.

Inserting a prompt hierarchy control object

The PromptHierarchy object is a control that allows the end user to select one or more values from a flat or hierarchical list of options to be displayed.

PromptHierarchy control objects allow a user to control what hierarchy levels and individual elements of a prompt hierarchy are expressed in query results displays. For example, you create a table that is populated by a query that has an associated prompt hierarchy. You then create a PromptHierarchy control and link it to the query's prompt hierarchy. Now, when you run the dashboard, you can open a window and select which parts of the query data that you want to display in the table.

To insert a PromptHierarchy control object:

1. In the editor window, select the visual dashboard scene to which you want to add the object. If the scene is not opened in the editor, then double-click the page or scene in the Project Explorer.

2. From the Palette view, select the PromptHierarchy object and click in the editor window or double-click the PromptHierarchy object. The PromptHierarchy object is drawn based on default properties.

   Note: You can also select the PromptHierarchy 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 PromptHierarchy object is drawn.

3. Select the query that you want to connect to the control object from the Query drop-down list.

   Note: All of the queries that have been previously added to the Queries folder in the Project Explorer are available in the Query drop-down list.

4. Select the prompt hierarchy that you want the object to control from the Prompt hierarchy drop-down list.

   Note: All of the prompt hierarchies that have been previously linked to the queries in your Queries folder in the Project Explorer are available in the Prompt hierarchy drop-down list. If no prompt hierarchies have been associated with your dashboard queries, this drop-down list is empty.

5. Click Finish.
   The PromptHierarchy Wizard closes and the new PromptHierarchy control object is displayed on the canvas.

Adding data symbol objects

A Data Symbol object is an icon that is used to represent a data point in a layout object.

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:

**GoogleMapMarker**
A data symbol object that displays a data point as a map marker on a Google Map object.

*Note: GoogleMapMarker is only available in visual dashboards and applications.*

**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 visual dashboard scene to which you want to add the object. If the scene is not opened in the editor, then double-click the page or scene in the Project Explorer.
2. From the Palette view, select the Control object and click in the editor window or double-click the data symbol object. The data symbol object is drawn based on default properties.

   **Note:** You can also select the data symbol 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 data symbol object is drawn.

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 in 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 in a report page or dashboard scene.

To insert a connector into a visual project:

1. In the editor window, select the visual dashboard scene to which you want to add the object. If the scene is not opened in the editor, then double-click the page or scene in the Project Explorer.
2. From the Palette view, select the connector object and click in the editor window or double-click the connector object. The connector object is drawn based on default properties.

   **Note:** You can also select the connector 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 connector object is drawn.

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.

If 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, 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 specified 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 application or visual dashboard. You view the embedded content in the embedded content object. An EmbeddedScene object is a container that provides the means of navigation from one scene to another. The embedded scene actually displays in the embedded scene object.

Inserting an embedded content object

An EmbeddedContent object is a container that provides the means to add external content to a visual project. The external content that can be a text file or a HTML page.

To insert an EmbeddedContent object in your visual dashboard scene:

1. In the editor window, select the visual dashboard scene to which you want to add the object. If the scene is not opened in the editor, then double-click the page or scene in the Project Explorer.
2. From the Palette view, select the EmbeddedContent object and click in the editor window or double-click the EmbeddedContent object. The EmbeddedContent object is drawn based on default properties.
Chapter 10: Working with visual projects

Note: You can also select the EmbeddedContent 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 EmbeddedContent object is drawn.

3. Click in the editor window to insert the EmbeddedContent object. The EmbeddedContent object is drawn with 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. In the EmbeddedContent mode area click ExternalData to add a separate file or DirectData to add the content as a string.

5. If you selected ExternalData specify the location of the file you want to add in the Content location field. You can click the ellipsis (...) to search for the location.

6. Specify the type of content that will be embedded in the Content type field. Select Text to display content in text format and HTML to display content in HTML format.

7. If you selected Text content type, select the required character encoding from the Character encoding list.

8. If you selected DirectData, click the ellipsis (...), add the required content in the open window, and click OK.

9. Specify the type of content that will be embedded in the Content type field. Select Text to display content in text format and HTML to display content in HTML format.

10. If you selected Text content type, select the required character encoding from the Character encoding list.

11. Click OK. An EmbeddedContent object is created. An entry is made for the object in the Project Explorer view.

Note: Embedded content objects support only a limited amount of HTML tags in the Flash rendering runtime mode. For more information, see Supported HTML tags in the Flash rendering mode.

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
Inserting a layout object using the Layout wizard

Layout objects display query results in visual formats.

To insert a layout object using the [layoutobjectname] Wizard:

2. In the opened window, select the table or query that will be run to obtain the layout object’s query result set and click Next.
3. Specify values for the remaining fields in the [layoutobjectname] Wizard. See the individual topics listed below for more detailed information on how to insert specific layout objects.

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 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. Click No and then click Next. The Specify table options page opens.
3. Specify the number of rows the table will have in the **Rows** field.
4. Specify the number of columns the table will have in the **Columns** field.
5. Select the alignment of the text in the cells from the **Alignment** combo box.
   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. Click **Yes** and then 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. Specify 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 only in Flash and HTML5 rendering modes.
12. Select the necessary **Scroll Scope** value from the list:
    - **Entire Table** to scroll the whole table.
    - **Data Only** to scroll only table data.
    - **Data and Heading** to scroll only table data and column headers.
13. Select the necessary value from **Vertical scroll** list to set the vertical scrolling.
14. Specify the width limit of the table in the **Width Limit** field. Zero (0) value means that the table width is unlimited. Any value different from zero (0) sets the maximum table width only in Flash and HTML5 rendering modes.
15. Select the necessary value from **Horizontal scroll** list to set the horizontal scrolling.
16. Click **Next**.
The Specify paging options page opens.

17. Select the **Enable paging** check box to enable paging.
   Paging allows you to display only a certain amount of rows at a time.

18. Specify the amount of rows that you want to display at a time in the **Rows per page** field.

19. Specify the first row in the result set to display in the **First displayed row** field.

20. Select the **Add navigation bar** check box to automatically create page forward and page backwards buttons that control paging actions.

21. Click **Finish**.
   The wizard closes and the table is displayed in the editor window.

**Inserting a grid**

The **Grid** object displays query results in a simple grid.

To insert a grid:

1. Double-click the **Grid** object in the **Layouts** palette of the **Palette** view.
2. In the Grid Wizard, navigate to the table or query that will provide data for the **Grid** object.
3. Click **Finish**.
   The Grid Wizard closes and the **Grid** object is displayed on the scene.

**Related topics**

**Grid**

**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. On the first page of the wizard, 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 bars 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. From the **Label field** drop-down list, select a data column 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. On the first page of the wizard, 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.

**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. On the first page of the wizard, 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 options for the pie chart page opens.
3. Click Standard and click Next.
The Select options for the pie chart page opens.
4. From the Wedge values drop-down list, select 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. From the Label field drop-down list, select 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 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. On the first page of the wizard, 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 your 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.

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 Move Right 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 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 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. From 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. From the Date drop-down list, select the name of the data column in your query result set that contains the trading date.

5. From the High drop-down list, select the name of the data column in your query result set that will contain the stocks high price on each date.

6. From 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.

7. Select the Show open price check box to display the open price for the stock.

8. From 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. From 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. From 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. From 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. From the Date drop-down list, select the name of the data column in your query result set that contains the trading date.
5. From 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. From 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. From the High drop-down list, select the name of the data column in your query result set that will contain the stocks high price on each date.
9. Select the Show low price check box to display the low price for the stock.
10. From 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. From 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. From the Date drop-down list, select 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. From the Value drop-down list, select 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. From the **Begin date** drop-down list, select 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. From the **End date** drop-down list, select 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. Click **Data contains open boundaries. Plot as Polylines.** to specify that your linear map will be plotted as polylines.
Select 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. Click **Data contains closed boundaries. Plot as polygons.** to specify that your linear map will be plotted as polygons.

   Select 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. From the **Boundary Field** drop-down list, select 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 SimpleForm object**

The **SimpleForm** object allows you to optimize data retrieving process when you need to display values from only one row in a query.

You can put several **Label** objects on a **SimpleForm** to display values from different cells of the specified query row.

To insert a SimpleForm:

1. Double-click the **SimpleForm** object in the **Layouts** palette of the **Palette** view.

2. In the SimpleForm Wizard, navigate to the table or query that will provide data for the **SimpleForm** layout and click **Finish**.
   The SimpleForm Wizard closes and the **SimpleForm** object is placed on the scene.

3. Add required **Label** objects to the **SimpleForm** layout.
   a. In the **Project Explorer** view, navigate to the **SimpleForm** layout that you inserted.
   b. In the **Palette** view, expand **Primitives** and select **Label**.
   c. Drag the **Label** object to the **Level1** node under **SimpleForm** layout in the **Project Explorer** view.

4. With the help of **Expression Designer**, specify the value to display in each **Label** object.
   a. In the **Project Explorer** view, select the **Label** object you added.
   b. In the **Properties** view, double-click the **Text** property value.
   c. In the Expression Designer window, replace default text with the name of the column that contains the value you want to display. For this, expand **Queries** → **[Query name]** → **Output columns** and select the required column.

   **Note:** **[Query name]** stands for the name of the query or table that you specified for the **SimpleForm** layout.

5. Click **OK** to close the Expression Designer window.

   The **SimpleForm** layout with all required labels is added to the scene. You can check the result in Runtime mode.

**Related topics**

*SimpleForm*
Inserting a matrix

Matrices display data in a two-dimensional array of cells.

To insert a matrix:

1. Double-click the **Matrix** object in the **Layouts** palette of the **Palette** view. The Matrix 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 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.
10. Double-click the created object and add any controls (for example, **labels**) with the necessary values to display them in the matrix.

Inserting a spiral

The **Spiral** object displays data in an infinite inward spiral, in which each data point becomes smaller.

The Spiral layout is useful for displaying data that is not mathematically related. Zooming in on the Spiral object reveals more data further into the spiral, and zooming out shows more data on the outer portion of the spiral.

To insert a spiral:

1. Double-click the **Spiral** object in the **Layouts** palette of the **Palette** view.
2. In the Spiral Wizard, navigate to the table or query that will provide data for the **Spiral** object.
3. Click **Finish**. The Spiral Wizard closes and the Spiral object is displayed on the scene.

You can customize the look and behavior of a Spiral object by modifying its properties and properties of its supporting objects: DataTemplate and Ellipse.

**Related topics**

Spiral
DataTemplate
Ellipse

**Working with nodes of TreeChart and OrganizationChart**

You can insert **TreeChart** or **OrganizationChart** into a visual dashboard or application and expand or collapse its nodes.

**Inserting an organization chart**

Organization charts display data in a tree structure that reads from top to bottom and is typically associated with a personnel hierarchy.

To insert an organization chart:

1. Double-click the **OrganizationChart** object in the **Layouts** palette of the **Palette** view. The OrganizationChart 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 Organization chart options 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 Organization chart.
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 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. From 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 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. From 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.
10. Double-click the created object and add any controls (for example, labels) with the necessary values to display them in the organization chart.

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 TreeChart object in the Layouts palette of the Palette view.
   The TreeChart 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 Tree chart options 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 Tree chart.
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 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. From 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.
9. Double-click the created object and add any controls (for example, labels) with the necessary values to display them in the tree chart.

Modifying the nodes state
You can modify the state of the TreeChart or OrganizationChart nodes on a visual dashboard or application.

### Inserting an example chart

See the following example of adding an OrganizationChart to a visual dashboard or application. This visual project is created on the basis of `DEMO.EMPLOYEE` table from the default Samples repository.

To create a visual project on the basis of `DEMO.EMPLOYEE` table:

1. Create a visual dashboard or application.
2. Insert the query with the following SQL statement:
   ```sql
   SELECT A.EMPLOYEE_ID, A.FULL_NAME, A.SUPERVISOR_ID
   FROM DEMO.EMPLOYEE A
   ```
3. Add `OrganizationChart` to the visual project.
4. On the second page of the OrganizationChart wizard specify the following chart options:
   a. Set `SUPERVISOR_ID` in the `Parent` field.
   b. Set `EMPLOYEE_ID` in the `Child` field.
   c. Set `FULL_NAME` in the `Label` field.
5. Click `Finish`. The example chart is added to the current visual project.

### Saving the chart nodes state

You can save the chart nodes state in the TreeChart or OrganizationChart on a visual dashboard or application programmatically. See an example of saving the chart nodes state on the basis of the project with `DEMO.EMPLOYEE` table.

You must have a visual dashboard or application with an added OrganizationChart or TreeChart.

To save the chart nodes state:

1. In the `DEMO.EMPLOYEE` example project, add two global parameters `ExpandedNodesSaved` and `CollapsedNodesSaved` of the Text type.
2. Add two buttons to the project: `Button1` and `Button2`.
3. For `Button1` specify the `Caption` property as `Save`.
4. Add the action `Set Values` on the `Click` event. In the Add New Action wizard specify two actions:
   b. `CollapsedNodesSaved=OrganizationChart1.DataLocator.CollapsedNodes`
5. For `Button2` specify the `Caption` property as `Load`.
6. Add the action `Set Values` on the `Click` event. In the Add New Action wizard specify two actions:
   b. `OrganizationChart1.DataLocator.CollapsedNodes=CollapsedNodesSaved`

**Note:** At first, it is necessary to bind the value of `ExpandedNodes`, and then the value of the `CollapsedNodes` property.

7. Click `Runtime` tab. Expand and collapse the nodes. Click the `Save` button to write the nodes state to the global parameters.
8. Expand and collapse the nodes, click the `Load` button to restore the tree to the state saved by clicking the `Save` button.

Expanding or collapsing all chart nodes
You can expand or collapse all chart nodes in the tree or organization charts on a visual dashboard or application programmatically. See an example of expanding or collapsing the chart nodes on the basis of the project with `DEMO.EMPLOYEE` table.

You must have a visual dashboard or application with an added OrganizationChart or TreeChart.

To expand or collapse all the chart nodes:

1. In the `DEMO.EMPLOYEE` example project, add a button and set its Caption value to *Expand All*.
2. Add the action **Set Values** on the **Click** event with the expression:
   a. `OrganizationChart1.DataLocator.ExpandedNodes=field("Query1", "EMPLOYEE_ID")`

   **Note:** The separator in the expression depends on the locale settings.

3. Add a button and set its Caption value to *Collapse All*.
4. Add the action **Set Values** on the **Click** event with the expression:
   a. `OrganizationChart1.DataLocator.CollapsedNodes=field("Query1", "EMPLOYEE_ID")`

   **Note:** The separator in the expression depends on the locale settings.

5. Click **Runtime** tab and expand all chart nodes by clicking the **Expand All** and collapse them all by clicking **Collapse All**.

   **Note:** If you add the example buttons to the DataTemplate they will work only with the node connected with this DataTemplate.

Expanding or collapsing all chart nodes by selecting their labels

You can expand or collapse the current chart node in the tree or organization chart on a visual dashboard or application by selecting their labels. See an example of expanding or collapsing the chart nodes on the basis of the project with `DEMO.EMPLOYEE` table.

You must have a visual dashboard or application with an added OrganizationChart or TreeChart.

To expand or collapse a chart node:

1. In the `DEMO.EMPLOYEE` example project, in the **Project Explorer** view expand the **OrganizationChart1** component, **DataTemplate1(Query1)**, **Leve11** and select **Label1**.
2. Add the **Set Values** action for the **Click** event of the **Label 1**: 

3. Click **Runtime** tab and expand or collapse chart nodes by clicking their labels.

Expanding to a particular chart node

You can expand a tree or organization chart to a particular chart node on a visual dashboard or application programmatically. See an example of expanding or collapsing the chart nodes on the basis of the project with `DEMO.EMPLOYEE` table.

You must have a visual dashboard or application with an added OrganizationChart or TreeChart.
To expand a chart to a particular node:

1. In the DEMO.EMPLOYEE example project, add the Combo object.
2. In the Combo wizard, specify the Query1 that is used for the OrganizationChart1.
3. Add EMPLOYEE_ID column values to the Combo.
4. Expand OrganizationChart1 component, DataTemplate1(Query1), Level1, and then select Label1.
5. Modify its Frame.FillStyle.FillColor property by adding an expression:
   
   ```
   =if(EMPLOYEE_ID=Combo1.Selection,#F00000,#78b1ea)
   ```
   To highlight the node after selecting it.

   **Note:** The separator in the expression depends on the locale settings.

6. Add the Set Values action for the Change event of the Combo object:
   
   a. OrganizationChart1.DataLocator.CollapsedNodes=field("Query1", "EMPLOYEE_ID")

7. Click Runtime tab. Select the required IDs from the Combo object, the proper nodes will be expanded and highlighted with red.

---

**Inserting a dial**

Dials display points of data as a range between two numbers on a metered scale.

To insert a dial:

1. Double-click the Dial object in the Layouts palette of the Palette view.
   The Dial Wizard opens.
2. Specify the minimum value that is displayed on the dial in the Min field.
3. Specify the maximum value that is displayed on the dial in the Max field.
4. Specify the starting angle of the dial in the Start field. 
   For example, specifying -90 will display a horizontal line that stretches from the center of the dial to the left of the canvas.
5. Specify the ending angle of the dial in the End field. 
   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 Add New Entry.
   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 field and click Delete Entry.
11. Click Finish. 
    The Dial Wizard closes and the new dial object is displayed on the dashboard canvas.
Working with Google map objects

The Map palette contains map objects that you can use to display Google maps with different components on it.

**Note:** Google Maps are deprecated in CorVu V2.2. Although CorVu V2.2 will support existing Google Map objects, you must use the Open Street map object for any new maps created using CorVu V2.2. See Working with map objects for information about using the Open Street maps.

The following map objects are available:

- GoogleMap
- GoogleMapMarker
- GoogleMapPolyline
- GoogleMapPolygon
- MapObject

The GoogleMap object can be used only in Flash-based dashboards to display result sets as geographic data points.

**Related topics**

GoogleMap

**Inserting a GoogleMapObject**

You can insert a GoogleMapObject to display query results as geographic map object.

**Note:** The GoogleMapObject can be rendered in Flash-mode only.

To insert a GoogleMapObject object:

1. Double-click the GoogleMap object in the Map palette of the Palette view.
   
   The GoogleMap Wizard opens

2. Select the desired GoogleMap viewpoints in the Viewpoints area to specify the start position of the GoogleMap and its zoom level.
   
   - Select Auto check box if you want a viewpoint to be set automatically. All the added objects will be automatically inserted in the map.
   
   - Clear Auto check box to make Latitude field and Longitude field available. So you can enter the necessary values for the viewpoints to set the map center coordinates.
   
   - Select the necessary Zoom level from the list, to set the necessary zoom value.
   
   - Select Fixed check box to make the set values fixed so a user will not be able to scroll the map or zoom it.

   **Note:** If you select the Fixed check box, the Position, Zoom, Navigation and Overview map controls are disabled.

3. Select the necessary controls from the GoogleMap controls area to add them to the map object. The list of the available ones represents the standard GoogleMap controls. Click Next.

4. Select the available map types by selecting the check boxes in the Available map types area and set the default map type from the Map type by default list. The specified map type will be loaded as the default.

5. Click Next.

6. Click Object to add GoogleMapObject component to the map and click Next.
7. 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 Create GoogleMap Object page opens.

8. Specify the GoogleMapObject properties by selecting the Tooltip field to set the column which is used as a tooltip at the map, Bounds field to set the boundaries of GoogleMapObject on the map and Content field to set the column with displayed content, for example an image..

9. Click Finish.

The wizard closes, and the GoogleMap object is displayed in the editor window.

Note: To open the inserted GoogleMap object in CorVu Web Pro you must obtain and specify the correct Google Map API Key.

Related topics
GoogleMap
Setting dashboard runtime preferences

Inserting a GoogleMapMarker
You can insert a GoogleMapMarker to display query results as a geographic map marker.

GoogleMapMarkers identify points on the map. They are designed to be interactive.

Note: The GoogleMapMarker object can be rendered in Flash-mode only.

To insert a GoogleMapMarker:

1. Double-click the GoogleMap object in the Map palette of the Palette view.

   The GoogleMap Wizard opens

2. Select the desired GoogleMap viewpoints in the Viewpoints area to specify the start position of the GoogleMap and its zoom level.

   ▪ Select Auto check box if you want a viewpoint to be set automatically. All the added objects will be automatically inserted in the map.

   ▪ Clear Auto check box to make Latitude field and Longitude field available. So you can enter the necessary values for the viewpoints to set the map center coordinates.

   ▪ Select the necessary Zoom level from the list, to set the necessary zoom value.

   ▪ Select Fixed check box to make the set values fixed so a user will not be able to scroll the map or zoom it.

   Note: If you select the Fixed check box the Position, Zoom, Navigation and Overview map controls are disabled.

3. Select the necessary controls from the GoogleMap controls area to add them to the map object.

   The list of the available ones represents the standard GoogleMap controls. Click Next.

4. Select the available map types by selecting the check boxes in the Available map types area and set the default map type from the Map type by default list. The specified map type will be loaded as the default.

5. Click Next.

6. Click Marker to add GoogleMapMarker component to the map and click Next.

7. 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 Define marker’s properties page opens.

8. Specify a data column in the Latitude field list that will supply the latitudinal location of each map marker.
Chapter 10: Working with visual projects

9. Specify a data column in the **Longitude field** list that will supply the longitudinal location of each map marker.

10. Specify a data column in the **Caption** field list that will be used to populate the labels for each map marker.

11. Select the **Enable information window** check box to allow users to open an information window when they click a map marker.

12. Specify a data column in the **Information field** list that will be used to populate the information window of each map marker.

13. Click **Finish**.

The wizard closes, and the GoogleMap object is displayed in the editor window.

---

**Note:** To open the inserted GoogleMap object in CorVu Web Pro you must obtain and specify the correct Google Map API Key.

---

**Related topics**

- GoogleMapMarker
- Setting dashboard runtime preferences

---

**Inserting a GoogleMapPolyline**

You can insert a **GoogleMapPolyline** to display query results as a geographic map polyline.

**GoogleMapPolyline** objects create linear overlays on Google Map. It consists of a series of points and creates a series of line segments that connect those points in an ordered sequence.

You can define characteristics for the lines.

---

**Note:** The GoogleMapPolyline object can be rendered in Flash-mode only.

---

To insert a GoogleMapPolyline object:

1. Double-click the **GoogleMap** object in the **Map** palette of the **Palette** view.

   The GoogleMap Wizard opens

2. Select the desired GoogleMap viewpoints in the **Viewpoints** area to specify the start position of the GoogleMap and its zoom level.
   - Select **Auto** check box if you want a viewpoint to be set automatically. All the added objects will be automatically inserted in the map.
   - Clear **Auto** check box to make **Latitude field** and **Longitude field** available. So you can enter the necessary values for the viewpoints to set the map center coordinates.
   - Select the necessary **Zoom level** from the list, to set the necessary zoom value.
   - Select **Fixed** check box to make the set values fixed so a user will not be able to scroll the map or zoom it.

   **Note:** If you select the **Fixed** check box the Position, Zoom, Navigation and Overview map controls are disabled.

3. Select the necessary controls from the **GoogleMap controls** area to add them to the map object.

   The list of the available ones represents the standard GoogleMap controls. Click **Next**.

4. Select the available map types by selecting the check boxes in the **Available map types** area and set the default map type from the **Map type by default** list. The specified map type will be loaded as the default.

5. Click **Next**.
6. Click **Polyline** to add GoogleMapPolyline component to the map and click **Next**.
7. 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 Create GoogleMap Polyline page opens.
8. Select **Direct** check box to define a polyline in latitude and longitude and click **Next**.
   a. Specify the direct GoogleMapPolyline properties by selecting the **Tooltip** field to set the column which is used as a ToolTip at the map.
   b. Specify the **Vertices field** to set the column to be used as a source vertex and **Point format** to set the way of data interpretation Latitude-Longitude or Longitude-Latitude.

9. Select **Encoded** check box to define a polyline in GoogleMap encoded format and click **Next**.
   a. Specify the encoded GoogleMapPolyline properties by selecting the **Tooltip field** to set the column which is used as a tooltip at the map.
   b. Specify a series of points within a Polyline to build the line: **Points field**, **Levels field**, **NumLevels field**, **ZoomFactor field**.

10. Click **Finish**. The wizard closes, and the GoogleMap object is displayed in the editor window.

**Note:** To open the inserted GoogleMap object in CorVu Web Pro you must obtain and specify the correct Google Map API Key.

**Related topics**
- GoogleMapPolyline
- Setting dashboard runtime preferences

**Inserting a GoogleMapPolygon**

You can insert a **GoogleMapPolygon** object in your visual dashboard to display query results as geographic map polygon.

Like the GoogleMapPolyline object, the GoogleMapPolygon consists of a series of sequential data points. However, instead of being open-ended, polygons are designed to define regions within a closed loop.

As with GoogleMapPolylines, you can define characteristics for the edge of the polygon (the "line") and the fill area within the enclosed region.

**Note:** The GoogleMapPolygon object can only be rendered in Flash-mode.

To insert a GoogleMapPolygon object:

1. Double-click the **GoogleMap** object in the **Map** palette of the **Palette** view.

   The GoogleMap Wizard opens
2. Select the desired GoogleMap viewpoints in the **Viewpoints** area to specify the start position of the GoogleMap and its zoom level.
   - Select **Auto** check box if you want a viewpoint to be set automatically. All the added objects will be automatically inserted in the map.
   - Clear **Auto** check box to make **Latitude field** and **Longitude field** available. So you can enter the necessary values for the viewpoints to set the map center coordinates.
   - Select the necessary **Zoom level** from the list, to set the necessary zoom value.
   - Select **Fixed** check box to make the set values fixed so a user will not be able to scroll the map or zoom it.
Chapter 10: Working with visual projects

**Note:** If you select the **Fixed** check box the Position, Zoom, Navigation and Overview map controls are disabled.

3. Select the necessary controls from the **GoogleMap controls** area to add them to the map object. The list of the available ones represents the standard GoogleMap controls. Click **Next**.

4. Select the available map types by selecting the check boxes in the **Available map types** area and set the default map type from the **Map type by default** list. The specified map type will be loaded as the default.

5. Click **Next**.

6. Click **Polygon** to add the GoogleMapPolygon to the map and click **Next**.

7. 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 Create GoogleMap Polygon page opens.

8. Select **Direct** check box to define a polygon in latitude and longitude and click **Next**.

   a. Specify the direct GoogleMapPolygon properties by selecting the **Tooltip** field to set the column which is used as a ToolTip at the map.

   b. Specify the **Vertices field** to set the column to be used as a source vertex and **Point format** to set the way of data interpretation Latitude-Longitude or Longitude-Latitude.

9. Select **Encoded** check box to define a polygon in GoogleMap encoded format and click **Next**.

   a. Specify the encoded GoogleMapPolygon properties by selecting the **Tooltip** field to set the column that is used as a ToolTip at the map.

   b. Specify a series of points within a polygon to build the line: **Points field, Levels field, NumLevels field, ZoomFactor field**.

10. Click **Finish**. The wizard closes, and the GoogleMap object is displayed in the editor window.

**Note:** To open the inserted GoogleMap object in CorVu Web Pro you must obtain and specify the correct Google Map API Key.

### Related topics

- GoogleMapPolygon
- Setting dashboard runtime preferences

### Creating a map

The **Map** palette contains map objects that you can use to display maps with different components on them.

The Map allows you to implement various features in your visual dashboards and application like displaying data on a map according to address information. For more information, see *Displaying data on a map according to address information retrieved from a query*.

The following map objects are available:

**Map**

Map is a basic object with the help of which you can represent query results on a geographical map. You can customize the way the query results are displayed on the map by adding the objects listed below.

**MapMarker**

MapMarker adds a marker on a geographic map to display query results.

**MapPolyline**

MapPolyline adds a polyline on a geographic map to display query results.
**MapPolygon**
MapPolygon adds a polygon on a geographic map to display query results.

**MapObject**
MapObject displays query results as geographic map object.

The Map object can be used only in Flash-based dashboards to display result sets as geographic data points.

To create a map:

1. Double-click the Map object in the Map palette of the Palette view.
   The Map Wizard opens
2. Select the necessary Map viewpoints in the Viewpoints area to specify the start position of the Map and its zoom level.
   ▪ Select Auto check box if you want a viewpoint to be set automatically. All the added objects will be automatically inserted in the map.
   ▪ Clear Auto check box to make Latitude field and Longitude field available. So you can enter the necessary values for the viewpoints to set the map center coordinates.
   ▪ Select the necessary Zoom level from the list, to set the necessary zoom value.
   ▪ Select Fixed check box to make the set values fixed so a user will not be able to scroll the map or zoom it.

   **Note:** If you select the Fixed check box, the Position and Zoom map controls are disabled.

3. Select the necessary controls from the Map controls area to add them to the map object. The list of the available ones represents the standard Map controls. Click Next.
4. On the second page of Map Wizard you must select a component to add to the map. For more information see the topics:

   **Inserting a MapObject**
   You can insert a MapObject to display query results as geographic map object.
   You must have a map on your scene to add a new component.

   **Note:** This component can be used with Flash-based dashboards only.

To insert a MapObject object:

1. On the second page of Map Wizard, click Object to add MapObject component to the map and click Next.
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 Create Map Object page opens.
3. Specify the MapObject properties by selecting the Tooltip field to set the column which is used as a tooltip at the map, Bounds field to set the boundaries of MapObject on the map and Content field to set the column with displayed content, for example an image.
4. Click Finish.

   The wizard closes, and the MapObject is displayed in the editor window.

   **Note:** You can add additional components by dragging them to a new Data Template from the Palette view.
Inserting a MapMarker

You can insert a MapMarker to display query results as a geographic map marker.

You must have a map on your scene to add a new component.

MapMarkers identify points on the map. They are designed to be interactive.

**Note:** This component can only be used with Flash-based dashboards.

To insert a MapMarker:

1. On the second page of Map Wizard, click **Marker** to add MapMarker component to the map and click **Next**.
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 Define marker’s properties page opens.
3. Specify a data column in the **Latitude field** list that will supply the latitudinal location of each map marker.
4. Specify a data column in the **Longitude field** list that will supply the longitudinal location of each map marker.
5. Specify a data column in the **Caption** field list that will be used to populate the labels for each map marker.
6. Select the **Enable information window** check box to allow users to open an information window when they click a map marker.
7. Specify a data column in the **Information field** list that will be used to populate the information window of each map marker.
8. Click **Finish**.

The wizard closes, and the Map object is displayed in the editor window.

**Note:** You can add additional components by dragging them to a new Data Template from the Palette view.

Customizing map markers

You can customize map markers by designing you own ones from the Primitives.

Add a MapMarker component on the map and set its CustomMarker value to true.

To customize a map marker:

1. In the Project Explorer window expand the MapMarker node.
2. From the Primitives of the Palette view select the AlignmentPanel and drag it to the MapMarker node.
3. From the Primitives of the Palette view select the object you want to add as a marker and drag it to the added AlignmentPanel.

The added object is inserted as a marker. You can customize it by using the properties of this object.
Inserting a MapPolyline

You can insert a MapPolyline to display query results as a geographic map polyline.

You must have a map on your scene to add a new component.

MapPolyline objects create linear overlays on Map. It consists of a series of points and creates a series of line segments that connect those points in an ordered sequence.

You can define characteristics for the lines.

**Note:** This component can only be used with Flash-based dashboards.

To insert a MapPolyline object:

1. On the second page of Map Wizard, click Polyline to add MapPolyline component to the map and click Next.
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 Create Map Polyline page opens.
3. Specify the direct MapPolyline properties by selecting the Tooltip field to set the column which is used as a hover at the map.
4. Specify the Vertices field to set the column to be used as a source vertex.
5. Select the necessary Map projection value from the list, to specify the geographic projection of Vertices.
6. If you selected EPSG: 4326 in the Map projection list, specify the Point format to set the way of data interpretation Latitude-Longitude or Longitude-Latitude.
7. Click Finish.
   The wizard closes, and the Map object is displayed in the editor window.

**Note:** You can add additional components by dragging them to a new Data Template from the Palette view.

Inserting a MapPolygon

You can insert a MapPolygon to display query results as a geographic map polyline.

You must have a map on your scene to add a new component.

MapPolygon objects create linear overlays on Map. It consists of a series of points and creates a series of line segments that connect those points in an ordered sequence.

You can define characteristics for the lines.

**Note:** This component can only be used with Flash-based dashboards.

To insert a MapPolygon object:

1. On the second page of Map Wizard, click Polyline to add MapPolygon component to the map and click Next.
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 Create Map Polyline page opens.
3. Specify the direct MapPolygon properties by selecting the **Tooltip** field to set the column which is used as a hover at the map.

4. Specify the **Vertices field** to set the column to be used as a source vertex.

5. Select the necessary **Map projection** value from the list, to specify the geographic projection of Vertices.

6. If you selected EPSG:4326 in the **Map projection** list, specify the **Point format** to set the way of data interpretation Latitude-Longitude or Longitude-Latitude.

7. Click **Finish**.

The wizard closes, and the Map object is displayed in the editor window.

---

**Note:** You can add additional components by dragging them to a new **Data Template** from the **Palette** view.

---

**Related topics**

MapPolygon

**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 a 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 the layout to display the contents of its data template.
Changing the default connector

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 it 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
Copy 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.

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:
Nesting layout objects

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 or visual application 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 the check box for individual objects or the layer itself in the **Layers** view.
   If a layer is unchecked, all of the objects that are assigned to the layer are hidden on the scene canvas. If an individual object is unchecked, 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.

- **JavaScript Modules**
  JavaScript Modules enable you to use custom JavaScript functions in visual projects.

- **Translation Tables**
  Translation tables enable dashboard designers to publish a single visual dashboard that can be viewed in multiple languages.
Working with color sequences

A color sequence is a type of global resource that can be used to modify the color properties of a visual project object based on an index value. Each index value specifies which wedge in a color sequence wheel will contain the color for an object.

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 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 Color. 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 Undo changes. 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 Move Forward. 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 Move Back. The wedge moves backward one slot in the wheel.
8. To add additional wedges to the pie, click Add. 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 Remove.
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:
Chapter 10: Working with visual projects

- Literally: You can use a specific number to indicate the index value. For example:
  
  $\text{colorSeq(Colorseq1, 4)}$
  
  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:
  
  $\text{colorSeq(Colorseq1, value1 + 5)}$
  
  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:
  
  $\text{colorSeq(Colorseq1, PointNumber())}$
  
  The PointNumber() function will be used with a color sequence named Colorseq1. The PointNumber() function assigns a unique color to the object for each row returned by a query.

Working with color maps

A color map is a type of global resource that can be used to vary the color of a visual project object depending on a value associated with the object.

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 field.
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 break point 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:

     ```plaintext
     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:

     ```plaintext
     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.
Chapter 10: Working with visual projects

- Using a function: You can use a function to generate a data value. For example:

  ```python
colorMap(Colormap1, PointNumber())
  ```

The `PointNumber()` function will be used with a color map named `Colormap1`. The `PointNumber()` function assigns a unique color to the object for each row returned by a query.

Creating a security list

Security lists can be used to tailor visual report or dashboard content to the security level of the user who is viewing the report or dashboard.

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:
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 select 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.

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.

CorVu Workstation Pro 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 and applications. 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 and applications. 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.
Chapter 10: Working with visual projects

- **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 selected, 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.

## 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 the **Picture** object in your visual project.

You can associate a stock image to the **Picture** object in one of the following ways:

1. From the **Project Explorer** view, click a stock image that you have added to the **Images** folder and drag the stock image to your visual project canvas. A picture object is created and the **ImageSource** property is displayed in the **Properties** view and is automatically updated with the name of the stock image.
From the Palette view, select the Picture object and insert it in your visual project. In the Properties view, click the ImageSource property cell. Type the name of the stock image using the stockImage() function as follows: =stockImage("My_Image") where My_Image is the name of a stock image that you have added to the Images folder for the project.

The picture in your visual project displays the specified stock image. You can then modify additional object properties such as image scaling.

**Working with Expiration Schedules**

Expiration schedules dictate when given data expires.

Expiration schedules can be set up to automatically refresh the data that is called in a visual project. If a visual project’s query is assigned an expiration schedule, the query results obtained from the query will be saved for the given amount of time that the schedule dictates. When a user runs the visual report or dashboard, the query results saved in the cache will be used and the query will not be rerun. Once the time and date set up by the schedule is reached, the query results expire, and the next time a user runs the visual report or dashboard, the query will be rerun. Expiration schedules can save time and database resources by only running queries when updated data is needed.

To create a new expiration schedule:

1. The Expiration Schedule Configuration wizard can be opened in one of the following ways:
   • From the Project Explorer view, expand the Globals node. Right-click the Expiration Schedules node. Select Insert Schedule from the context menu. The Expiration Schedule Configuration wizard opens.
   • From the Project Explorer view, expand the Globals node. Double-click the Expiration Schedules node. The Expiration Schedule Configuration wizard opens.

2. Enter a name for the expiration schedule in the Schedule name field

3. Select how often the schedule expires from the Data expires radio group. Valid choices are:
   • hourly to have the schedule expire every hour.
   • daily to have the schedule expire every day.
   • weekly to have the schedule expire every week.
   • monthly to have the schedule expire every month.
   • at specific date to have the schedule expire only at a specific date.
   • never expires to never have the schedule expire. The same query results will be used every time the project is run.
   • always expired to always have the schedule expired. The query results will never be saved and the query will be rerun every time the project is run.
   • after the visual project is closed to have the schedule expire after you close the visual project.

4. If you selected never expires or always expired, click Finish.
   The Expiration Schedule Configuration wizard closes and the new expiration schedule is saved under the Expiration Schedules node in the Project Explorer view.

5. If you selected any of the other options, click Next.
   The Set schedule properties page of the Expiration Schedule Configuration wizard is different depending on your selection.

6. Specify the Set schedule properties options and click Finish.
   The Expiration Schedule Configuration wizard closes and the new expiration schedule is saved under the Expiration Schedules node in the Project Explorer view.
Specifying hourly expiration schedule options
Hourly expiration schedules expire every hour.

To set the options for an hourly expiration schedule:

1. Open the Set schedule properties page of the Expiration Schedule Configuration wizard.
2. Specify the date that the schedule will begin to expire in the **Begin date** field.
3. Specify at which time on the specified date the schedule will begin to expire in the **Begin time** field.
4. Set the expiration interval by specifying the number of hours and minutes in the **Expiration interval** group of fields.
5. Click **Finish**.
   The Expiration Schedule Configuration wizard closes and the new expiration schedule is saved under the **Expiration Schedules** node in the **Project Explorer** view.

Specifying daily expiration schedule options
Daily expiration schedules expire on a day to day basis.

To set the options for a daily expiration schedule:

1. Open the Set schedule properties page of the Expiration Schedule Configuration wizard.
2. Specify the date that the schedule will begin to expire in the **Begin date** list box.
3. Specify at which time on the specified date that the schedule will begin to expire in the **Begin time** list box.
4. Specify an option in the **Data expires** radio group. Valid options are:
   - **daily**: the schedule will expire every day at the specified begin time, starting at the specified begin date.
   - **each**: the schedule will expire after the number of days specified in the **day** list box, at the specified begin time, starting at the specified begin date.
   - **days of week**: the schedule will expire every day of the week that is selected, at the specified begin time, starting at the specified begin date.
5. Click **Finish**.
   The Expiration Schedule Configuration wizard closes and the new expiration schedule is saved under the **Expiration Schedules** node in the **Project Explorer** view.

Specifying weekly expiration schedule options
Weekly expiration schedules expire on a week to week basis.

To set the options for a weekly expiration schedule:

1. Open the Set schedule properties page of the Expiration Schedule Configuration wizard.
2. Specify the date that the schedule will begin to expire in the **Begin date** list box.
3. Specify at which time on the specified date that the schedule will begin to expire in the **Begin time** list box.
4. Specify how many weeks from the specified begin date until the schedule expires in the **each week** list box.
5. Specify on which days of the specified week that the schedule will expire in the **days of week** check box group.
6. Click **Finish**.
   The Expiration Schedule Configuration wizard closes and the new expiration schedule is saved under the **Expiration Schedules** node in the **Project Explorer** view.
Chapter 10: Working with visual projects

Specifying monthly expiration schedule options

Monthly expiration schedules expire on a month to month basis.

To set the options for a monthly expiration schedule:

1. Open the Set schedule properties page of the Expiration Schedule Configuration wizard.
2. Specify the date that the schedule will begin to expire in the Begin date list box.
3. Specify at which time on the specified date that the schedule will begin to expire in the Begin time list box.
4. Specify an option in the Data expires radio group. Valid options are:
   - day to have the schedule expire on the day of the month specified in the day list box.
   - or to have the schedule expire at the same relative day of the month. In the first drop-down list, you specify that the schedule will expire on the first, second, third, fourth, or last day that you specify in the second drop-down list.
5. Specify which months the schedule will expire in the Select months check box group.
6. Click Finish.
   The Expiration Schedule Configuration wizard closes and the new expiration schedule is saved under the Expiration Schedules node in the Project Explorer view.

Specifying options for schedules that expire on specific dates

Expiration schedules can be set to expire only on specific dates.

1. Open the Set schedule properties page of the Expiration Schedule Configuration wizard.
2. Click the first row of the Date and time list.
   Click the ellipsis (...) on the right. The Select date and time window opens.
3. Use the left and right arrow buttons to select the month and year that the schedule will expire.
4. On the calendar, select the day of the month that the schedule will expire.
5. In the Time field, select the time at which the schedule will expire.
6. Click OK.
   The Select date and time window closes and the new date and time is stored in the Date and time list.
7. You can repeat these steps to specify multiple dates and times when the schedule will expire.
8. Click Finish.
   The Expiration Schedule Configuration wizard closes and the new expiration schedule is saved under the Expiration Schedules node in the Project Explorer view.

Working with translation tables

Translation tables enable visual designers to publish a single visual dashboard that can be viewed in multiple languages.

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 selected 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 click 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 CorVu Workstation Pro 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 **Edit with Expression Designer** on 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 Add New Action.
    
    The Add New Action window opens.
11. Double-click the **Set values** action.
    
    The Set values action parameters page opens.
12. Click Add New Entry 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 **Apply** 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 11: Working with visual portfolios

Visual portfolio is a special type of a visual project. You can use visual portfolio to assemble, manage, and demonstrate information from different CorVu data objects and OLE objects in the form of a page-based document.

Visual portfolio provides you with all the tools and features you need to quickly assemble the right data from diverse sources into a concise portfolio. You can view visual portfolio right from CorVu or presented in convenient printed form.

A visual portfolio consists of a series of pages that contain one or more of the following data objects:

- Analytical queries, visual queries, and U2 queries
- Quick reports
- Dynamarts and compound dynamarts
- Forecasts
- Drill-down paths
- Various OLE objects (for example, Microsoft Word documents and Excel spreadsheets).

Creating visual portfolios

You can create a visual portfolio and populate it with objects by using the Visual Portfolio editor in combination with the Visual Designer perspective.

Before starting a visual portfolio project, you need to plan and evaluate carefully its structure and contents. Planning includes identifying your target audience, and choosing the appropriate presentation mechanism to successfully meet your needs.

The process for creating a basic visual portfolio involves the following steps:

- Creating a visual portfolio document
- Creating cover and report pages
- Populating report pages with the required content
- Creating a Table of Contents
- Printing and publishing the visual portfolio

Creating a visual portfolio document

You can create a visual portfolio and populate it with objects by using the Visual Portfolio editor and the Visual Designer perspective.

To start working with visual portfolios, open the Visual Designer perspective, which provides you with all the required components.

To create a visual portfolio document:

1. Open the New Visual Portfolio wizard. Select File → New → Other, expand the CorVu Objects, and then select Visual Portfolio.
2. Type a unique name for the visual portfolio in the Name field.
3. Click Finish.
4. Add the content to your visual portfolio. For more information see Adding content to a visual portfolio topic.
Adding content to a visual portfolio

A new visual portfolio document is added under the Visual Portfolios node in the Project Explorer. You can view the visual portfolio structure and the objects it contains. Each visual portfolio has the following core components:

- Parameters
- Pages
- Overlay pages

**Related topics**

[Visual portfolio document](#)

### Adding content to a visual portfolio

A visual portfolio consists of a series of pages that contains different types of objects. You can add any number of pages to place all the required elements.

#### Cover page

To introduce the subject of your visual portfolio to your intended audience it is a good idea to start with a cover page. The design of a cover page usually comprises a title, company name, author details, reporting date, logo, and background images.

#### Report pages

You use the Visual Portfolio editor together with the Visual Designer perspective to create report pages and add the required content, including CorVu data objects, OLE objects, descriptive text, and images.

#### Visual portfolio objects

To add all the required data to your visual portfolio, you use a collection of the following objects available from the Palette view:

- Label
- Picture
- Chapter Header
- Table of Contents
- CorVu Object
- OLE Object

### Adding pages to a visual portfolio

Within your visual portfolio, you add pages that hold all the data objects that you want to include.

To create a visual portfolio page:

1. In the Project Explorer view, navigate to the desired visual portfolio.
2. To add a page, right-click the Pages node and select one of the following options from the pop-up menu:
Chapter 11: Working with visual portfolios

- Insert Page at Start
- Insert Page at End

3. In case you want to add a page to a specific position, expand the Pages node, right-click the existing page and, select the Insert Page Here option from the pop-up menu. The new page appears before the selected one.

4. To provide a more intuitive navigation between the different pages in the visual portfolio, rename the page. Right-click the desired page, select Rename from the pop-up menu, and then enter the new name.

Remove or rearrange visual portfolio pages as needed. For more information, see Managing visual portfolio pages, on page 338.

Related topics
Page

Adding chapter headers

Chapter headers organize the content in your visual portfolio in the same way as chapters in a book.

Use the visual portfolio editor with the Palette view to add chapter headers to your visual portfolio.

To add a chapter header:

1. Open the desired page in the editor. In the Project Explorer view, expand the Pages node and double-click the desired page.

2. From the Palette view, select the Chapter Header object, and then click in the opened visual portfolio page. The default representation of the object is drawn on the page. The name of the object is added to the Project Explorer tree under the selected page node.

3. To specify the chapter title, select the Chapter Header object on the page, click inside the selection box, and then input the desired text.

4. To place the Chapter Header object to another location on the page, drag it to the desired location.

5. Specify the chapter header text, location, alignment, and formatting options in the Properties view.
   For more information, see Chapter Header

Related topics
Chapter Header

Adding sub-headers

Sub-headers can be added to provide more detailed content and structure within chapters.

You use the Visual Portfolio editor with the Palette view to add chapter headers to your visual portfolio.

To add a sub-header:

1. Add a Chapter Header object to the page and specify its title and position.
   For more information, see Adding chapter headers, on page 334.

2. Select the added object either in the visual portfolio editor or in the Project Explorer view. The properties of the object are now displayed in the Properties view.

3. Select the sub-header level in the IndentLevel property. Available options are the numbers 0 - 5. These numbers signify the amount of indent space in the Table of Contents. The default value is 0.
Adding CorVu objects

You can add various CorVu objects to your visual portfolio to create portfolios from your own CorVu data.

To add a CorVu object to your visual portfolio:

1. In the Project Explorer view, expand the Pages node and double-click the page that you want to work with. The page opens in the Visual Portfolio editor.
2. From the Palette view, select the CorVu object, and then click somewhere in the opened visual portfolio page.
3. In the Open wizard, select the desired object to add. For information, see Opening saved objects, on page 56.
   You can add the following types of objects:
   - Analytical queries, visual queries, and U2 queries
   - Quick reports
   - Dynamarts and compound dynamarts
   - Forecasts
   - Drill-down paths
4. To run any added CorVu object, right-click it and select Execute.
5. To run all added CorVu objects, click the Execute Objects toolbar button.
6. To reload any added CorVu object, right-click it and select Reload.
7. To reload all added CorVu objects, click the Reload toolbar button.
   The object is drawn on the opened page. You can resize the object or change its location on the page.

Related topics
CorVu Object

Adding OLE objects

You can add different OLE objects such as a Microsoft Excel spreadsheets and Microsoft Word documents to your visual portfolio.

To add an OLE object:

1. Open the page that you want in the editor. In the Project Explorer view, expand the Pages node and double-click the desired page.
2. From the Palette view, select OLE Object, and then click in the opened page area.
3. In the Open window specific to your operating system, select the object you want to add.
4. To place the added OLE object to another location on the page, drag it to the required location.
   The object is drawn on the opened page. You can resize the object and change its location on the page according to your purposes.

Related topics
OLE Object
Adding images

You can add images to your visual portfolio pages by using the **Picture** object.

The following image types are supported:

- Graphics Interchange format (.gif)
- Joint Photographic Experts Group (.jpeg)
- Portable Network Graphics (.png)

To add an image to a visual portfolio:

1. Open the desired page in the editor. In the **Project Explorer** view, expand the **Pages** node and double-click the desired page.
2. From the **Palette** view, select the **Picture** object, and then click in the opened page area.
3. In the Picture dialog, select how you prefer to store the image. Two options are available:
   - **Embedded** – an image is saved within the visual portfolio document.
   - **Linked** – an image is only linked to the visual portfolio document.
4. In the **Path** field, specify the path to the desired image to add. You can also click the ellipsis (…) to navigate to and select the desired image in the specific to your operating system Open window.
   The Picture object is drawn on the page.
5. To place the image to another location on the page, drag it to the desired location.
6. Specify the precise location of the image, its size, border style, and enable or disable auto scaling in the **Properties** view.

You can also replace the image that is inserted to the **Picture** object by editing the **ImageSource** parameter value in the **Properties** view.

Related topics

- Picture

Adding text objects

Use the **Label** object to insert plain text in your visual portfolio.

To insert text into a visual portfolio:

1. In the **Project Explorer** view, expand the **Pages** node and double-click the page you want to open. The page opens in the **Visual Portfolio** editor.
2. From the **Palette** view, select the **Label** object, and then click in the opened visual portfolio page.
   The default representation of the object is drawn on the page. The name of the object is added to the **Project Explorer** tree under the selected page node.
3. Select the **Label** object on the page, click inside the selection box, and then input the desired text.
4. To place the **Label** object to another location on the page, drag it to the location you want.

Related topics

- Label
Working with overlay pages

You can use Overlay Page component to group a number of objects and put them over your visual portfolio pages as a single entity.

Assume that you need to add a specially designed frame, your company logo, and name to each page of your visual portfolio. You can manually put an image of the frame, logo, and text label with the company name to every page but this task is time consuming and can cause several issues. First, be careful with the size and location of each object in every page so that they look identically. Second, if you decide to make improvements such as replacing the frame or adding text labels, update all the pages manually.

By using overlay pages, you avoid such issues. You need to create an overlay page, add the required content (a frame, logo, and label), and link the required pages with that overlay page in the Project Explorer view. To modify the created template, you edit the overlay page; all the pages that are linked to this overlay page are updated automatically.

By default, Visual Portfolio already contains the following blank templates of overlay pages:

- **Odd** – for content that should be displayed only on odd pages.
- **Even** – for content that should be displayed only on even pages.
- **Header** – for content that should be displayed in page headers.
- **Footer** – for content that should be displayed in page footers.

You use them without creating extra overlay pages.

The following steps outline how to create and use overlay pages:

1. In the Project Explorer view, navigate to the Visual Portfolio to which you want to add an overlay page.
2. To add an overlay page, right-click the Overlay Pages node and select one of the following options from the pop-up menu:
   - Insert Page at Start
   - Insert Page at End
3. In case you want to add an overlay page to a specific position, expand the Overlay Pages node, right-click the existing page, and select Insert Page Here option from the pop-up menu. The new page is added before the selected one.
4. Populate the page with the desired content by using the Palette view.
5. To quickly link the overlay page with odd, even pages, or both, right-click the overlay page in the Project Explorer view, and select one of the following options:
   - Apply to All Pages
   - Apply to Odd Pages
   - Apply to Even Pages
6. To link the overlay page with certain visual portfolio pages, expand the Pages node, right-click a page, and then select Overlay Pages → [The required overlay page].

All the content of the overlay page displays over the visual portfolio page.
Creating the Table of Contents

You can add the Table of Contents object to your visual portfolio.

The Table of Contents updates automatically each time a Chapter Header object is added to the visual portfolio. You cannot edit the text in the Table of Contents object manually. To modify, you change the chapter header (or sub-header) text.

To add an image to a visual portfolio:

1. Open the desired page in the editor. In the Project Explorer view, expand the Pages node and double-click the desired page.
2. From the Palette view, select the Table of Contents object, and then click somewhere in the opened visual portfolio page. The Table of Contents object is drawn on the page. It includes all the chapter headers already added to your visual portfolio.
3. Resize the Table of Contents object and move it to the desired position.
4. Specify the precise location of the Table of Contents, its size, background, and border styles in the Properties view.

In the Properties view, you specify the font style for all chapter headers of different level. For more information, see Table of Contents

Managing visual portfolio pages

You can change the order of pages in your visual portfolio in the Manage Pages window.

To change the order of pages:

1. In the Project Explorer view, navigate to the visual portfolio which pages you want to reorder.
2. To open the Manage Pages window, right-click the Pages node and select Manage Pages. This window lists all the pages added to your visual portfolio.
3. Select a page and click Move Up or Move Down to change the page order.
4. To remove a page, select the page that you want to delete and click Remove.
5. Once you finished reordering pages, click OK to close the Manage Pages window.

Use the following toolbar buttons to navigate through the different pages of the visual portfolio:

Switch to the First Page
Navigates back to the first page of the visual portfolio.

Switch to the Previous Page
Navigates back to the previous page of the visual portfolio.

Switch to the Next Page
Navigates forward to the next page of the visual portfolio.

Switch to the Last Page
Navigates forward to the last page of the visual portfolio.
Managing parameters for CorVu objects

You can create a visual portfolio based on a variety of data objects such as queries, quick reports, forecasts, and drill-down paths that can request additional parameters.

You use the Manage Shared Prompts window to define how the requested parameters must be passed to the CorVu objects added to the visual portfolio.

You can group similar parameters that are used in the source objects so that all the parameters within one group receive the value that is passed to this group. By default, there are as many groups as parameters in the source objects within the action group. Each group stores one particular parameter and borrows its name. You can move parameters between the groups. If you move the last parameters from the group, this group is removed automatically because empty groups are not allowed.

To manage parameters in Cor Vu objects:

1. In the Project Explorer view, navigate to your visual portfolio and expand it.
2. To open the Manage Shared Prompts window, right-click the Parameters node, and then select Manage Parameters.
3. From the Available prompts list in the Manage Shared Prompts window, select a prompt that you want to add to a group.
4. From the Shared prompts list, select a group that you want to add the prompt to.
5. Click Add to Group. The selected prompt is added to the prompt group.
6. To remove a prompt from a group, select the prompt that you want from the Shared prompts list, and then click Detach Prompt.
7. To rename a prompt group, select the prompt group that you want to rename, and then click Rename prompt group.
8. In the Prompt Group window, specify the name and string to display, and then click OK.
9. When you finish managing shared prompts, click OK.

You can view the specified prompt values in the Prompt Values window. To open the Prompt Values window, right-click the Parameters node in the Project Explorer view and select View Prompt Values.

Printing visual portfolios

When you finish assembling and designing your visual portfolio, you can choose either to print it or publish to PDF.

To print a visual portfolio:

1. Open the visual portfolio that you want to print in the Visual Portfolio editor.
2. Select File → Print to open standard Print dialog.
3. In the Print dialog, select a printer and specify the print options, including the range of pages and number of copies.
4. Click Print.

Publishing visual portfolios to PDF

You can save visual portfolios in PDF format and then open them using another software applications.

When you finish assembling and designing your visual portfolio, you can choose either to print it or publish to PDF.
To publish a visual portfolio to PDF:

1. Open the visual portfolio that you want to export in the Visual Portfolio editor.
2. Select File → Export to open the Export dialog.
3. From the list of export destinations, select Report and then click Next.
4. Select the visual portfolio that you want to export from the list of opened documents.
5. In the Report file field, specify where to save the exported PDF file. You can also click Browse to navigate to the required directory.
6. Make sure that the Export type value is set to PDF.
7. Click Finish.

Using visual portfolio in procedures and scheduled tasks

You can use procedures and scheduled tasks to display, mail, save, and export visual portfolios.

Procedure commands

The following procedure commands are available for visual portfolios:

- DISPLAY command, on page 360
- EXPORT command:
  - EXPORT REPORT TO
- MAIL TO command:
  - MAIL ObjectName TO Address
  - MAIL REPORT TO Address
  - MAIL REPORT ReportName TO Address
- PRINT command:
  - PRINT ObjectName
  - PRINT REPORT
- SAVE command:
  - SAVE REPORT AS ReportName
- SHOW command:
  - SHOW REPORT

Scheduled tasks

For more information on working with visual portfolios in scheduled tasks see Specifying export visual portfolio to mail operation options and Specifying export visual portfolio operation options topics.
Chapter 12: Working with JavaScript Tables

You can use JavaScript Tables to collect the data from network resources and process it in the form of relational tables.

For example, you need to collect the stock exchange information from a financial portal for a certain period of time and save this information so that you can use it in CorVu as a typical relational table.

For this, you create a JavaScript Table, specify its structure, and save it to a virtual data source. In the JavaScript Table editor, you write a script that will populate the table with the required data retrieved over HTTP or HTTPS. When you run a query based on the created JavaScript Table, the script retrieves the data and saves it in a table that is used by the query.

And if it is necessary, you can use external functions and objects from different modules either saved in a repository or located on a web server.

You can run queries based on a JavaScript Table, use them to create quick reports, forecasts, or add to visual dashboards and applications.

Creating JavaScript Tables

You use the Create JavaScript Table wizard to add a new JavaScript Table and specify its structure.

To create a JavaScript Table:

1. Select File → New → Other. In the opened window select JavaScript Tables and click Next. The Create New JavaScript Table wizard opens.
2. On the Add New JavaScript Table page, specify the name for the table you create and select a visual data source where to save it. Click Next.

   **Note:** The Data Source list contains only virtual data sources from the current repository connection.

3. On the JavaScript Table Structure page, define the table structure including the list of columns, their order, value types and required parameters. You can also add comments to any column.
4. Click Finish to close the wizard.
   The JavaScript Table with the specified name and structure is added to the selected virtual data source. After closing the wizard, JavaScript Table editor opens.
5. In the JavaScript Table editor, you create scripts to populate the JavaScript Table with the required data.
6. When you finish working with script, save the changes and close the editor.

Now you can use the created JavaScript Table as a typical relational table. For example, you can run queries based on the created JavaScript Table, use them to create quick reports, forecasts, or add to visual dashboards and applications.

Writing scripts to populate JavaScript Table

Once you created a JavaScript Table and specified its structure, you need to write a script that will populate the table with the required data.

You write the script in the JavaScript Table editor according to general JavaScript syntax rules. The editor opens automatically after you close the Create New JavaScript Table wizard.
To manage the content of a JavaScript Table, you can also use the following special functions of the table global object:

- `replaceData()`
- `appendData()`
- `eraseData()`
- `setTag()`
- `getTag()`
- `include()`

To trace HTTP request and response and record logs, use `log()` function.

You can also use prompted variables in the script. A user will be prompted to enter variable values when running a query based on the created JavaScript Table. If you use a prompted variable in the script, note the following peculiarities:

- The JavaScript Table cache is ignored.
- When a user works with a JavaScript Table in CorVu Web Pro, all the retrieved data will be available to that user only.

When you finish working with the script, save the changes and close the editor.

### replaceData()

Completely replaces the content of a JavaScript Table with the data specified by a two-dimensional array of the following form: `[[<raw 1 column 1>, <raw 1 column 2>,...], [<raw 2 column 1>, <raw 2 column 2>,...],...]`.

**Note:** The structure of input array must correspond to the structure of the JavaScript Table.

**Syntax**

```javascript
replaceData(<two-dimensional array>);
```

**Example**

```javascript
var data = [['Property_1', 50], ['Property_2', 60]];
table.replaceData(data);
```

### appendData()

Inserts the data specified by a two-dimensional array to the end of the JavaScript Table.

**Syntax**

```javascript
appendData(<two-dimensional array>);
```

**Example**

```javascript
var data = [['Property_1', 50], ['Property_2', 60]];
table.appendData(data);
```
**eraseData()**

Removes JavaScript Table records that satisfy the specified SQL condition. If you call `eraseData()` function without arguments or with `null` argument, all table records are removed.

**Syntax**

```javascript
eraseData(<SQL condition>);
```

**Examples**

**Example 1**

```javascript
table.eraseData("ID < 100 AND DEPT = 15 OR SALARY IS NULL");
```

**Example 2**

```javascript
function eraseOutdated()
{
    var expirationDate = new Date();
    expirationDate.setDate(-30);
    var day = expirationDate.getDay();
    var month = expirationDate.getMonth() + 1;
    var year = expirationDate.getFullYear() + 1900;
    var strDate = year + '-' + month + '-' + day;
    table.eraseData("""Last Trade Date"" < '"' + strDate + '"'"");
}
```

**setTag()**

Links the JavaScript Table with some additional information specified by a string. This information can be retrieved by the `getTag()` method. To clear the information, use `setTag()` with `null` argument or without arguments.

**Syntax**

```javascript
setTag(<string>);
```

**Example**

```javascript
function setTableModificationTime()
{
    var currentDate = new Date();
    table.setTag(currentDate.getTime());
}
```
getTag()

Returns the additional information related to the JavaScript Table.

**Syntax**

```javascript
getTag();
```

**Example**

```javascript
function getTableModificationTime()
{
    var tag = parseInt(table.getTag());
    return tag;
}
```

include()

You can include external code in your JavaScript Table script by using the include() method.

**Note:** You can use external functions and objects only after connecting the required external module to your script with the include() method.

**Syntax**

```javascript
include(strLocation);
```

**Parameter**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>strLocation</td>
<td>Specifies either a path to a JavaScript Module in a repository or a link to external source from the Internet.</td>
</tr>
</tbody>
</table>

**Examples**

**Example 1.** External module from the Internet containing the `algorithm()` method:

```javascript
table.include("http://example.com/algorithm.js");
table.replaceData(algorithm());
```

**Example 2.** JavaScript Module from the `rsbi:/workspaces/Default/algorithm` repository contains the `algorithm` method:

```javascript
table.include("rsbi:/workspaces/Default/algorithm");
table.replaceData(algorithm());
```

log()

Displays messages in the Output view and writes them to CorVu log.

**Note:** The `log.metadatalayer.javascript` logger category must be set to `info` or higher.
Syntax

log();

Examples

log(data);
log("Data is already up to date");

Related topics

Setting log preferences

RSBIHttpRequest class

The class that is used to make HTTP requests. It has the same functions as the standard
JavaScript object XMLHttpRequest 1.0. The latter is also supported but it is recommended to use
RSBIHttpRequest.

Example

var request = new RSBIHttpRequest();
request.open('GET',url, false);

JavaScript Table Example

The following example demonstrates how to use JavaScript Tables.

Purpose

You need to collect stock exchange information from a financial portal for a certain period of time and
save this information so that you can use it in CorVu as a typical relational table. The table must not be
updated too often and store information longer than 30 days.

Preparing table structure

Before you begin writing the script, you need to make sure that the specified table structure
corresponds to the data that you need to retrieve.

Planning the script structure

According to the purpose the script must retrieve information from server, parse the data to fill the
table, save and check the last modification time, and remove outdated information.

var lastModificationDate = getTableModificationTime();
if (needUpdate(lastModificationDate))
{
    var raw_data = retrieveRawData(url);
    var data = parseData(raw_data);
    table.appendData(data);
    eraseOutdated();
    setTableModificationTime();
}
else
The following functions are used:

- **setTableModificationTime()** to attach the last modification time to the table.
- **getTableModificationTime()** to get the last table modification time.
- **needUpdate()** to check if update is required depending on the last modification time.
- **retrieveRawData()** to get the required data from the portal.
- **parseData()** to split the received string according to the table structure.
- **eraseOutdated()** to remove old records.
- **log()** to trace the response status and write information to the Output view.

**Retrieving data from the Internet**

First of all, you need to retrieve the required data from the Internet. For this, **RSBIHttpRequest** class is used:

```javascript
function getStockData(url)
{
    var request = new RSBIHttpRequest();
    request.open('GET', url, false);
    var response = '';
    request.onreadystatechange = function()
    {
        if(request.readyState === 4)
        {
            response = request.responseText;
        }
    }
    request.send(null);
    return response;
}
```

The `getStockData()` function returns the response string with all required data from the portal.

It is called from the `retrieveRawData()` function that forms an HTTP request and uses `log()` function to trace response information. You can also use prompted variables (the `symbols` variable) in script – user will be prompted to specify the values when running the query. The `&GQW_DQ` global variable is used for substitution of double quotes.

```javascript
function retrieveRawData()
{
    var symbols = &GQW_DQ&symbols&GQW_DQ;
    var url = "http://finance.yahoo.com/d/quotes.csv?s="+symbols+"&f=snl1d1t1ohgdr";
    var raw_data = getStockData(url);
    log(raw_data);
    return raw_data;
}
```

**Parsing received data**

Secondly, you need to parse the retrieved data according to your purposes so that you can fill the table with the help of **appendData()** or **replaceData()** functions. The easiest way is to split the response string:

```javascript
function parseData(raw_data)
{
    { log("Data is already up to date ");
    }
}
```
var data = [];
var rows = raw_data.split('n');
for (var i=0; i < rows.length; i++)
{  
  var row = rows[i].trim();
  if (row !== ' ')
  {  
    var cells = row.split(',');
    data.push(cells);
  }
}
return data;

Specifying data update interval

You can attach modification time (as any additional information in string format) to the table with the help of setTag() function. The getTag() function is used to retrieve this information.

1. Setting the current time (last modification) as a table tag:

   function setTableModificationTime()
   {
     var currentDate = new Date();
     table.setTag(currentDate.getTime());
   }

2. Retrieving the last modification time:

   function getTableModificationTime()
   {
     var tag = table.getTag();
     if (tag === null)
     {
       var d = new Date();
       d.setDate(-2); // to force update for the first time use
       tag = d.getTime();
     }
     else
     {
       tag = parseInt(tag);
     }
     return tag;
   }

3. Checking whether update is required:

   function needUpdate(lastModificationDate)
   {
     var yesterday = new Date();
     yesterday.setDate(-1);
     return lastModificationDate <= yesterday.getTime();
   }

Removing outdated information

You can use the eraseData() function to remove outdated information from the table:

function eraseOutdated()
{
  var expirationDate = new Date();
}
Expiration Date

expirationDate.setDate(-30);
var day = expirationDate.getDay();
var month = expirationDate.getMonth() + 1;
var year = expirationDate.getFullYear() + 1900;
var strDate = year + '-' + month + '-' + day;
table.eraseData('"Last Trade Date"<' + '' + strDate + '');
}

Displaying results

When you finish working with the script, save the changes and close the editor.

To run the query, navigate to the saved JavaScript Table in the Repository Explorer view and double-click it. Run the opened query.

Editing JavaScript Tables

You can edit the structure and the script of already created JavaScript Table.

To edit JavaScript table:

1. In the Repositories or Repository Explorer view, navigate to the required JavaScript Table.
2. To edit the structure of the JavaScript Table, right-click this table and select Properties from the popup menu.
   a. Open the Structure page in the Properties window.
   b. Change the table structure including the list of columns, their order, value types and required parameters. You can also add comments to any column.
   c. To cancel unsaved changes, click Restore Defaults.
   d. To save the updated structure of the JavaScript Table, click Apply, and then OK.

   Note: All the data will be removed from the table.

3. To edit the script of the JavaScript Table, right-click this table and select Edit Source Script from the popup menu. In the opened JavaScript Table editor, rewrite the script according to your purposes. Once you finished editing the source script, save the JavaScript Table.
Chapter 13: Text Analytics

You can use CorVu Text Analytics to respond to business problems, whether independently or in conjunction with query and analysis of fielded, numerical data.

Using Text Analytics

You can use the CorVu Text Analytics feature to analyze and interpret the relevance of words and phrases cited across multiple unstructured texts. This functionality is available for Windows operating systems only.

Text Analytics perspective

The Text Analytics perspective provides the interface from which you create and work with Text Analytics projects.

From the Text Analytics perspective you can add unstructured texts to a project and extract data analytically. The Text Analytics feature applies data mining and extraction techniques against unstructured text documents by using rules-based knowledge engine.

Opening the Text Analytics perspective

To open Text Analytics perspective, select Window > Open perspective > Other > Text Analytics.

Text Analytics perspective views

The Text Analytics perspective consists of the text viewer in the editor area and several views, which are defined below:

▪ **Highlighter**
  
  This view displays the text with the highlighted word selected in the Word Map view.

  Use this view to scroll the text with the highlighted word and find its textual context.

▪ **Sources Explorer**

  This view displays the tree structure of the files, belonging to the projects.

  Use this view to add folders to or delete folders from a project and open word matches in the Highlighter view.

▪ **Text Analytics Explorer**

  This view displays the lists of Text Analytics and Simple Text Analytics projects that can be run to analyze the specified documents.

  You use this view to manage a project.

▪ **Entities Explorer**

  If you work with Text Analytics projects, this view displays the tree structure of the entities which are found on the basis of knowledge base rules.

  If you work with Simple Text Analytics projects, this view displays the following tree branches:

  ○ Predefined dictionaries. This branch lists the built-in dictionaries if you decide to use them for text analysis process.
User dictionaries. This branch lists your own dictionaries that you specify in the New Dictionary wizard.

Use this view to select the necessary nodes and display the appropriate analysis results in the Word Map view.

• **Word Map**
  
  This view displays the text analysis results in the form of a word cloud.
  
  The level of importance of each word is indicated by font size.
  
  Use this view to select a necessary word and find it in the textual context in the **Highlighter** view or to export the results of the analysis to a file or a database.

### Customizing the knowledge base

You can customize the knowledge base by setting load values and processing sequences.

A knowledge base provides a set of rules by which the Text Analytics feature analyzes text. These rules contain lexical information and patterns that form the basis for the analysis.

Through customization you can control some of the knowledge base processing capabilities. For example, you can augment the scope of the knowledge base by adding extensions and you can enhance performance by specifying loads and processing sequences.

CorVu includes a sample knowledge base, which is in the following directory:

```
CorVu Workstation Pro install_directory\Sample AeroText KB\sample.kb
```

You can customize the sample knowledge base or any other existing knowledge base to provide a more tailored analysis of unstructured text documents.

**Note:** This task does not describe how to create or configure the knowledge base itself. If you want to modify the rules in the sample knowledge base, or if you want to create a new knowledge base, you have to use the Rocket AeroText product. For information about Rocket AeroText, go to [http://www.rocketsoftware.com/products/aerotext/descriptions](http://www.rocketsoftware.com/products/aerotext/descriptions).

It is assumed that the person performing this task has an understanding of knowledge bases.

To customize the knowledge base:

1. Open the New AeroText Knowledge Base Configuration wizard using one of the following methods:
   
   • Select **File > New > Other > AeroText Knowledge Base Configuration**
   
   • From the Text Analytics perspective, click **File > New > AeroText Knowledge Base Configuration**

2. In the **Configuration name** field, enter the name of the customized configuration and click **Finish**.
   
   By default, the Main view opens.

3. Specify the knowledge base to configure in the **Knowledge Base** field by clicking the ellipsis (…) button or by typing the path to the knowledge base file.
   
   All the tab controls are disabled until you specify the **Knowledge Base** field.

   The **Tokenizer** is a read-only field. It displays a name of the file with a tokenizer (for example, `*.dll` library for Windows).

4. Specify a predefined rule from the knowledge base in the **Rule slot** field.
Requirements for adding extensions to the knowledge base

The set of rules in the knowledge base includes actions that fill rule slots with extracted information.
- Rule slots are used to hold extracted text and sometimes related information.
- If this slot is found in the knowledge base, this rule will be included in the results and be displayed in the Entities Explorer tree.

5. Specify the content source in the **Content from** area.

By default, content is taken from the Summary. If you want the source to come from a specific slot, select Content slot and enter the slot name.

6. Use the icons in the **Knowledge Base loads** pane to set load information.

You can name a particular load in the **Name** field, add a comment in the **Comment** field, select the correct rule base and post action in the **Rulebase** and **Post action** lists.

7. Use the icons in the **Knowledge Base Processing sequence** pane to set the processing sequence.

You can name a particular sequence of actions in the **Name** field, add a comment in the **Comment** field, select the actions from the **All actions** list and add them to the **Active actions** list. The selected actions are applied to analyze the documents.

Each action in the processing sequence defines which cache or parts of a cache will be given to the pattern matcher and which rule base will be run against that cache.

A cache is a container that can store one or more records. If any caches are to be attached (i.e., pattern matching run against the contents of the cache), they are specified in the action at this level. Information obtained during rule matching is stored in a cache by the rule actions.

8. Click the **Extensions** tab to view the Extensions editor area.

You can use this editor to extend dictionaries in the knowledge base.

9. Select the necessary **Feature name** from the list.

Features define a list of possible matching tokens. Features are collections of words or other features that can be categorized or grouped pragmatically, semantically, or syntactically.

Extending the dictionaries in a knowledge base can improve the effectiveness of your text analysis, without requiring you to modify the knowledge base itself. For example, you if a new medication is released to the public, you can add the name of the medication to your medicines dictionary.

10. Select the necessary result set with the help of ellipsis (…) in the **Result set** field. You can use a local file or a repository object as a result set.

11. Specify the column from which the words for the dictionary will be selected from the **Source column** list.

12. In the **Preview** area you see the first 50 items of the added dictionary.

13. Close the editor and save the created configuration to a file.

You have a customized Knowledge Base configuration. Now you can begin using the Text Analytics feature.

**Requirements for adding extensions to the knowledge base**

You can add extensions to the knowledge base dictionaries. Adding extensions can increase the scope of entity extraction performed by Text Analytics feature.

A dictionary is a list of words or word combinations, usually grouped by different topics.

If you add extensions the knowledge base dictionary, the extensions must meet the following specifications:

- An extension must not begin with a numeral
Creating Text Analytics projects

You can create a Text Analytics project using the New Text Analytics Project wizard. The project serves as a container for documents to be analyzed by the Text Analytics feature.

Before creating a Text Analytics project, you must create a knowledge base configuration file.

If any of the text to be analyzed exists in Microsoft Word documents (.doc or .docx) or in PDF files, you must have Microsoft Word and the Adobe Acrobat Reader installed on your computer in order for Text Analytics to analyze the text.

To create a Text Analytics project:

1. Open the New Text Analytics Project wizard in one of the following ways:
   - Click the New Text Analytics Project button ( ) on the toolbar.
   - From the Text Analytics perspective, click File → New → Text Analytics Project.
   - Select File → New → Other to open the New wizard. From the CorVu Objects folder select Text Analytics Project.
   - From the Text Analytics perspective, click New Text Analytics Project button ( ) on the Text Analytics Explorer view.
2. Specify the Text Analytics project name in the Name field.
3. In the Sources to analyze area, you can double-click Folders or Result sets to specify a folder or result set objects that contains the data you want to analyze.
   
   **Note:** You will be able to specify additional folders from the Sources Explorer view or in the Edit Text Analytics project window.

4. You can also click Add source button to open New Input Source wizard:
5. If you want to add files from a local folder select System folder, click Next. The New Input Folder page opens.
   a. Browse the necessary directory with the help of ellipsis (…) button on your local disk.
   b. Select Include sub-folders check to box analyze files in all sub-folders. Click Finish.
6. If you want to add a result set, select Result set from the list and click Next. The New Input Result Set page opens. Enter a path to the folder or use the ellipsis (…) to navigate to the folder. Click Next.
   a. On the second page of New Input Result Set wizard specify the ID column, which is a unique result set row identifier.
   b. Select the Label column from the list to specify the displaying name of the rows in the Sources Explorer.
   c. Select Source column from the list to specify the source text for the analysis. If you select a binary column as a source, the Encoding field becomes available and you can select the necessary encoding for a column.
   d. In the Preview area you see the first lines of the text from the added file which will be analyzed. Click Finish.
7. In the **Options** area select the **Wrap text** check box if you want each line to fit in the viewable editor window and allow the text to be read from top to bottom without any horizontal scrolling. This option applies to .txt files only.

8. Select the **Save result set content files** check box to save the temporary result set files together with the project files. If this check box is cleared the result set files will be checked out every time when the project is opened.

9. Select **Automatically highlight results if possible** check box to automatically highlight the search results in .doc, .docx, and PDF files on when they are opened. Results for HTML and plain text files are always highlighted.

**Note:** In .doc, .docx, and PDF files the search is performed like the simple search in Adobe Acrobat Reader and Microsoft Word applications.

10. In the New Input Source wizard click **Next**.

    In the Knowledge Base configuration window, specify the path to the knowledge base configuration file in the **Configuration** field.

    When specified, the configuration file provides metadata for Text Analytics project, including the lists of available loads and processing sequences.

    If this metadata is imported successfully, the **Load** and **Processing sequence** lists become available.

    **Note:** You can click **Browse** to locate the knowledge base configuration file.

11. From the **Load** list, select the required load that is used for the analysis. A named setting that determines how documents will be loaded into Text Analyzer.

12. From the **Processing sequence** list, select the required knowledge base processing sequence. This sequence of actions will be performed to every text file under analysis.

13. Select the **Run immediately** check box to run the analysis of the current project immediately.

14. Click **Finish** to close the New Text Analytics Project wizard.

**Note:** When the wizard closes you are prompted to switch to **Text Analytics** perspective if it is not opened yet. Select the **Remember my decision** check box to remember your choice and prevent opening the Switch perspective window again.

---

### Editing Text Analytics project

You can edit a Text Analytics project using the Edit Text Analytics Project window.

To edit a Text Analytics project:

1. From the **Text Analytics** perspective, right-click the required Text Analytics project in the **Text Analytics Explorer** view, and then select **Edit** from the pop-up menu.

2. In the **Sources to analyze** list, specify the sources that contain files to be analyzed. You can add and remove folders from the list by clicking **Add source** and **Remove source**.

**Note:** If you remove a source from the project, the analysis results of files contained in this source are also removed from the **Entities Explorer** view.
3. You can also modify source files by clicking **Edit source**. In the Edit Text Analytics Project dialog specify the necessary parameters and click **OK**.

4. Select **Wrap text** check box if you want each line to fit in the viewable editor window, allowing text to be read from top to bottom without any horizontal scrolling.

5. Select **Save result set content files** check box to save the temporary result set files with the project files. If this check box is cleared the result set files will be checked out every time when the project is opened.

6. Specify the path to the knowledge base configuration file in the **Configuration** field on the **Knowledge Base setting** page.

   **Note:** You can click **Browse** to locate the knowledge base configuration file.

7. From the **Load** list, select the required load that is used for the analysis. A named setting that determines how documents will be loaded into Text Analyzer.

8. From the **Processing sequence** list, select the required knowledge base processing sequence. This sequence of actions will be performed to every text file under analysis.

9. Click **OK** to close the Edit Text Analytics Project window.

   Depending on the changes you apply to the Text Analytics project, you might be prompted to repeat the analysis of the text documents. If you have added a folder to the project, you are prompted to analyze the files contained in this folder. If you have changed the knowledge base settings, you are prompted to re-execute the project.

### Analyzing documents in a Text Analytics project

You use the Text Analytics projects to analyze unstructured text documents and reconcile the necessary information cited across these documents.

The work with Text Analytics project involves the usage of the **Text Analytics** perspective that provides four views and a text viewer.

To analyze documents in a Text Analytics project:

1. Run the Text Analytics project that you have created.

   To do this, select the required project from the **Text Analytics Explorer** view, and then click **Run** toolbar button in the **Text Analytics Explorer** view.

   • If you have opened an existing project, you do not have to run it again unless its structure has been edited since the last execution time. The files that have been already analyzed are marked with a green dot ( verde).

   • If the file has been modified after the last analysis, it is marked with a yellow dot ( amarillo).

   • If the file has been added after last running the project, it is marked with a red dot ( rojo). All the project files are marked with red in case after its last analysis the knowledge base settings have been modified.

2. Use the **Entities Explorer** view to observe the analysis results tree. The nodes of this tree represent structure of entities that have been found and used in the analyzed files.

   An entity is a representation of people, places, organizations, and time expressions found in a text document. They are grouped / categorized by shared characteristics and are analyzed grammatically, semantically, or pragmatically. The entities refer to feature that can be extended by adding terms to them.
The number of unique elements found in these files according to a specific entity is shown in brackets after the name of this entity.

3. You can study the graphical representation of word usage statistic with the help of the **Word Map** view. To do this, double-click the required entity in the **Entities Explorer** view. The Word Map displays the words and word combinations found in analyzed files on the basis of the selected entity.

4. Use the **Word count** slider to change the number of words and word combinations the Word Map displays. The value can vary from 1 to maximum value that you have specified in the Word Map Setting window.

**Note:** The default maximum value is set to 100.

5. If you want to continue working with any of displayed words or phrases, select the required element on the Word Map. The specified element is highlighted with black color. The **Sources Explorer** view displays only those files that contain the specified element.

**Note:** You can search for specific element displayed on the Word Map, by typing it in the **Search** field.

6. Use **Previous** and **Next** buttons to page the files of a particular source if there are more than 40 of them in it.

7. In the **Sources Explorer** view, double-click the file that you want to observe. The file opens in the **Highlighter** at the position of the first occurrence of the word or phrase that you have selected in the **Word Map** view. Every instance of this selected element is highlighted.

8. Use the navigation buttons on the toolbar to scroll the document to the positions of first, previous, next, or last occurrence of the selected element.

9. Select **Wrap text** check box on the toolbar if you want each line to fit in the viewable editor window, allowing text to be read from top to bottom without any horizontal scrolling. If this check box is cleared the text in the **Highlighter** will have its original formatting.

### Specifying Word Map view settings

You can modify **Word Map** view settings to change the representation of the analysis results in the view.

To specify Word Map view settings:

1. Click the **Word Map settings** icon ( ) in the Word Map view. Word Map settings dialog opens.

2. Select the rotation from the **Rotation** list. The available rotations are:
   - **Horizontal** sets a horizontal rotation of the words in the cloud
   - **Vertical** sets a vertical rotation of the words in the cloud
   - **Both** sets both horizontal and vertical rotations of the words in the cloud
   - **Any** sets an arbitrary rotation of the words in the cloud

3. From the **Color** list select the color palette for the word cloud. The available palettes are:
   - **Random RGB** sets a wide range of colors and their shades
   - **Random Palette** sets 8 preset colors

4. Select the font from the **Font** list of the available ones.
5. Set the maximum words amount to be displayed in the view by specifying it in the **Max words** field. The default value is 100.

6. Select the **Compact** check box to represent the word in the cloud more neatly.

   **Note:** It takes more time to generate a compact word map.

7. Click **Defaults** to restore the default settings values.

8. Click **OK** to save the changes.

9. Click the **Generate** icon ( ) in the **Word Map** view to apply the settings and refresh **Word Map** view.

### Exporting Word Map results

You can export the results of the analysis from the Word Map view to a file or database.

To export the analysis results:

1. Click the **Export Word Map** ( ) icon from the toolbar in the **Word Map** view.

2. Select **Use word count filter** check box to limit the amount of the exported words by the value of **Word count** slider.

3. In the **Export content** select the button that corresponds to what you want to export.
   a. Select **Export image** to export the results as an image to a file.
      1. In the **Path** field, select the necessary path to save the image.
      2. From the **Format** list, select the necessary image format and click **Finish**.
         The image will be exported to the specified file.
   b. Select **Export results** to export the words and their weights to file or database.
      1. Click **File system** to export the Word Map results data to a file. The fields that are required to export Word Map results to a file system are displayed.
      2. Click **Database** to export the Word Map results data to a database. The fields that are required to export Word Map results to a database are displayed.

   **Note:** The steps for Exporting Word Map results to a file or database are the same as exporting query results to a file or database. See **Exporting query results to a database** and **Exporting query results to a file** for a description of the steps.
Chapter 14: 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 CorVu 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 or to a file. All commands issued through procedures are governed by your resource limits. Procedures can contain:

- any CorVu procedure command
- comment lines
- blank lines
- RUN commands that run other procedures or queries
- substitution variables

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 → Other → 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. You can schedule a procedure to run at another time using the CorVu task scheduler or your local scheduler.

Procedure commands

Each procedure command is described below. Descriptions include syntax diagrams and usage examples.

Interface differences

Each CorVu 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 choose at least one from the list.

>>> DISPLAY --- QUERY ----------------------<<
  +- PROC ---+
  +- REPORT --

>>> DISPLAY --- QUERY ----------------------<<
  +- PROC ---+
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.

```plaintext
>>>-- ERASE --- ObjectName ------------<<
  +- QUERY +-  
  +- PROC +-  
  +- TABLE +-  

>>>-- ERASE --- ObjectName ------------<<
  +- QUERY +-  
  +- PROC +-  
  +- TABLE +-  
```

Sometimes two values are separated by a forward slash (/). This indicates that one of the two values must be entered:

```plaintext
-------------------------------------<<
  +- ( CONFIRM = YES/NO +-  
```

**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**

```plaintext
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 may 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.
Chapter 14: Working with procedures

Syntax

>>>-- CONNECT --------------- TO ServerName ---------------------------<<
  +- UserName -->  (PASSWORD=password

<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=saless

DISPLAY command

The DISPLAY command shows an object derived from the content in an open window (such as query results), or an object saved in the repository.

Syntax

To display an object:

>>>-- DISPLAY -------------- ObjectName ------------<<
  +- QUERY -----+          (+- &VVariable = Value -----+
  +- PROC -----+
  +- TABLE -----+
  +- REPORT ----+
  +- DASHBOARD++

>>>-- DISPLAY -------------- ObjectName ------------<<
  +- QUERY -----+          (+- &VVariable = Value -----+
  +- PROC -----+
  +- TABLE -----+

To display an object derived from the content in the last open window:

>>>-- DISPLAY ------------ QUERY -------------<<
  +- PROC ----+
  +- REPORT +-+

>>>-- DISPLAY ------------ QUERY -------------<<
  +- PROC ----+

In the DISPLAY command you can provide either the object name or the full object key of the object you want to display.

If you want to display the object that is stored in the same repository folder with the procedure you can use only the object name in the DISPLAY command. For example, the Q.STAFF query is stored...
in the same repository folder as the procedure, then you can use DISPLAY Q.STAFF to display the Q.STAFF query.

If you want to display the object that is neither stored in the procedure nor saved to the same repository folder with the procedure you must use the full object key in the DISPLAY command. For example, DISPLAY "rsbi:.workspaces/Traditional DBA View/Query1".

**Note:** To view an object key, select the required object in the Repository Explorer or Workspaces views. The key value is displayed in the Properties view. If the key contains spaces, make sure it is enclosed in double quotes, as it 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 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 DISPLAY command. If there are variables in the query or procedure that are not given values on the DISPLAY command, and are not global variables, the user will be prompted for the values. When you include variable assignments in a procedure, you must use two ampersands to prevent variable substitution before the procedure is run.</td>
</tr>
</tbody>
</table>

**Examples**

```
DISPLAY USER1.QUERY1

DISPLAY "rsbi:.workspaces/Traditional DBA View/Query1"

DISPLAY QUERY
```

**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. It is ignored when TYPE=INSERT. There is no default.</td>
</tr>
</tbody>
</table>
Examples

DRAW Q.STAFF (TYPE=SELECT

EDIT command

The \texttt{EDIT} command will display an object derived from the content in an open editor window (such as queries, procedures, and tables), or an object saved in the repository.

Syntax

To edit an object in the editor window:

```
>>> EDIT ------------------ObjectName -----------------------------<<
  +- QUERY ------+            ( +- &&Variable = Value ----+
  +- PROC ------+
  +- TABLE ------+

>>> EDIT ------------------ObjectName -----------------------------<<
  +- QUERY ------+            ( +- &&Variable = Value ----+
  +- PROC ------+
  +- TABLE ------+
```

To edit an object derived from the content in the last opened window:

```
>>> EDIT -----------+- QUERY --+------------<<
  +- PROC ---+

>>> EDIT -----------+- QUERY --+------------<<
  +- PROC ---+
```

In the \texttt{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 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 \texttt{Q.STAFF} query is stored in the same repository folder as the procedure, then you could use \texttt{EDIT Q.STAFF} to edit the \texttt{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 \texttt{EDIT} command. For example, \texttt{EDIT "rsbi:/.workspaces/Traditional DBA View/Query1"}.

\textbf{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 \texttt{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>
</tbody>
</table>
ERASE command

The ERASE command removes an object (query, procedure, or table) from the database.

Syntax

```plaintext
>>  -- ERASE ----------- ObjectName -----------------------<<
    +- QUERY --+            +- ( CONFIRM = YES/NO  -+
    +- 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 window 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

```plaintext
>>> Execute CommandLine ------------------------<<
```
Parameter | Description
--- | ---
CommandLine | An application command line, object name, or URL to launch. The command is launched and execution of the procedure continues without waiting for completion of the command.

Example

```
EXECUTE c:\programs\notepad.exe
```

## EXPORT command

The **EXPORT** command saves objects that are in the database or opened in the editor (in temporary storage).

### Syntax

**To export an object from a database to a file:**

```plaintext
>>>-- EXPORT --- QUERY --- ObjectName TO FileName ------------------------<<
   (+- PROC ---+                         +- (CONFIRM = YES/NO ------+
   (+- SAVEATSERVER = YES/NO -+

>>>-- EXPORT FORM ObjectName TO FileName ----------------------------------<<
   ( +- CONFIRM = YES/NO ------------+
   ++ LANGUAGE = ENGLISH/SESSION ---+
   ++ SAVEATSERVER = YES/NO -------+

>>>-- EXPORT TABLE ObjectName TO FileName --------------------------------------<<
   ( +- CONFIRM = YES/NO ---------------------+
   ++ DATAFORMAT = CSV/DBF/HTML/IXF/PDF/SHP/TEXT/WQML/XLS/XML --------------------+
   ++ OUTPUTMODE = BINARY/CHARACTER/PC ------+
   ++ LOBSINFILE = YES/NO ------------------+
   ++ LOBSTO = path1;path2;... -------------+
   ++ LOBFILE = basefile1;basefile2;... ----+
   ++ CCSID=integer or Java encoding name --+
   ++ UNICODE= YES/NO ----------------------+
   ++ SAVEATSERVER = YES/NO ----------------+
```

**To export an object that is opened in the editor (in temporary storage) to a file:**

```plaintext
>>>-- EXPORT --- QUERY --- TO FileName ----------------------------------------<<
   (+- PROC ---+              +- (CONFIRM = YES/NO -+
   (+- SAVEATSERVER = YES/NO -+

>>>-- EXPORT FORM TO FileName -------------------------------------------------<<
   ( +- CONFIRM = YES/NO -----------+
   ++ LANGUAGE = ENGLISH/SESSION -+
   ++ SAVEATSERVER = YES/NO -----+

>>>-- EXPORT DATA TO FileName -------------------------------------------------<<
   ( +- CONFIRM = YES/NO ---------------------------------+
   ++ DATAFORMAT = CSV/DBF/HTML/IXF/PDF/SHP/TEXT/WQML/XLS/XML ---------------------------------------------+
```
EXPORT command

+- OUTPUTMODE = BINARY/CHARACTER/PC ------------------+
+- LOBSINFOFILE = YES/NO -----------------------------+
+- LOBSTO = path1;path2;... ------------------------+
+- LOBFILE = basefile1;basefile2;... ----------------+
+- CCSID = integer or Java encoding name --------------+
+- UNICODE = YES/NO ---------------------------------
+- MODE = GRID/RAW ----------------------------------+
+- SAVEATSERVER = YES/NO -----------------------------+

>>>-- EXPORT REPORT TO FileName ------------------------------<<

( + CONFIRM = YES/NO --------------------------------+
 + DATAFORMAT = HTML/TEXT/PDF ------------------------+
 + SPLIT = YES/NO -------------------------------------+
 + ORDER = 0/1/2 --------------------------------------+
 + WIDTH = integer -------------------------------------+
 + LENGTH = integer ------------------------------------+
 + UNITS = INCHES/CENTIMETERS/MILLIMETERS/POINTS------+
 + SEPARATOR = any HTML text --------------------------+
 + SAVEATSERVER = YES/NO -----------------------------+

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectName</td>
<td>The name of the object to export from the database.</td>
</tr>
<tr>
<td>FileName</td>
<td>The name of the file to which the object will be exported.</td>
</tr>
<tr>
<td>CONFIRM</td>
<td>Specifies whether or not to display a confirmation window before replacing an existing file. If CONFIRM is not specified or is NO, the corresponding resource limit is used.</td>
</tr>
<tr>
<td>LANGUAGE</td>
<td>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.</td>
</tr>
<tr>
<td>DATAFORMAT</td>
<td>Specifies the file format of the exported data. If you omit this parameter, the GQW_EXP_DT_FRMT global variable supplies the format to be used. You can specify in GQW_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>OUTPUTMODE</td>
<td>Specifies the output format of the data.</td>
</tr>
<tr>
<td>LOBSINFOFILE</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>
</tbody>
</table>
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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 window are supported. Any encoding names that are supported by your JVM can be used as well.</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 IXF 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>
</tbody>
</table>
| ORDER       | Controls how the report will be split into pages. Use only when exporting visual reports.  
- 0 - each visual report page will be exported into separate files.  
- 1 - a single output file contains visual report pages in "across then down" order.  
- 2 - a single output file contains visual report pages in "down then across" order. |
| WIDTH       | Specifies the width in units for a report page. Use only when exporting visual reports. |
| LENGTH      | Specifies the length in units for a report page. Use only when exporting visual reports. |
| UNITS       | Specifies the measurement units that are used in the WIDTH and HEIGHT parameters. Use only when exporting visual reports. |
| SEPARATOR   | Specifies a string that will be used as a page separator. Use quotes if necessary. 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.  
- 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. |
| SAVEATSERVER | Specifies whether or not to include the Root output directory in the export path of an object. This directory is set on Server-Side File System page of the Preferences window. In CorVu Web Pro 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

```
EXPORT QUERY USER1.QUERY TO C:\Queries\query1.qry
```

The following is an example of using SAVEATSERVER parameter. Consider the case when the root output parameter is set to C:\Temp on the Server-Side File System preferences page:

```
EXPORT QUERY USER1.QUERY TO query1.qry (SAVEATSERVER=YES
```
The result of the command will be a new file in C:\Temp\query1.qry

Related topics
Setting the directory location preference for objects generated by Export actions, on page 28

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**

```plaintext
>>>-- FORWARD MAX -----------------------<<
```

**Example**

FORWARD MAX

IMPORT command

The **IMPORT** command imports objects that are currently stored in a file and opens them in an active window (temporary storage) or saves them in the current database.

**Syntax**

To import an object from a file, open it in an active window and save it in the current database:

```plaintext
>>>-- IMPORT --- QUERY --- ObjectName FROM FileName ------------------------------<<
  (+) PROC ++
  (+) CONFIRM = YES/NO --------+
  (+) COMMENT = Text -----------+
  (+) 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):

```plaintext
>>>-- IMPORT --- QUERY --- FROM FileName -------------------------------<<
  (+) PROC ++

>>>-- IMPORT --- QUERY --- FROM FileName -------------------------------<<
  (+) PROC ++

>>>-- IMPORT -- DATA --- FROM FileName -------------------------------<<
  (+) LOBSFROM = path1;path2; --+
```
Chapter 14: Working with procedures

### Parameter Description

<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 window 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>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
```

### MAIL TO Command

The MAIL TO command sends the specified object as an email attachment.

The specified object can be in a database or opened in the editor (in temporary storage).

#### Syntax

To send an object that is stored in a database:

```
>>> MAIL ObjectName TO Address --
    QUERY
    PROC

>>> MAIL ObjectName TO Address --
    TABLE
```

To send an object that is opened in temporary storage:

```
>>> MAIL ObjectName TO Address --
    TABLE
```

---

368
MAIL TO Command

++ 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 ++
  ++ TYPE = PDF/HTML--------------------------+
  ++ 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 ---------------+
  ++ TYPE = PDF/HTML--------------------------+
  ++ METHOD = SPLIT/CONT ----------------------+

To send an object is currently opened (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 ---------------+
  ++ TYPE = PDF/HTML--------------------------+
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 parameter, the GQW_EXP_DT_FRMT global variable supplies the format to be used. You can specify in GQW_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 GQW_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 &lt;PRE&gt; tag, TEXT is for a 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>DATEFORMAT</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 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.</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 more information about Java format strings, see the Java 2 SDK, Standard Edition Documentation.</td>
</tr>
</tbody>
</table>
Parameter | Description
--- | ---
TIMEFORMAT | 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.

Note: If the format string includes spaces, enclose it in quotes. For more information about Java format strings, see the Java 2 SDK, Standard Edition Documentation.

Example

MAIL QUERY TO abc123@mail.com

MAIL TO abc@mail.com;
+jsmith@mail.com
+FROM=jsmithTest@mail.com
+CCLIST=abc123@mail.com;
+jsmith@mail.com
+SMTPSERVER=mail.company.com
+SMTPUSER=user
+SMTPPASSWORD=password
+ATTACHMENT='C:\docs\file.txt;
+C:\docs\file2.txt'

PRINT command

The PRINT command prints an object that is opened in the editor (in the temporary storage) or stored in the database.

Syntax

To print an object that is stored in a database:

```plaintext
>>> PRINT ------------ ObjectName --------------------------<<
++ QUERY +
++ PROC +
++ TABLE +
```

To print the object opened in the editor (in temporary storage):

```plaintext
>>> PRINT QUERY --------------------------<<
++ PROC +
++ (PRINTER = Printer ++

>>> PRINT REPORT --------------------------<<
{ ++ PRINTER = Printer ++
++ COPIES = NumCopies---------++
```

Parameter | Description
--- | ---
ObjectName | The name of the object to print.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRINTER</td>
<td>Specifies the name of the printer on which to print the object or report.</td>
</tr>
<tr>
<td>COPIES</td>
<td>Specifies the number of copies to print when printing the object.</td>
</tr>
</tbody>
</table>

**Example**

PRINT QUERY
PRINT REPORT

### 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:

```plaintext
>>>-- RESET GLOBAL (VarName1, VarName2,... ------------------<<
```

To reset all global variables:

```plaintext
>>>-- RESET GLOBAL ALL ----------------------------------<<
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VariableName</td>
<td>The names of the specific variables that will be deleted. You can name up to 10 variables previously set by the <code>SET GLOBAL</code> command. ALL deletes the names and values of all the variables previously set by the <code>SET GLOBAL</code> command. Variables whose names begin with DSQ are restricted, and can not be deleted.</td>
</tr>
</tbody>
</table>

**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:

```plaintext
>>>-- RUN ------------ ObjectName -------------------------------------------<<
    +- QUERY +++++ ( ++ &&Variable = Value ---------+
```
Chapter 14: Working with procedures

**Parameter** | **Description**
---|---
ObjectName | The name of the query, procedure, or dynamart to run.
&&Variable | 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, procedure, or dynamart that are not given values on the RUN command, and are not global variables, the user will be prompted for the values. When you include variable assignments in a procedure, you must use two ampersands to prevent variable substitution before the procedure is run.
CONFIRM | Specifies whether or not to display a confirmation window 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.
REFRESH | Specifies whether or not to refresh data previously saved in a dynamart. If this parameter is set to YES, the dynamart is opened in the Query editor, query is run to refresh data, and then updated data is automatically saved in the dynamart.
ROWLIMIT | Specifies the maximum number of rows to retrieve for the query.

**Example**

```
RUN QUERY USER1.QUERY1 (&&Var1=10
```
SAVE command

The **SAVE** command records the contents of an object from 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 ---------------------------------------------+)
  (+- SCOPE = NumChars ---------------------------------------------+)
  (+- ROWIDADD = YES/NO -------------------------------------------+)
  (+- ROWIDNAME = text ---------------------------------------------+)
  (+- ROWIDDISP = ALWAYS/BYDEFAULT ---------------------------------+)
  (+- ROWIDDISPnnn = EXCLUDE/CONVERT/ALWAYS/BYDEFAULT--+)
  (+- ACTION = REPLACE/APPEND -----------------------------------------+)
  (+- SPACE = database.tablespace--------------------------------------+)
  (+- MODE = GRID/RAW --------------------------------------------------+)
```

**Parameter**

<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, procedure, table, or portfolio) 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>CONFIRM</td>
<td>Specifies whether or not to display a confirmation window 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>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>
</tbody>
</table>
Parameter | Description
--- | ---
SPACE | 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.

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.
- **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.

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 CorVu session unless manually deleted.

**Syntax**

```sql
>>>-- SET GLOBAL ( VariableName = Value, ... -----<<
```

Parameter | Description
--- | ---
VariableName | The name of the global variable to set or create. Global variable names can be from 1 to 17 characters long. Variables whose names begin with DSQ are restricted, and can not be created or deleted.

Value | The value to give the global variable. The value can be from 1 to 55 characters long. The values of variables whose names begin with DSQ are restricted. See Working with Global Variables for more information.

Example

```sql
SET GLOBAL (Var1=abc, Var2=def
```
SET LOCAL command

The **SET LOCAL** command sets the values of existing local variables or creates new variables and values. Any new local variables created are valid only for the current query, report, or dashboard.

**Syntax**

```plaintext
>>>-- SET 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 or create. Local variable names can be from 1 to 17 characters long. Variables whose names begin with DSQ are restricted, and can not be created or deleted.</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

LIMIT LOCAL command

The **SET LOCAL** command limits the values that can be entered for local variables. Any new local variables created are valid only for the current query, report, or dashboard. Only the specified available values will be available in the Enter substitution variable window when you run the object.

**Syntax**

```plaintext
>>>-- 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 or create. Local variable names can be from 1 to 17 characters long. Variables whose names begin with DSQ are restricted, and can not be created or deleted.</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**

LIMIT LOCAL (Var1=abc, Var2=def
SHOW command

The SHOW command displays objects from the temporary storage and is similar to the DISPLAY command.

The SHOW REPORT and SHOW PROC commands activate the window containing the current report or procedure, respectively. The SHOWGLOBALS command opens the Global Variables window.

Syntax

```
>>>-- SHOW --- PROC -----------------------------------------------------<<
    +- GLOBALS +-<<

>>>-- SHOW REPORT --------------------------------------------------------<<
    ( VIEW = QUICK/VISUAL
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIEW</td>
<td>Specifies the appearance of the target query or report.</td>
</tr>
<tr>
<td>RESULTSET</td>
<td>Specifies the number of the result set that is to be displayed. This parameter is only available if VIEW is equal to RESULTS and the query is a stored procedure call.</td>
</tr>
</tbody>
</table>

Example

```
SHOW REPORT (VIEW=QUICK
```
Chapter 15: Working with global variables

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

For objects that use substitution variables, values are entered via a dialog 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 CorVu application. That means their values are re-initialized each time you open the CorVu application. System global variables begin with the letters GQW, GA, GEC, DSQDC, and GCP. 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 GQW, GA, GEC, DSQDC, and GCP. User global variables values are not saved when you terminate the CorVu 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 **Add**. 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 GQW, GA, GEC, GDC, and GCP. Click **OK**. The user global variable is added to the list and will be available for use during the current CorVu 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 **Remove**. 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 window.
Editing system global variable values

You use the Global Variables page of the Preferences window 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 GQW, GA, GEC, GDC, or GCP.
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 CorVu application. Their values are re-initialized each time you open the application.

The system defined global variables are divided into the following groups:

GQW global variables

Global variables whose names begin with GQW provide information about the current query environment.

The following GQW global variables are currently available:

<table>
<thead>
<tr>
<th>Name</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GQW_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>GQW_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>Name</td>
<td>Length</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>--------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GQW_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:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• zero (0) for text format</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• two (2) for HTML format</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• three (3) for CSV format</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• four (4) for IXF format</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• five (5) for dbase III files</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• six (6) for Shapefile format</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• seven (7) for WQML format</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• eight (8) for XML format</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• nine (9) for PDF format.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ten (10) for XLS format.</td>
</tr>
<tr>
<td>GQW_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>GQW_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>GQW_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>GQW_PROC_OUTPUT</td>
<td></td>
<td>Output file name for a procedure.</td>
</tr>
<tr>
<td>GQW_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>GQW_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>
</tbody>
</table>
### GA Global Variables

Global variables whose names begin with GA provide information about the current state of the query session.

The following GA global variables are available:

<table>
<thead>
<tr>
<th>Name</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GA_BATCH</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>Name</td>
<td>Length</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GA_CONNECT_ID</td>
<td>8</td>
<td>The user ID that is used to connect to the current database.</td>
</tr>
<tr>
<td>GA_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>GA_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>GA_NLF_LANG</td>
<td>1</td>
<td>National language of session. Value is &quot;E&quot; for the English language.</td>
</tr>
<tr>
<td>GA_NUM_FETCHED</td>
<td>0</td>
<td>The number of rows fetched by the current query object.</td>
</tr>
<tr>
<td>GA_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>GA_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>GA_RELEASE</td>
<td>2</td>
<td>Numeric release number of the application.</td>
</tr>
<tr>
<td>GA_REP_USER</td>
<td>8</td>
<td>If connected to a secured repository, contains the user name used to authenticate the repository connection.</td>
</tr>
<tr>
<td>GA_QUERY_MODEL</td>
<td>1</td>
<td>Model of the current query object. Value can be one (1) for relational.</td>
</tr>
<tr>
<td>GA_SYSTEM_ID</td>
<td>1</td>
<td>Current operating system. Values can be one of the following:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 8 - Windows NT and above</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 9 - Linux</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 10 - HP-UX</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 11 - AIX</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 12 - Solaris</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 13 - iSeries</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 14 - z/OS</td>
</tr>
<tr>
<td>GA_VER_RLS</td>
<td>10</td>
<td>External version and release number for the application.</td>
</tr>
</tbody>
</table>

**GEC global variables**

Global variables whose names begin with GEC control how commands and procedures are executed. The following GEC global variables are available:
### GEC_CON_ACC_RES

<table>
<thead>
<tr>
<th>Name</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GEC_CON_ACC_RES</td>
<td>1</td>
<td>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, or diagram queries), but also from other operations such as DISPLAY TABLE. Possible values are: <strong>• 0</strong> - No concurrent access resolution options on the PREPARE statement associated with the pending SQL SELECT statement are specified. This is the default value. <strong>• 1</strong> - SKIP LOCKED DATA. This value can be specified for executable SELECT statements directed to DB2 for z/OS Version 9 or later. <strong>• 2</strong> - USE CURRENTLY COMMITTED. This value can be specified for executable SELECT statements directed to DB2 for z/OS Version 10 and later. <strong>• 3</strong> - WAIT FOR OUTCOME. This value can be specified for executable SELECT statements directed to DB2 for z/OS Version 10 and later.</td>
</tr>
</tbody>
</table>

### GEC_EXTND_STG

<table>
<thead>
<tr>
<th>Name</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GEC_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. 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>
</tbody>
</table>

### GEC_LAST_RUN

<table>
<thead>
<tr>
<th>Name</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GEC_LAST_RUN</td>
<td>1</td>
<td>Specifies the set of commands that cause the LAST_USED column of the Q.OBJECT_DIRECTORY table to be updated. Possible values are: <strong>• 0</strong> - Last used is updated on any activity. <strong>• 1</strong> - Last used is updated when RUN, SAVE, or IMPORT commands are performed.</td>
</tr>
<tr>
<td>Name</td>
<td>Length</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>--------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| GEC_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:  
```sql
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:  
```sql
SET GLOBAL (DSQEC_LIST_OWNER=I_%)
```
| GEC_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). |
| GEC_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. |
| GEC_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). |

**GDC global variables**

Global variables whose names begin with GDC control how information is displayed.

The following GDC global variables are available:

<table>
<thead>
<tr>
<th>Name</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GDC_COL_LABELS</td>
<td>1</td>
<td>Specifies whether column headings will be column names or database labels. 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).</td>
</tr>
<tr>
<td>GDC_CURRENCY</td>
<td>18</td>
<td>Defines the custom currency symbol to use when the DC edit code is specified.</td>
</tr>
<tr>
<td>GDC_DISPLAY_RPT</td>
<td>1</td>
<td>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. The default value is zero (0).</td>
</tr>
</tbody>
</table>
### GCP global variables

Global variables whose names begin with GCP control the operation of the table editor. The following GCP global variables are available:

<table>
<thead>
<tr>
<th>Name</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GCP_TEDFLT</td>
<td>1</td>
<td>Defines the reserved character used in the Table Editor to indicate a default value for a column. The default value is &quot;+&quot;. This variable can also be set in the Enter default values as field found in the Preferences window.</td>
</tr>
<tr>
<td>GCP_TENULL</td>
<td>1</td>
<td>Defines the reserved character used in the Table Editor to indicate a null value for a column. The default value is &quot;.&quot;. This variable can also be set in the Enter null values as field found in the Preferences window.</td>
</tr>
</tbody>
</table>

---

Name | Length | Description
--- | -------- | ---------------------------------------------
GDC_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. |
Chapter 16: Managing your query environment

There are many ways that you can customize your query and reporting environment.

Specifying print options for query result sets and quick reports

In CorVu Workstation Pro, you can specify the page setup options that will be used when you print the object opened in the editor window.

To control the appearance of printed pages for query results and quick reports:

1. With an active object in the editor window, open the Page Setup window by selecting File → Page Setup.
2. Specify the printing options that will apply to your object.

The sets of options vary depending on the type of object you want to print. The following table shows what page options you can specify for query result sets and quick reports:

<table>
<thead>
<tr>
<th>Query result set</th>
<th>Quick report</th>
</tr>
</thead>
<tbody>
<tr>
<td>Printer and page parameters including the type and size, orientation and margins</td>
<td>Paper parameters including the type and size, orientation and margins</td>
</tr>
<tr>
<td>Header and Footer text</td>
<td>Footer text</td>
</tr>
<tr>
<td>Whether to print row and column numbers, column headers, grid lines, aggregation type, and page header and footer</td>
<td>Whether to print row and column numbers, column headers and page footer</td>
</tr>
<tr>
<td>Page order</td>
<td>Fixed groups</td>
</tr>
<tr>
<td>Fixed side and top groups</td>
<td></td>
</tr>
</tbody>
</table>

Note: You can use special codes to insert specific data to header and footer fields.

3. To make the specified values default, click Set as Default.
4. Click OK to close the Page Setup window. The options that you specified are used to control the appearance of the printed object.

Viewing resource limits

CorVu 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 CorVu 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 CorVu 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 Object Tracking page to see if CorVu object tracking is activated for your user ID.

8. Select the LOB Options page to see if you can access and retrieve data from tables containing large object data (LOB).

9. Select the Cache page to control caching options for each user.

10. Select the Object List page to define the default owner name that will be used to filter data source objects.

11. 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
- ALTER
- CALL
- COMMENT
- CREATE
- DELETE
- DROP
- EXPLAIN
- GRANT
- INSERT
- LABEL
- LOCK
- REFRESH
- RENAME
- REVOKE
- SET
- SELECT
- UPDATE

Specifying Options resource limits

The **Options** resource limits specify the levels of access you have to a data source and the data source objects.

**Options** resource limits can be set for data sources. The **Options** resource limits are as follows:

**Allow access to data source from user perspective**

This field specifies whether you have permission to access a data source from the CorVu Workstation Pro user perspective 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 CorVu Workstation Pro 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 clear, data source changes can occur without confirmation.

**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.

**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 selected, 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**
This field specifies the isolation level for a user’s queries. The isolation level specifies to what degree a query will be protected from the effects of other running applications. This option only applies for the following:

- DB2 for MVS Version 4 or later
- DB2 for OS/390 Version 5 or later
- DB2 UDB for OS/390 Version 6 or later
- DB2 Server for VM & VSE Version 5 or later

You can select one of the following isolation levels:

- **Default** specifies that the isolation level for a query will be the default specified for all applications.
- **Repeatable Read** specifies that the isolation level for a query will be Repeatable Read (RR). The RR isolation level provides the most protection from other applications.
- **All (RS)** specifies that the isolation level for a query will be Read Stability (RS).
- **Cursor Stability (CS)** specifies that the isolation level for your query will be Cursor Stability (CS). The CS isolation level provides the maximum concurrency with data integrity.
- **Change (UR)** specifies that the isolation level for your query will be Uncommitted Read (RS) which allows an application to read uncommitted data.
- **No Commit (NC)** specifies that the isolation level for your query will be No Commit (NC). An application running with an isolation level of NC can not make permanent database changes.

**Account**
This field specifies the default account information that will be sent to a data source when you connect to that data source.

**Account can be overridden**
This field specifies whether you are allowed to override the default accounting information specified in the **Account** field when connecting to a data source. If selected, you can enter a new accounting string using the **Set User Information** window. If clear, you cannot enter new
accounting information and you are prevented in some cases from accessing data source objects that are not owned by your resource limits group.

**Require account information**
This field specifies whether you must specify a valid accounting string in the **Set User Information** window before connecting to a database when this schedule is in effect.

**Allow printing**
This check box must be selected in order for you to print objects that are stored on the data source.

### Specifying Save Data resource limits

The **Save Data** resource limits specify whether you as a member of the current resource limits group and using the current schedule can save data at the database data source.

**Save Data** resource limits can be set for data sources. The **Save Data** resource limits are as follows:

**Allow save data command**
This field specifies whether you can save data at the database data source. Saving data can be extremely resource-intensive. As a result it can have a significant impact on your database and network.

**Default table space**
This field specifies the default table space name that will be used for tables created with the save data command. This field is only available if the **Allow save data command** has been selected. The syntax of the table space name that you specify must conform to the database rules for table space names. Any value that you specify is used as part of the CREATE TABLE SQL statement that runs when you save data to a new table. Entering an asterisk (*) specifies that the default table space will be taken from a user's profile in the `RDBI.PROFILE_TABLE`. The default table space in the `RDBI.PROFILE_TABLE` is specified by the CorVu Workstation Pro administrator when your user 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.

### 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 box 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 the data about performing commands against an object.

**Note:** In cases where the object is saved to the repository, the data is stored in the column `LAST_USED` of the table `RSBI.OBJECTS`.

The following options are available:
- Update when object is accessed (legacy mode)
  The data is updated by any of the following commands performed on the object:
    - DISPLAY
    - EXPORT
    - IMPORT
    - PRINT
    - RUN
    - SAVE
- Update when object is run, saved or imported
  The data is updated after performing RUN, SAVE or IMPORT commands.
- Update only when object is run
  The data is updated only after performing RUN command.

**Last Used timestamp options can be overridden**
By default, the check box is not selected and it shows that GEC_LAST_RUN variable value is ignored.

If the check box is selected, then by changing the variable value a user can override the resource limit.

Specifying LOB Options resource limits

The **LOB Options** resource limits control the ability to access, retrieve, and save data to and from tables containing large objects (LOBs).

A large object (LOB) is a DB2 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.
- **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 group of fields specifies the time period the query result set data will remain in cache. You can set the time period by specifying the number of days, hours, and minutes. After this period, the cached query result set data will be updated.

**Cache options can be overridden**

This field specifies whether the cache options can be overridden.

### Specifying Object List resource limits

The **Cache** resource limits specifies default and allowed owner names for data sources.

With **Object List** resource limits, you can define the default owner name that will be used to filter data source objects. You can also specify a specific list of owner names that can be used to filter data source objects aside from the default name.

The **Object List** resource limits are as follows:

**Default Owner Name**

This field specifies the default owner name that will be used to filter the data source object list.

**Default Owner Name can be overridden**

This check box allows the user to override the current default owner name.
Allowed Owner Names
This field specifies the list of additional owner names that can be used to filter the data source object list. Owner names should be separated by a comma.

Allow <USERNAME>
This check box allows the current user's owner name to be used to filter the data source object list.

Allow <SQLID>
This check box allows the current user's SQLID to be used to filter the data source object list.
Appendix A: Accessibility

Software accessibility features help those with physical disabilities, such as restricted mobility or limited vision, use their computers.

Standard keyboard equivalents in CorVu Workstation Pro

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. CorVu supports the standard keyboard equivalents that include:

- Shortcut, or accelerator keys, to perform the most frequently used functions in pull-down menus without going to the menu. For example:
  - Ctrl+S to save
  - Ctrl+P to print
  - Ctrl+R to run a query
  
  Shortcuts display in the pull-down menu next to its function.
- Mnemonics, or access keys, are available to perform each function on a menu or window. A mnemonic for a function is the underlined character in the function name. For example:
  - Press F to open the File menu
  - Press O to open the Open window

  Use ALT to activate the mnemonic and move the keyboard focus.

For more information, refer to the documentation for your operating system for a complete list of standard keyboard equivalents.

Standard keyboard navigation

Keyboard navigation refers to using keys instead of a mouse to move from item to item on your screen.

The movement is usually in the order specified by the operating system or your application. CorVu 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.

CorVu applications:

- Inherit their settings from the operating system
- Do not interfere with keyboard accessibility features built into the operating system
- Comply with the IBM Software Accessibility Checklist.

For more information, please go to:
CorVu applications support assistive technology products, such as screen readers and voice synthesizers.

Each CorVu application interface requires special navigation when using a screen reader with query results. See Navigation in CorVu Workstation Pro, on page 399 or Navigation in CorVu Web Pro, on page 401 for more information.

Navigation in CorVu Workstation Pro

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 button clicks</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>To:</td>
<td>Process:</td>
</tr>
<tr>
<td>-------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Left to right switch between <strong>SQL, Diagram, Design, Display, and</strong></td>
<td>Press Alt+Page Up</td>
</tr>
<tr>
<td><strong>Results</strong> views in the active editor</td>
<td></td>
</tr>
<tr>
<td>Right to left switch between <strong>SQL, Diagram, Design, Display, and</strong></td>
<td>Press Alt+Page Down</td>
</tr>
<tr>
<td><strong>Results</strong> views in active editor</td>
<td></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 <strong>HTML</strong> from the **</td>
</tr>
<tr>
<td></td>
<td><strong>Export Type</strong> field. Specify where to save the file in the **File</td>
</tr>
<tr>
<td></td>
<td><strong>name</strong> field.</td>
</tr>
<tr>
<td></td>
<td>6. Press <strong>Finish</strong>.</td>
</tr>
<tr>
<td></td>
<td>7. Open the *.htm file in a browser.</td>
</tr>
<tr>
<td>or</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1. Run the query.</td>
</tr>
<tr>
<td></td>
<td>2. Select Alt+R to display the <strong>Results</strong> 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 <strong>CSV</strong> from the **</td>
</tr>
<tr>
<td></td>
<td><strong>Export Type</strong> field. Specify where to save the file in the **File</td>
</tr>
<tr>
<td></td>
<td><strong>name</strong> field.</td>
</tr>
<tr>
<td></td>
<td>6. Press <strong>Finish</strong>.</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 <strong>Diagram</strong> query editor.</td>
</tr>
<tr>
<td></td>
<td>2. Tab to put focus on the <strong>Add</strong> icon.</td>
</tr>
<tr>
<td></td>
<td>3. Press the spacebar to display the Add window.</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>window.</td>
</tr>
</tbody>
</table>
To: Process:

**Edit query results**
1. Press **CTL+F7** or **CTL+SHIFT+F7** until either the **Workspaces** or **Repository Explorer** view is highlighted.
2. Use the down or up arrow keys to select the folder that contains a table.
3. Use the right arrow key to expand folders and the left arrow key to collapse folders.
4. Use the arrow keys to expose and select the table that you want to edit.
5. Press the **Context Menu** key (the key to the left of the right-side CTRL key).
6. Press the mnemonic key for Open With (the h key).
7. Use the down arrow key to highlight **Table Editor**.
8. Press the ENTER key. The table opens in Table Editor.

**Change font for query results**
1. Run the query.
2. Press Alt+A to select all.
3. Press Alt+R to display the **Results** menu.
4. Press N to display the Font window.

---

### Navigation in CorVu Web Pro

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>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>Move focus to the main menu</td>
<td>Press the left ALT key. Navigate through the menu commands using the arrow keys.</td>
</tr>
<tr>
<td>Open context menus</td>
<td>Press the context menu key if you have an extended keyboard. Press the right Ctrl key. Navigate through the menu commands using the arrow keys.</td>
</tr>
<tr>
<td>Activate top-level menu</td>
<td>Press ALT+Ctrl+1</td>
</tr>
<tr>
<td>Switch between open editors</td>
<td>Press ALT+Ctrl+F6</td>
</tr>
<tr>
<td>Switch between open views</td>
<td>Press ALT+Ctrl+F7</td>
</tr>
</tbody>
</table>
Appendix B: Messages

This topic lists and describes the messages and return codes issued by the program components of CorVu NG Workstation and CorVu NG Web.

CorVu NG Workstation and CorVu NG Workstation messages are numbered as follows:

<table>
<thead>
<tr>
<th>CPTprefix</th>
<th>MESSnumber</th>
</tr>
</thead>
</table>

**CPTprefix**

The abbreviation of the component of the application that threw the error. This can be an indication of the first place to look in order to resolve the error.

**MESSnumber**

The specific number of the error message.

For example: ABC0001

All of the error messages of each component are 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 CorVu NG 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 `retrieveMessagesFromServerOnGetMessage` option in the Advanced JDBC Settings window of the Edit Repository Connection wizard.

---

**API1001:**
**Explanation:** This is a container for other messages.

**User response:** No action is required.

**API1002:**
**The API has not been initialized.**

**Explanation:** The API instance has not been initialized. This could mean that the QMFwin API/QMFwin API/Shuttle API was not initialized, or a repository connection was not established.

**User response:** Check the QMFwin API/QMFwin API/Shuttle API logs to determine the error that caused the API instance to fail, or establish a repository connection.

**API1003:**
**No data found.**

**Explanation:** There was an attempt to obtain a row that does not exist in a result set.

**User response:** No action required.

**API1006:**
**The specified source {0} is not valid.**

**Explanation:** An invalid value was specified for the SourceType parameter of the initializeProc() or initializeQuery() function.

**User response:** Specify a valid value for the SourceType parameter of any initializeProc() or initializeQuery() function.
API1007:
The specified source name is not valid.

Explanation: An invalid value was specified for the Source, CollectionName, PackageName, or ConsistencyToken parameter of an initializeProc(), initializeQuery(), or initializeStaticQuery() function.

User response: Specify a valid value for the Source, CollectionName, PackageName, or ConsistencyToken parameter of any initializeProc(), initializeQuery(), or initializeStaticQuery() function.

API1008:
The specified database server \{0\} is not valid.

Explanation: An unknown data source alias was specified in a getServerAttributes(), getServerType() or initializeServer() function.

User response: Specify a valid data source alias for any getServerAttributes(), getServerType() or initializeServer() function.

API1009:
No database server has been specified.

Explanation: The initializeServer() function was not invoked before other functions were invoked that require an initialized data source.

User response: Ensure that the initializeServer() function is invoked before other data source dependent functions are called.

API1010:
The specified row limit \{0\} is not valid.

Explanation: An invalid value was specified for the RowLimit parameter of an open() function.

User response: Specify a positive value for the RowLimit parameter of any open() function.

API1011:
The specified query ID \{0\} is not valid.

Explanation: An invalid value was specified for the QueryID parameter of an API function.

User response: Specify a valid value for the QueryID parameter of all relevant functions.

API1012:
The SQL verb \{0\} cannot be used with open().

Explanation: An open() function was called for a query that does not contain the SELECT, VALUES, WITH, or CALL SQL verbs.

User response: Ensure that any open() function calls a query that contains one of the SQL verbs that was mentioned above.

API1013:
The SQL verb \{0\} cannot be used with execute().

Explanation: An execute() function was called for a query that contains the SELECT, VALUES, WITH, or CALL SQL verbs.

User response: Ensure that any execute() function does not call a query that contains one of the SQL verbs that was mentioned above.

API1015:
A new procedure could not be created. There may be too many documents already open, or there may be insufficient memory.

Explanation: An unexpected error occurred while creating a procedure using the initializeProc() function.

User response: Check the QMFwin APIQMFwin APIShuttle API logs to find the root error.
Appendix B: Messages

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 APIQMFwin APIShuttle 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 StringDelimeter parameter of an export() function.

User response: Specify a valid value for the StringDelimeter 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.

API1046:
The specified variable ({0}) is not valid.

Explanation: An invalid value was specified for the VariableName parameter of a setVariable() function.

User response: Specify a valid value for the VariableName parameter of any getVariable() function.

API1053:
The SQL verb ({0}) cannot be used with ExecuteStoredProcedure().

Explanation: A query that does not contain the CALL SQL verb was specified for an executeStoredProcedureEx() function.

User response: Specify a CALL query for any executeStoredProcedureEx() function.
API1054:  
Can't prepare a static section.  
Explanation: A static query was specified for a prepare() function; only dynamic queries can be specified.  
User response: Specify a dynamic query for any prepare() function.

API10004:  
The values specified for ForceDialog and SuppressDialog are mutually exclusive.  
Explanation: The initializeServer() function was called and the values that were specified for ForceDialog and SuppressDialog parameters were mutually exclusive.  
User response: Check the code that invokes the initializeServer() function.

API10005:  
The specified value is not valid.  
Explanation: The value that was specified for the parameter of a function is incorrect.  
User response: Check to ensure that all of the specified parameter values for all of the functions that were called in the API instance are valid.

BIC00000:  
Internal Error  
Explanation: There was an internal error.  
User response: Collect support data by selecting Help → Collect Support Data and forward it to customer support.

BIC00001:  
Illegal Argument  
Explanation: The specified argument is not valid.  
User response: Collect support data by selecting Help → Collect Support Data and forward it to customer support.

BIC00002:  
Null Pointer  
Explanation: A null pointer exception occurred.  
User response: Collect support data by selecting Help → Collect Support Data and forward it to customer support.

BIC00004:  
{0}  
Explanation: The specified exception occurred.  
User response: Correct the specified exception and try again.

BIC00005:  
Unknown exception.  
Explanation: An error occurred while retrieving a resource string.  
User response: Collect support data by selecting Help → Collect Support Data and forward it to customer support.

BIC00006:  
This feature is not supported by CorVu NG  
Explanation: The specified feature is not supported by CorVu NG  
User response: No action is required.

BIC00007:  
Object is not supported by CorVu NG  
Explanation: The specified object is not supported by CorVu NG
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:**
Evaluating expression failed: {0}

**Explanation:** The evaluation of the indicated expression failed.

**User response:** Specify a valid expression for evaluation.

---

**CLC00014:**
Evaluating expression has failed.

**Explanation:** The evaluation of the specified expression has failed.

**User response:** Check the syntax of the expression for errors and try again.

---

**CLC00016:**
Cannot parse an expression: {0}

**Explanation:** An internal error occurred.

**User response:** Collect support data by selecting Help → Collect Support Data and forward it to customer support.

---

**CLC00018:**
Unknown function name: {0}.

**Explanation:** The indicated function name is not valid

**User response:** Specify a valid function name.

---

**CLC00030:**
Unable to process URL: "{0}".

**Explanation:** The indicated URL is invalid.

**User response:** Specify a valid URL.

---

**CLC00034:**
"{0}" function description is missing.

**Explanation:** An internal error occurred.

**User response:** Collect support data by selecting Help → Collect Support Data and forward it to customer support.

---

**CLC00038:**
Multiple declarations of "{0}" library in LibraryFactory class.

**Explanation:** An internal error occurred.

**User response:** Collect support data by selecting Help → Collect Support Data and forward it to customer support.

---

**CLC00039:**
Multiple declarations of "{0}" library in LibraryFactory class.

**Explanation:** An internal error occurred.

**User response:** Collect support data by selecting Help → Collect Support Data and forward it to customer support.

---

**CLC00040:**
Cannot load "{0}" library in LibraryFactory class.

**Explanation:** An internal error occurred.

**User response:** Collect support data by selecting Help → Collect Support Data and forward it to customer support.

---

**CLC00041:**
Types "{0}" and "{1}" are not acceptable for "{2}" function.
**Explanation:** The arguments that were passed to the function are of invalid types.

**User response:** Check the syntax of the expression for errors and try again.

---

**CLC00042:**
Type "{0}" is not acceptable for "{1}" function.

**Explanation:** The specified function does not accept parameters of the given type.

**User response:** Check the syntax of the expression for errors and try again.

---

**CLC00043:**
Information service is undefined.

**Explanation:** An internal error occurred.

**User response:** Collect support data by selecting Help → Collect Support Data and forward it to customer support.

---

**CLC00044:**
Aggregation service is undefined.

**Explanation:** An internal error occurred.

**User response:** Collect support data by selecting Help → Collect Support Data and forward it to customer support.

---

**CLC00045:**
Hierarchy service is undefined.

**Explanation:** An internal error occurred.

**User response:** Collect support data by selecting Help → Collect Support Data and forward it to customer support.

---

**CLC00046:**
Color service is undefined.

**Explanation:** An internal error occurred.

**User response:** Collect support data by selecting Help → Collect Support Data and forward it to customer support.

---

**CLC00047:**
PPI service is undefined.

**Explanation:** An internal error occurred.

**User response:** Collect support data by selecting Help → Collect Support Data and forward it to customer support.

---

**CLC00048:**
Value provider is undefined.

**Explanation:** An internal error occurred.

**User response:** Collect support data by selecting Help → Collect Support Data and forward it to customer support.

---

**CLC00049:**
Function "{0}" is defined for {1} argument(s).

**Explanation:** An invalid number of parameters were specified for the indicated function.

**User response:** Specify a correct number of parameters for the indicated function.

---

**CLC00050:**
Function "{0}" is undefined for argument {1}.

**Explanation:** An invalid number of parameters were specified for the indicated function.

**User response:** Specify a valid number of parameters for the indicated function.

---

**CLC00051:**
One of the operands does not have measurement units.
Appendix B: Messages

Explanation: One of the operands of the function does not have measurement units.
User response: Make sure that all of the operands in the function have measurement units.

CLC00052:
Cannot multiply two measured values.
Explanation: Measured values cannot be multiplied.
User response: Specify values that are not measured to be multiplied.

CLC00053:
Cannot create list from elements of "[0]" type.
Explanation: A list cannot be created using elements of the indicated type.
User response: Specify elements of a different type to be listed.

CLC00054:
List cannot contain elements of different types.
Explanation: The specified list cannot contain elements of different types.
User response: Specify elements of the same type to be listed.

CLC00055:
Types "[0]", "[1]" and "[2]" are not acceptable for "[3]" function.
Explanation: The indicated function is not compatible with the given types.
User response: Check the syntax of the expression for errors and try again.

CLC00056:
Argument(s) with measurement units are not acceptable for "[0]" function.
Explanation: The indicated function cannot contain arguments with measurement units.
User response: Remove all measurement units from the arguments of the function.

CLC00057:
Incorrect index given.
Explanation: An error occurred in the LineText function.
User response: Check the arguments that are specified for the LineText function for errors.

CLC00058:
Types "[0]", "[1]", "[2]" and "[3]" are not acceptable for "[4]" function.
Explanation: The indicated function is not compatible with the given types.
User response: Check the expression syntax for errors and try again.

CLC00059:
Calculator exception cause: [0]
Explanation: A generic error occurred.
User response: Check the syntax of the expression for errors and try again.

CLC00060:
Value for variable "[0]" is not defined.
Explanation: The value for the indicated variable is not defined.
User response: Define the value for the indicated variable.

CLC00061:
Function "[0]" caused error: [1]
Explanation: The indicated function caused the indicated error.
User response: Correct the indicated error and rerun the function.

CLC00062:
Property "{0}" contains error: {1}
Explanation: The indicated property caused the indicated error.
User response: Correct the indicated error and rerun the function.

CLC00063:
List sizes are not comparable.
Explanation: Invalid parameters with type List were specified for add, multiply, divide, or subtract functions.
User response: Specify a valid value for the List parameter of any add, multiply, divide, or subtract functions.

CLC00066:
Cannot be evaluated for empty list.
Explanation: No value was specified for the List parameter of a function.
User response: Specify a valid value for the List parameter of any relevant function.

CLC00067:
Function @if has odd number of parameters: @if(cond1, exp1, cond2, exp2, ..., expn). expn is used when there aren't true conditions
Explanation: The syntax of an @if function is invalid.
User response: Check the syntax of any @if functions for errors and try again.

DAL01001:
The feature is not supported.
Explanation: The specified feature is not supported by the database.
User response: Collect support data by selecting Help → Collect Support Data and forward it to customer support.

DAL01002:
No column named "{0}" was found.
Explanation: The indicated column was not found.
User response: Check the column name for errors or specify a different column name.

DAL01004:
The current row is unsuitable for get/set operations.
Explanation: The specified row can not be retrieved or set.
User response: Collect support data by selecting Help → Collect Support Data and forward it to customer support.

DAL01005:
The database cursor has been closed.
Explanation: The requested operation could not be completed because the cursor is closed.
User response: Collect support data by selecting Help → Collect Support Data and forward it to customer support.

DAL01006:
An error occurred while accessing the file. {0}
Explanation: An input/output error occurred while accessing the file.
User response: Collect support data by selecting Help → Collect Support Data and forward it to customer support.
### DAL01007:
**Character encoding** `{0}` **is not supported.**

**Explanation:** The indicated character encoding is not supported.

**User response:** Use a character set that is supported by your current JVM.

### DAL01008:
**An error occurred while accessing the database.** `{0}`

**Explanation:** An error occurred while accessing the indicated database.

**User response:** Collect support data by selecting Help → Collect Support Data and forward it to customer support.

### DAL01009:
**An internal error occurred.** `{0}`

**Explanation:** The indicated internal error occurred.

**User response:** Collect support data by selecting Help → Collect Support Data and forward it to customer support.

### DAL01011:
**Class** `{0}` **is not suitable for server description**

**Explanation:** An internal error occurred.

**User response:** Collect support data by selecting Help → Collect Support Data and forward it to customer support.

### DAL01012:
**Requested operation is not available in offline mode**

**Explanation:** The specified operation is not available in offline mode.

**User response:** Specify a different operation to apply to the active object, or switch to online mode and try again.

### DAL01013:
**Cannot create connection to server ''{0}'' due to the following:** `{1}`

**Explanation:** A connection to the indicated server could not be established due to the indicated error.

**User response:** Resolve the indicated error and retry to connect to the server.

### DAL01014:
**Invalid set of connection parameters used,**

**Explanation:** An internal error occurred.

**User response:** Collect support data by selecting Help → Collect Support Data and forward it to customer support.

### DAL01016:
**Cannot parse data (base64 encoding rules violation)**

**Explanation:** The XML file is corrupted.

**User response:** Fix any errors in the XML file and try again.

### DAL01017:
**Duplicate query with name ''{0}'' detected in package ''{1}''**

**Explanation:** An internal error occurred; the indicated package contains several queries with the same specified name.

**User response:** Collect support data by selecting Help → Collect Support Data and forward it to customer support.

### DAL01018:
**The source data is not compatible with the target table.**

**Explanation:** The attempt to save data in the database table failed because of the following:

- Number of columns does not match.
• Columns have incompatible types.

**User response:** Select compatible target table and try again.

**DAL01019:**
Operation is not supported.

**Explanation:** An internal error occurred; the requested operation can not be performed due to internal limitations.

**User response:** Collect support data by selecting Help → Collect Support Data and forward it to customer support.

**DAL01020:**
Invalid SQL type definition: "{0}"  

**Explanation:** An internal error occurred; the indicated SQL type definition is invalid (it can be mapped to more than one known data types).

**User response:** Collect support data by selecting Help → Collect Support Data and forward it to customer support.

**DAL01021:**
This product does not contain SQLj support module

**Explanation:** CorVu NG does not support or can not initialize the SQLj support module.

**User response:** Collect support data by selecting Help → Collect Support Data and forward it to customer support.

**DAL01022:**
Cannot have two different driver libraries with the same driver name.

**Explanation:** The driver with the specified driver name is already defined in another library definition.

**User response:** Use (edit) the already defined driver library instead of creating a new one.

**DAL01023:**
An error has occurred while saving driver libraries list: {0}

**Explanation:** The indicated error occurred while the driver libraries list was being saved.

**User response:** Resolve the indicated error and try again.

**DAL01025:**
An error has occurred while loading the driver libraries list: {0}

**Explanation:** The indicated error occurred while trying to load the driver libraries list.

**User response:** Resolve the indicated error and try again.

**DAL01026:**
Operation cannot be performed due to resource limits restriction: {0}

**Explanation:** The operation could not be performed due to the indicated resource limits restriction.

**User response:** Check your current resource limits by selecting Resource Limits from the View menu. Specify an operation within your resource limits or contact your administrator for specific permissions.

**DAL01027:**
User name is missing.

**Explanation:** The user name is missing from the User Information dialog name field.

**User response:** Specify a user name.

**DAL01028:**
Password is missing.

**Explanation:** The password is missing from the User Information dialog.

**User response:** Specify a password.
<table>
<thead>
<tr>
<th>DAL01029:</th>
<th>Wrong JDBC URL: {0}.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>The indicated JDBC URL cannot be processed by CorVu NG. The URL may be incorrect.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Check the JDBC URL for errors or specify a different URL.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DAL01030:</th>
<th>Query parameter {0} is of wrong Java type. Data type &quot;{1}&quot; is required.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>An internal error occurred while running the query. The query parameters have invalid types.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Collect support data by selecting Help → Collect Support Data and forward it to customer support.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DAL01031:</th>
<th>Cached server descriptor cannot be restored due to the following: {0}</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>The cached server descriptor cannot be restored due to the following error.</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>DAL01032:</th>
<th>An error occurred while making remote method call. {0}</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>The indicated error occurred while the operation was running on the server side.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Refer to the indicated error description for possible response(s).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DAL01033:</th>
<th>Client request cannot be processed by remote server.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>An internal error occurred; the client request cannot be processed by a remote server.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Collect support data by selecting Help → Collect Support Data and forward it to customer support.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DAL01034:</th>
<th>Attempt to use incomplete metadata descriptor is detected.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>An internal error occurred.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Collect support data by selecting Help → Collect Support Data and forward it to customer support.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DAL01035:</th>
<th>Editing is not supported for this data set.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>An internal error occurred; this data set cannot be edited.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Collect support data by selecting Help → Collect Support Data and forward it to customer support.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DAL01036:</th>
<th>The current row is unsuitable for editing.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>An internal error occurred; the current row can not be edited.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Collect support data by selecting Help → Collect Support Data and forward it to customer support.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DAL01037:</th>
<th>The database server does not support table renaming</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>Tables in this database server can not be renamed by CorVu NG.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>No action is required.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DAL01038:</th>
<th>The edit operation was cancelled.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>The table editing operation was cancelled by the user.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>No action is required.</td>
</tr>
</tbody>
</table>
**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 cannot be modified or restored.

**User response:** Collect support data by selecting Help → Collect Support Data and forward it to customer support.

**DAL01044:**
Connection is blocked due to the following: {0}

**Explanation:** The attempted connection was blocked due to the indicated error or restriction.

**User response:** Depending on the indicated problem, either resolve the specified error or contact your administrator to remove the restriction.

**DAL01046:**
This product does not contain static mode support module for selected database.

**Explanation:** This product does not contain static mode support module for selected database.

**User response:** Collect support data by selecting Help → Collect Support Data and forward it to customer support.

**DAL01048:**
Driver class name "{0}" is not loaded. This may be caused by an incorrect driver class name or missing JDBC driver libraries.

**Explanation:** The indicated driver class name cannot be loaded. This may be caused by an incorrect driver class name or missing JDBC driver libraries.

**User response:** Check that there is a proper definition for the JDBC driver library that refers to the indicated driver class.

**DAL01049:**
Selected protection mode is not supported by database.

**Explanation:** The selected protection mode is not supported by the specified database.
User response: Collect support data by selecting Help → Collect Support Data and forward it to customer support.

DAL01050:
Driver specific functionality is not available due to: {0}
Explanation: Driver specific functionality is not available due to the indicated error.
User response: Resolve the indicated error and retry. Verify that the JDBC driver library for the driver is properly configured and that it contains all of the necessary JAR files. If the problem persists, collect support data by selecting Help → Collect Support Data and forward it to customer support.

DAL01200:
Generic data access error has occurred. {0}
Explanation: An error with no textual description occurred. Additional information may follow the message. This error code may refer to different error conditions depending on product versions.
User response: Collect support data by selecting Help → Collect Support Data and forward it to customer support.

DAL01201:
Generic data access error has occurred. {0}, {1}
Explanation: An error with no textual description occurred. Additional information may follow the message. This error code may refer to different error conditions depending on product versions.
User response: Collect support data by selecting Help → Collect Support Data and forward it to customer support.

DAL01202:
Generic data access error has occurred. {0}, {1}, {2}
Explanation: An error with no textual description occurred. Additional information may follow the message. This error code may refer to different error conditions depending on product versions.
User response: Collect support data by selecting Help → Collect Support Data and forward it to customer support.

DAL01203:
Generic data access error has occurred. {0}, {1}, {2}, {3}
Explanation: An error with no textual description occurred. Additional information may follow the message. This error code may refer to different error conditions depending on product versions.
User response: Collect support data by selecting Help → Collect Support Data and forward it to customer support.

DAL01204:
Generic data access error has occurred. {0}, {1}, {2}, {3}, {4}
Explanation: An error with no textual description occurred. Additional information may follow the message. This error code may refer to different error conditions depending on product versions.
User response: Collect support data by selecting Help → Collect Support Data and forward it to customer support.

DAL01205:
Generic data access error has occurred. {0}, {1}, {2}, {3}, {4}, {5}
Explanation: An error with no textual description occurred. Additional information may follow the message. This error code may refer to different error conditions depending on product versions.
User response: Collect support data by selecting Help → Collect Support Data and forward it to customer support.

DAL01206:
Generic data access error has occurred. {0}, {1}, {2}, {3}, {4}, {5}, {6}
Explanation: An error with no textual description occurred. Additional information may follow the message. This error code may refer to different error conditions depending on product versions.
User response: Collect support data by selecting Help → Collect Support Data and forward it to customer support.
DAL01207:
Generic data access error has occurred. {0}, {1}, {2}, {3}, {4}, {5}, {6}, {7}

Explanation: An error with no textual description occurred. Additional information may follow the message. This error code may refer to different error conditions depending on product versions.

User response: Collect support data by selecting Help → Collect Support Data and forward it to customer support.

DAL01208:
Generic data access error has occurred. {0}, {1}, {2}, {3}, {4}, {5}, {6}, {7}, {8}

Explanation: An error with no textual description occurred. Additional information may follow the message. This error code may refer to different error conditions depending on product versions.

User response: Collect support data by selecting Help → Collect Support Data and forward it to customer support.

DDM0001:
Unexpected error occurred.

Explanation: An internal error in the driver code was detected.

User response: Send an error report to the developers.

DDM0002:
Object {0} have no fields with codepoint {1}

Explanation: An internal error in the driver code was detected.

User response: Send an error report to the developers.

DDM0003:
Socket open exception. {0}

Explanation: Unable to establish TCP/IP connection.

User response: Check the host and port details in the JDBC URL; ping the server to check that the network connection works.

DDM0004:
Socket write exception. {0}

Explanation: There was a network connection failure or an internal error in the driver code was detected.

User response: Check you network connection. If the connection works fine, send an error report to the developers.

DDM0005:
Data format exception detected.

Explanation: An internal error in the driver code was detected.

User response: Send an error report to the developers.

DDM0006:
Invalid JDBC URL {0}

Explanation: An internal error in the driver code was detected.

User response: Send an error report to the developers.

DDM0007:
DSS header format error detected.

Explanation: An internal error in the driver code was detected.

User response: Send an error report to the developers.

DDM0008:
Unsupported CCSID {0}
Appendix B: Messages

**Explanation:** An internal error was detected in the driver code.

**User response:** Send an error report to the developers.

**DDM0009:**
Cipher error {0}

**Explanation:** An internal error in the driver code was detected.

**User response:** Send an error report to the developers.

**DDM0010:**
Object {0} does not contain required field {1}

**Explanation:** An internal error in the driver code was detected.

**User response:** Send an error report to the developers.

**DDM0011:**
Object {0} does not support value {1}

**Explanation:** An internal error was detected in the driver code.

**User response:** Send an error report to the developers.

**DDM0012:**
Syntax error detected: {0}

**Explanation:** An internal error in the driver code was detected.

**User response:** Send an error report to the developers.

**DDM0013:**
Syntax error detected: {0}

**Explanation:** An internal error in the driver code was detected.

**User response:** Send an error report to the developers.

**DDM0014:**
Object {0} can hold no more repeated fields {1}

**Explanation:** An internal error in the driver code was detected.

**User response:** Send an error report to the developers.

**DDM0015:**
Illegal port number value: {0}

**Explanation:** The JDBC URL contains an invalid value in the Port field.

**User response:** Specify a correct port number in the JDBC URL.

**DDM0016:**
{0} is invalid length for the field {1} of object {2}

**Explanation:** An internal error was detected in the driver code.

**User response:** Send an error report to the developers.

**DDM0017:**
Unknown FD:OCA descriptor type id {0}

**Explanation:** An internal error was detected in the driver code.

**User response:** Send an error report to the developers.

**DDM0018:**
Null value for non nullable data detected.
Explanation: An internal error was detected in the driver code.
User response: Send an error report to the developers.

DDM0019:
Whole number of FD:OCA {0} triplet repeating groups cannot be placed in {1} bytes
Explanation: An internal error was detected in the driver code.
User response: Send an error report to the developers.

DDM0020:
Illegal CPT triplet detected.
Explanation: An internal error was detected in the driver code.
User response: Send an error report to the developers.

DDM0021:
Illegal duplicated field {1} in object {0}
Explanation: An internal error was detected in the driver code.
User response: Send an error report to the developers.

DDM0022:
Referenced FD:OCA triplet {0} not found
Explanation: An internal error was detected in the driver code.
User response: Send an error report to the developers.

DDM0023:
FD:OCA {0} triplets cannot be referenced from RLO triplet
Explanation: An internal error was detected in the driver code.
User response: Send an error report to the developers.

DDM0024:
Replication factor {0} is detected in RLO triplet while only 1 and 0 is allowed by DRDA
Explanation: An internal error was detected in the driver code.
User response: Send an error report to the developers.

DDM0025:
FD:OCA {0} triplets cannot be referenced from GDA triplet
Explanation: An internal error was detected in the driver code.
User response: Send an error report to the developers.

DDM0026:
Unknown DRDA type {0}
Explanation: An internal error was detected in the driver code.
User response: Send an error report to the developers.

DDM0027:
Infinite element count is not allowed for RLO handler of {0}
Explanation: An internal error was detected in the driver code.
User response: Send an error report to the developers.

DDM0028:
An error occurred while parsing triplet handler for DRDA type {0}
Appendix B: Messages

**Explanation:** An internal error was detected in the driver code.

**User response:** Send an error report to the developers.

**DDM0029:**
Unspecified value for JDBC statement variable {0}

*Explanation:* An internal error was detected in the driver code.

*User response:* Send an error report to the developers.

**DDM0030:**
Illegal type conversion.

*Explanation:* An internal error was detected in the driver code.

*User response:* Send an error report to the developers.

**DDM0031:**
Illegal date or time format detected: {0}

*Explanation:* An internal error was detected in the driver code.

*User response:* Send an error report to the developers.

**DDM0032:**
Prepared statement parameter type {0} (from java.sql.Type set) is not supported.

*Explanation:* An internal error was detected in the driver code.

*User response:* Send an error report to the developers.

**DDM0033:**
Unknown DB2 type {0}

*Explanation:* An internal error was detected in the driver code.

*User response:* Send an error report to the developers.

**DDM0035:**
Statement is closed.

*Explanation:* An attempt to work with a closed Statement object was detected.

*User response:* Fix your program code.

**DDM0036:**
ResultSet is closed.

*Explanation:* An attempt to work with a closed ResultSet object was detected.

*User response:* Fix your program code.

**DDM0037:**
Object is closed.

*Explanation:* An attempt to work with a closed object was detected.

*User response:* Fix your program code.

**DDM0038:**
Operation cannot be performed in current ResultSet state.

*Explanation:* An attempt to work with a ResultSet column of a result set that is not positioned on any valid record was detected.

*User response:* Fix your program code.
DDM0039:
Error reply is received.

Explanation: An internal error in the driver code was detected.

User response: Send an error report to the developers.

DDM0040:
Error reply is received for \{0\} command.

Explanation: An internal error in the driver code was detected.

User response: Send an error report to the developers.

DDM0041:
Error detected while processing \{0\} object.

Explanation: An internal error in the driver code was detected.

User response: Send an error report to the developers.

DRL00001:
The maximum number of connections to the server has been reached.

Explanation: The maximum number of connections to the server has been reached.

User response: Increase the maximum of number of connections parameter in the Repository Storage dialog or set the value to zero to remove this restriction.

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: GDAlength override exceeds limits.
Explanation: An internal error in the driver code was detected.
User response: Send an error report to the developers.

DSCERRCD0052: GDAprecision 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.

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.
**Appendix B: Messages**

**User response:** Send an error report to the developers.

**ERM0008:**
**Parameter Not Supported.**

**Explanation:** An internal error in the driver code was detected.

**User response:** Send an error report to the developers.

**ERM0009:**
**Resource Limits Reached.**

**Explanation:** An internal error in the driver code was detected.

**User response:** Send an error report to the developers.

**ERM0010:**
**Data Stream Syntax Error.**

**Explanation:** An internal error in the driver code was detected.

**User response:** Send an error report to the developers.

**ERM0011:**
**Target Not Supported.**

**Explanation:** An internal error in the driver code was detected.

**User response:** Send an error report to the developers.

**ERM0012:**
**Parameter Value Not Supported.**

**Explanation:** An internal error in the driver code was detected.

**User response:** Send an error report to the developers.

**ERM0013:**
**Conversational Protocol Error.**

**Explanation:** An internal error in the driver code was detected.

**User response:** Send an error report to the developers.

**ERM0014:**
**RDB Currently Accessed.**

**Explanation:** An internal error in the driver code was detected.

**User response:** Send an error report to the developers.

**ERM0015:**
**RDB Access Failed Reply Message.**

**Explanation:** An internal error in the driver code was detected.

**User response:** Send an error report to the developers.

**ERM0016:**
**Not Authorized To RDB.**

**Explanation:** An internal error in the driver code was detected.

**User response:** Send an error report to the developers.

**ERM0017:**
**RDB Not Found.**

**Explanation:** An internal error in the driver code was detected.
User response: Send an error report to the developers.

ERM0018: Resource Limits Reached.
Explanation: An internal error in the driver code was detected.
User response: Send an error report to the developers.

ERM0019: Data Stream Syntax Error.
Explanation: An internal error in the driver code was detected.
User response: Send an error report to the developers.

ERM0020: Invalid Request.
Explanation: An internal error in the driver code was detected.
User response: Send an error report to the developers.

ERM0021: Abnormal End Unit of Work Condition.
Explanation: An internal error in the driver code was detected.
User response: Send an error report to the developers.

ERM0022: RDB Package Binding Process Active.
Explanation: An internal error in the driver code was detected.
User response: Send an error report to the developers.

ERM0023: Query Not Open.
Explanation: An internal error in the driver code was detected.
User response: Send an error report to the developers.

ERM0024: RDB Not Accessed.
Explanation: An internal error in the driver code was detected.
User response: Send an error report to the developers.

ERM0025: Data Descriptor Mismatch.
Explanation: An internal error in the driver code was detected.
User response: Send an error report to the developers.

ERM0026: Object Not Supported.
Explanation: An internal error in the driver code was detected.
User response: Send an error report to the developers.

ERM0027: RDB Package Binding Not Active.
Explanation: An internal error in the driver code was detected.
User response: Send an error report to the developers.

ERM0028:  
RDB Update Reply Message.  
Explanation: An internal error in the driver code was detected.  
User response: Send an error report to the developers.

ERM0029:  
SQL Error Condition.  
Explanation: An internal error in the driver code was detected.  
User response: Send an error report to the developers.

ERM0030:  
End of Data.  
Explanation: An internal error in the driver code was detected.  
User response: Send an error report to the developers.

ERM0031:  
End of Query.  
Explanation: An internal error in the driver code was detected.  
User response: Send an error report to the developers.

ERM0032:  
Commitment Request.  
Explanation: An internal error in the driver code was detected.  
User response: Send an error report to the developers.

ERM0033:  
End Unit of Work Condition.  
Explanation: An internal error in the driver code was detected.  
User response: Send an error report to the developers.

ERM0034:  
Invalid Descriptor.  
Explanation: An internal error in the driver code was detected.  
User response: Send an error report to the developers.

ERM0035:  
Open Query Complete.  
Explanation: An internal error in the driver code was detected.  
User response: Send an error report to the developers.

ERM0036:  
Open Query Failure.  
Explanation: An internal error in the driver code was detected.  
User response: Send an error report to the developers.

ERM0037:  
Query Previously Opened.  
Explanation: An internal error in the driver code was detected.
User response: Send an error report to the developers.

ERM0038: Command Violation.
Explanation: An internal error in the driver code was detected.
User response: Send an error report to the developers.

ERM1200: Generic DRDA error has occurred. {0}
Explanation: The indicated generic DRDA error has occurred.
User response: Resolve the indicated generic DRDA error and try again.

ERM1201: Generic DRDA error has occurred. {0}, {1}
Explanation: The two indicated generic DRDA error have occurred.
User response: Resolve the two indicated errors and try again.

ERM1202: Generic DRDA error has occurred. {0}, {1}, {2}
Explanation: The three indicated generic DRDA errors have occurred.
User response: Resolve the three indicated errors and try again.

ERM1203: Generic DRDA error has occurred. {0}, {1}, {2}, {3}
Explanation: The four indicated generic DRDA errors have occurred.
User response: Resolve the four indicated errors and try again.

ERM1204: Generic DRDA error has occurred. {0}, {1}, {2}, {3}, {4}
Explanation: The five indicated generic DRDA errors have occurred.
User response: Resolve the five indicated errors and try again.

ERM1205: Generic DRDA error has occurred. {0}, {1}, {2}, {3}, {4}, {5}
Explanation: The six indicated generic DRDA errors have occurred.
User response: Resolve the six indicated errors and try again.

ERM1206: Generic DRDA error has occurred. {0}, {1}, {2}, {3}, {4}, {5}, {6}
Explanation: The seven indicated generic DRDA errors have occurred.
User response: Resolve the seven indicated errors and try again.

ERM1207: Generic DRDA error has occurred. {0}, {1}, {2}, {3}, {4}, {5}, {6}, {7}
Explanation: The eight 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.
Appendix B: Messages

User response: Resolve the nine indicated errors and try again.

EXPT1000:
Cannot perform export command: {0}
Explanation: The indicated export command can not be performed.
User response: Resolve the first indicated error and try again.

EXPT1001:
An unsupported data type {0} was encountered.
Explanation: The result set contains columns with the types that are not supported by the IXF370 format.
User response: Make sure that the result set does not contain columns with any of the following types: BINARY, VARBINARY, ROWID, LONGVARBINARY, CLOB, DBCLOB, BLOB, GRAPHIC, VARGRAPHIC, or LONGVARGRAPHIC.

EXPT1002:
The length of column {0} must be less than {1}.
Explanation: The length of the indicated column must be less than the indicated length due to restrictions of the IXF format.
User response: Make sure that the indicated data column is less than the indicated length. Try to export using another format (for example, XML).

EXPT1003:
The precision of column {0} must be an odd number between {1} and {2}, inclusive.
Explanation: The precision of the indicated column must be an odd number between the two indicated values due to restrictions of the IXF format.
User response: Make sure the precision of the indicated data column is an odd number between the two indicated values. Try to export using another format (for example, XML).

EXPT1004:
The data could not be exported due to an unsupported CCSID {0}.
Explanation: The indicated CCSID is not supported for export.
User response: Specify a valid CCSID for export. A list of supported CCSIDs can be obtained from the CCSID drop down menu in the Export dialog.

EXPT1005:
The data could not be exported because the single-byte code page is not specified.
Explanation: A result set containing CLOB, CHAR, VARCHAR, or LONGVARCHAR columns was exported in IXF format but a CCSID that contains a single-byte code page was not specified.
User response: Select a CCSID that contains a single-byte code page.

EXPT1006:
The data could not be exported because the double-byte code page is not specified.
Explanation: A result set containing DBCLOB, GRAPHIC, VARGRAPHIC, or LONGVARGRAPHIC columns was exported in IXF format but a CCSID that contains a double-byte code page was not specified.
User response: Select a CCSID that contains a double-byte code page.

EXPT1007:
The value in column {0} exceeds the maximum length supported by the dBASE file format.
Explanation: The value in the indicated column exceeds the character limit that is supported by the dBASE file format.
User response: Shorten the value in the indicated column or specify a different file format for export.
EXPT1008:  
The data type of column \{0\} is not supported in dBASE files.  
Explanation: The data type of the indicated column is not supported by the dBASE file format.  
User response: Specify a different data type for the indicated column or select a different file format for export. Supported types: CHAR, VARCHAR, SMALLINT, INTEGER, DECIMAL, BIGINT, DOUBLE, FLOAT, DATE, and TIME.

EXPT1009:  
The row length in a dBASE file cannot exceed 4000 characters.  
Explanation: The summary length of all of the result set's columns exceeds 4,000 characters, which is not supported by the dBASE file format.  
User response: Make sure that the length is shorter than 4,000 characters, or select a different file format for export.

EXPT1010:  
The specified data source does not exist.  
Explanation: There was an attempt to export a result set to a data source that does not exist  
User response: Specify a valid data source.

EXPT1011:  
The operation cancelled.  
Explanation: The user cancelled the export command when prompted for a user name and password.  
User response: Specify a valid user name and password for the export command.

EXPT1012:  
Name must be specified.  
Explanation: There was an attempt to export a result set to a table that does not exist.  
User response: Specify a valid export table.

EXPT1013:  
Invalid storage entry.  
Explanation: The storage file contains corrupted data.  
User response: Set up 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 CorVu NG that was used to create the form.
Appendix B: Messages

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 CorVu NG 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 CorVu NG 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 CorVu NG 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 CorVu NG 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 CorVu NG 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 CorVu NG 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.
<table>
<thead>
<tr>
<th>FRM1011:</th>
<th>There is no condition expression with ID {0}.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Specify YES, NO, or Ci where Ci is the existing condition.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FRM1012:</th>
<th>Line Wrapping Width must be NONE when Number of Fixed Columns is not NONE.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Specify NONE in either the Line Wrapping Width field or the Number of Fixed Columns field.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FRM1013:</th>
<th>The ID for calculation {0} is already being used by a different calculation.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>In the Calculations page of the form designer, two or more calculations with the same ID were specified.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Ensure that each calculation ID is unique.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FRM1014:</th>
<th>The ID for condition {0} is already being used by a different condition.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>In the Conditions page of the form designer, two or more conditions with the same ID were specified.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Ensure that each of the conditions are unique.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FRM1015:</th>
<th>Line Wrapping Width must be 0 if any column uses a column wrapping edit code.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Either set the Line Wrapping Width field to NONE, or do not use the line wrapping edit code.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FRM1016:</th>
<th>Automatically Reorder Report Columns must be TRUE when using the ACROSS usage code.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Either set the Automatically Reorder Report Columns field to TRUE or do not use the ACROSS usage code.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FRM1017:</th>
<th>The ACROSS usage code requires other columns to use the GROUP and aggregate usage codes.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>If any columns use the ACROSS usage code, then all other columns must use either the OMIT, GROUP, or AGGREGATION usage code.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Set the OMIT, GROUP, or AGGREGATION usage codes for all other columns.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FRM1018:</th>
<th>Only one column can use the ACROSS usage code.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>Only one column can use the ACROSS usage code.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Ensure that only one column uses the ACROSS usage code.</td>
</tr>
</tbody>
</table>

| FRM1019: | The GROUP usage code requires other columns to use aggregate usage codes. |
Appendix B: Messages

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 CorVu NG that was used to create the form or try to change the locale and reopen.

FRM1026:
The variable "{0}" is invalid in this context.

Explanation: The specified variable name is invalid when used in the expression.

User response: Specify a valid variable name for the expression.

FRM1027:
There is no column {0}.

Explanation: The expression contains a variable that refers to a column that does not exist.

User response: Specify a valid column number in the expression variable.

FRM1028:
The suffix '_B' is invalid in this context.
Explanation: The suffix '_B' can not be used in predefined variables, and as such is invalid for the current variable.
User response: Remove the suffix '_B' from all predefined variables.

FRM1029:
Unable to open linked data source.
Explanation: The form attempted to open a linked data source and an error occurred during this process.
User response: Specify a valid data source object for the form.

FRM1030:
An error occurred while registering user edit code.
User response: Review the FormCustomEditCode implementation documentation and check for errors.

FRM1031:
The user-defined edit routine class could not be found. {0}
Explanation: The user implementation class FormCustomEditCode can not be loaded by the Classloader.
User response: Check the className.jar CLASSPATH.

FRM1032:
An error occurred while initializing user-defined edit routine class. {0}
Explanation: An error occurred.
User response: See additional information in the wrapped exception.

GERR00001:
Generic driver error has occurred. {0}
Explanation: An internal error in the driver code was detected.
User response: Send an error report to the developers.

GERR00002:
Generic driver error has occurred. {0}, {1}, {2}
Explanation: An internal error in the driver code was detected.
User response: Send an error report to the developers.

GERR00003:
Generic driver error has occurred. {0}, {1}, {2}, {3}
Explanation: An internal error in the driver code was detected.
User response: Send an error report to the developers.

GERR00004:
Generic driver error has occurred. {0}, {1}, {2}, {3}, {4}
Explanation: An internal error in the driver code was detected.
User response: Send an error report to the developers.

GERR00005:
Generic driver error has occurred. {0}, {1}, {2}, {3}, {4}, {5}
Explanation: An internal error in the driver code was detected.
User response: Send an error report to the developers.

GERR00006:
Generic driver error has occurred. {0}, {1}, {2}, {3}, {4}, {5}, {6}
Appendix B: Messages

Explanation: An internal error in the driver code was detected.
User response: Send an error report to the developers.

GERR00007:
Generic driver error has occurred. {0}, {1}, {2}, {3}, {4}, {5}, {6}, {7}
Explanation: An internal error in the driver code was detected.
User response: Send an error report to the developers.

GERR00008:
Generic driver error has occurred. {0}, {1}, {2}, {3}, {4}, {5}, {6}, {7}, {8}
Explanation: An internal error in the driver code was detected.
User response: Send an error report to the developers.

IXF01002:
The LOB column index is invalid.
Explanation: The file contains an invalid index of a LOB column.
User response: The imported IXF file is corrupted or has an unknown format.

IXF01003:
The LOB column length is invalid.
Explanation: The IXF file contains a LOB column that has an invalid length.
User response: The imported IXF file is corrupted or has an unknown format.

IXF01004:
The end of the file was encountered prematurely.
Explanation: A premature end of file was encountered while importing data from the IXF file.
User response: The imported IXF file is corrupted or has an unknown format.

IXF01005:
The length of record {0} is invalid.
Explanation: A corrupted record was encountered while importing data from the IXF file.
User response: The imported IXF file is corrupted or has an unknown format.

IXF01006:
An invalid record length was encountered in record {0}.
Explanation: An invalid record length was encountered while importing data from the IXF file.
User response: The imported IXF file is corrupted or has an unknown format.

IXF01007:
An invalid name length was encountered in record {0}.
Explanation: An invalid column name length was encountered while importing data from the IXF file.
User response: The imported IXF file is corrupted or has an unknown format.

IXF01008:
An unrecognized record type was encountered in record {0}.
Explanation: An unknown record type was encountered while importing data from the IXF file.
User response: The imported IXF file is corrupted or has an unknown format.

IXF01009:
An invalid value was encountered in record {0}.
Explanation: An unexpected data value was encountered while importing data from the IXF file.
User response: The imported IXF file is corrupted or has an unknown format.

IXF01010:
An unsupported data type was encountered in record {0}.
Explanation: An unsupported data type was encountered while importing data from the IXF file.
User response: The imported IXF file is corrupted or has an unknown format.

IXF01011:
An unsupported CCSID was encountered in record {0}.
Explanation: The IXF file contains character data which is encoded using a CCSID that is not supported by CorVu NG.
User response: Create an IXF file using another CCSID to import to CorVu NG.

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 CorVu NG.
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.
<table>
<thead>
<tr>
<th>Message Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IXF01019</td>
<td>Data format is not supported in host IXF files.</td>
</tr>
<tr>
<td>Explanation</td>
<td>An unsupported column type was encountered while importing data from the IXF file.</td>
</tr>
<tr>
<td>User response</td>
<td>The imported IXF file is corrupted or has an unknown format.</td>
</tr>
<tr>
<td>LIC1000</td>
<td>License field &quot;[0]&quot; has invalid date format: {1}.</td>
</tr>
<tr>
<td>Explanation</td>
<td>The format of the date at the specified field is incorrect.</td>
</tr>
<tr>
<td>User response</td>
<td>Ensure that the license file contains a date in dd/mm/yyyy format.</td>
</tr>
<tr>
<td>LIC1001</td>
<td>Required field &quot;[0]&quot; missed.</td>
</tr>
<tr>
<td>Explanation</td>
<td>No value was specified for the indicated field.</td>
</tr>
<tr>
<td>User response</td>
<td>Ensure that a value is specified for the indicated field.</td>
</tr>
<tr>
<td>LIC1002</td>
<td>Field &quot;[0]&quot; has invalid value &quot;[1]&quot;. Valid value is {2}.</td>
</tr>
<tr>
<td>Explanation</td>
<td>An incorrect value was specified for the indicated field.</td>
</tr>
<tr>
<td>User response</td>
<td>Ensure that a valid license file is being used.</td>
</tr>
<tr>
<td>LIC1003</td>
<td>License file could not be found {0}.</td>
</tr>
<tr>
<td>Explanation</td>
<td>The license file could not be found.</td>
</tr>
<tr>
<td>User response</td>
<td>Use the Manage License dialog to add the license to the product.</td>
</tr>
<tr>
<td>LIC1004</td>
<td>Unexpected file error: {0}.</td>
</tr>
<tr>
<td>Explanation</td>
<td>A file system error occurred while loading the license file.</td>
</tr>
<tr>
<td>User response</td>
<td>Send an error report to the developers.</td>
</tr>
<tr>
<td>LIC1006</td>
<td>License signature verification failed.</td>
</tr>
<tr>
<td>Explanation</td>
<td>The license file digital signature verification failed.</td>
</tr>
<tr>
<td>User response</td>
<td>Provide a valid license file.</td>
</tr>
<tr>
<td>LIC1007</td>
<td>License has expired.</td>
</tr>
<tr>
<td>Explanation</td>
<td>The trial period of the license expired.</td>
</tr>
<tr>
<td>User response</td>
<td>Buy a full license.</td>
</tr>
<tr>
<td>LIC1008</td>
<td>Unexpected error.</td>
</tr>
<tr>
<td>Explanation</td>
<td>An unexpected error occurred while verifying the license.</td>
</tr>
<tr>
<td>User response</td>
<td>Ensure that the correct license file is being used.</td>
</tr>
<tr>
<td>LIC1009</td>
<td>The license is incorrect.</td>
</tr>
<tr>
<td>Explanation</td>
<td>The license verification failed because of an invalid license file.</td>
</tr>
<tr>
<td>User response</td>
<td>Use the original license file.</td>
</tr>
<tr>
<td>Code</td>
<td>Message</td>
</tr>
<tr>
<td>----------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>LIC1010</td>
<td>License file not found.</td>
</tr>
<tr>
<td>LIC1011</td>
<td>Could not load validation module.</td>
</tr>
<tr>
<td>MAIL1000</td>
<td>{0}</td>
</tr>
<tr>
<td>MAIL1001</td>
<td>SMTP server must be specified.</td>
</tr>
<tr>
<td>MAIL1002</td>
<td>Unknown type of attachment specified.</td>
</tr>
<tr>
<td>NLS00001</td>
<td>Generic error #{0} has occurred in module {1}.</td>
</tr>
<tr>
<td>NLS00002</td>
<td>Generic error #{0} has occurred in module {1} {{2}}.</td>
</tr>
<tr>
<td>NLS00003</td>
<td>Generic error #{0} has occurred in module {1} {{2}; {3}}.</td>
</tr>
<tr>
<td>NLS00004</td>
<td>Generic error #{0} has occurred in module {1} {{2}; {3}; {4}}.</td>
</tr>
<tr>
<td>NLS00005</td>
<td>Generic error #{0} has occurred in module {1} {{2}; {3}; {4}; {5}}.</td>
</tr>
</tbody>
</table>
Appendix B: Messages

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.
Appendix B: Messages

OLQ1001:
Internal Error.
Explanation: An internal error occurred.
User response: Collect support data by selecting Help → Collect Support Data and forward it to customer support.

OLQ1002:
Query was cancelled.
Explanation: The current query was cancelled.
User response: Rerun the query.

OLQ1003:
You must select at least one value for each level.
Explanation: At least one value must be selected in a slicer control.
User response: Select a value and try again.

OLQ1004:
Invalid type for saving the query.
Explanation: An invalid object type was selected in which to save the query. OLAP queries can only be saved in the *.oq file format.
User response: Save the OLAP query to a file with the *.oq file extension.

OLQ1005:
No data source was specified for the query.
Explanation: A nonexistent data source was specified for the OLAP query.
User response: Specify a valid data source for the OLAP query.

OLQ1006:
Only one reference to dimension hierarchy {0} can appear in the slicer.
Explanation: MDX DB2 DWE supports only one dimension per slicer.
User response: Select only one dimension.

PE00001:
Encountered "{0}" at column {1}. \nWas expecting one of: {2}.
Explanation: An invalid character was encountered at the indicated column.
User response: Replace the invalid character with one of the indicated characters.

PE00002:
Encountered "{0}" at column {1}. \nWas expecting: {2}.
Explanation: An invalid character was encountered at the indicated column.
User response: Replace the invalid character with the indicated character.

PE00003:
Expression needs a calculator to get a value.
Explanation: The expression does not return a value.
User response: No action is required.

PE00004:
Parse exception occurred.
Explanation: An error occurred while parsing the expression.
User response: Check the expression syntax for errors and try again.

PRC1000:
An unexpected error occurred while parsing the command.
Explanation: An internal error occurred.
User response: Collect support data by selecting Help → Collect Support Data and forward it to customer support.

PRC1001:
An error occurred while parsing the command: unexpected token "{0}".
Explanation: An unexpected token was encountered while parsing the procedure.
User response: Check the procedure's text for syntax errors (using the documentation or Content Assist via Ctrl +Space)

PRC1002:
An error occurred while parsing the command: unexpected character "{0}".
Explanation: An unexpected character was encountered while parsing the procedure.
User response: Check the procedure's text for syntax errors or invalid characters (using the documentation or Content Assist via Ctrl+Space)

PRC1003:
The parameter "{0}" is invalid in this context.
Explanation: An unsupported parameter token was encountered while parsing the procedure.
User response: Check the procedure's text for syntax errors (using the documentation or Content Assist via Ctrl +Space)

PRC1004:
The parameter "{0}" cannot be specified twice.
Explanation: The indicated parameter was specified twice in the same procedure command.
User response: Only specify one instance of the indicated parameter.

PRC1005:
"{1}" is not a valid value for parameter "{0}".
Explanation: The indicated value is not valid for the indicated parameter.
User response: Specify a valid value for the indicated parameter. See the Procedure Commands section of the user help for more information on procedure commands and their parameters.

PRC1006:
You cannot set variable "{0}".
Explanation: The indicated variable can not be set manually.
User response: Specify a valid variable and try again.

PRC1008:
An error occurred while parsing the command: unexpected end of line.
Explanation: An unexpected end of line was encountered while parsing the procedure.
User response: Check the procedure's text for syntax errors (using the documentation or Content Assist via Ctrl +Space)

PRC1009:
An error occurred while parsing the command: unrecognized command "{0}".
Explanation: The indicated command is not valid.
User response: Check the procedure text for errors or specify a different command.
### Appendix B: Messages

<table>
<thead>
<tr>
<th>Message ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRC1010:</td>
<td>Parsing error at lines {0}-{1} (command #{2}). {3}</td>
</tr>
<tr>
<td></td>
<td><strong>Explanation:</strong> A syntax error was encountered while parsing procedure.</td>
</tr>
<tr>
<td></td>
<td><strong>User response:</strong> Check the procedure's text for syntax errors (using the documentation or Content Assist via Ctrl + Space)</td>
</tr>
<tr>
<td>PRC1011:</td>
<td>An error occurred while parsing the command: invalid object name &quot;'{0}'&quot;.</td>
</tr>
<tr>
<td></td>
<td><strong>Explanation:</strong> The indicated object name is invalid.</td>
</tr>
<tr>
<td></td>
<td><strong>User response:</strong> Check the procedure text for errors and specify a valid object name. See the Procedure Commands section of the user help for more information.</td>
</tr>
<tr>
<td>PRC1012:</td>
<td>You must specify object type or object name.</td>
</tr>
<tr>
<td></td>
<td><strong>Explanation:</strong> A required object type or name was not specified.</td>
</tr>
<tr>
<td></td>
<td><strong>User response:</strong> 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.</td>
</tr>
<tr>
<td>PRC1013:</td>
<td>&quot;'{0}'&quot; is an invalid variable name.</td>
</tr>
<tr>
<td></td>
<td><strong>Explanation:</strong> The indicated variable name is invalid.</td>
</tr>
<tr>
<td></td>
<td><strong>User response:</strong> Check the variable name for syntax errors or specify another valid variable. Check the Procedure Commands section of the user help for more information.</td>
</tr>
<tr>
<td>PRC1014:</td>
<td>Close bracket reached before end.</td>
</tr>
<tr>
<td></td>
<td><strong>Explanation:</strong> An unexpected close bracket was encountered while parsing the procedure.</td>
</tr>
<tr>
<td></td>
<td><strong>User response:</strong> Check the procedure's text for syntax errors (using the documentation or Content Assist via Ctrl + Space)</td>
</tr>
<tr>
<td>PRC2001:</td>
<td>An error occurred while retrieving {0}. {1}</td>
</tr>
<tr>
<td></td>
<td><strong>Explanation:</strong> The indicated error occurred while retrieving the indicated object.</td>
</tr>
<tr>
<td></td>
<td><strong>User response:</strong> Resolve the error and rerun the procedure.</td>
</tr>
<tr>
<td>PRC2002:</td>
<td>{0} does not exist.</td>
</tr>
<tr>
<td></td>
<td><strong>Explanation:</strong> The indicated object does not exist.</td>
</tr>
<tr>
<td></td>
<td><strong>User response:</strong> Specify a valid object in the procedure command.</td>
</tr>
<tr>
<td>PRC2003:</td>
<td>{0} exists but is of the wrong type.</td>
</tr>
<tr>
<td></td>
<td><strong>Explanation:</strong> 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.</td>
</tr>
<tr>
<td></td>
<td><strong>User response:</strong> 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.</td>
</tr>
<tr>
<td>PRC2004:</td>
<td>{0} does not exist or is of the wrong type.</td>
</tr>
</tbody>
</table>
Messages

**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)

PRC2035:
The printer name "{0}" is invalid.

Explanation: The indicated printer name is invalid.

User response: Check the spelling of the printer name, or specify a different printer to use for the PRINT command.

PRC2036:
You are not authorized to use the table editor at this data source.

Explanation: The Enable Table Editing resource limit for the data source is turned off; you can not open Table Editor of the data source

User response: Ask your administrator to enable this resource limit.

PRC2037:
You are not authorized to edit tables at this data source.
**Explanation:** The Enable Table Editing resource limit for the data source is turned off; you cannot edit values in a query's grid cells.

**User response:** Ask your administrator to enable this resource limit.

**PROC1000:**
REXX procedure references cannot be determined.

**Explanation:** References for the procedure cannot be built for REXX procedures.

**User response:** You can view references at the Repository or Workspaces views for generic procedures only.

**PROC1001:**
Some substitution variables are not defined.

**Explanation:** Some of the substitution variables of the procedure are not defined.

**User response:** Ensure that all of the substitution variables are defined for the procedure, or run this procedure without the /batch parameter in the command line. In this case CorVu NG will ask for all of the necessary variables values.

**QEL00001:**
You are not allowed to run queries returning LOB data.

**Explanation:** You are not allowed to run queries that contain large object data.

**User response:** Check your LOB Options resource limits by selecting Resource Limits from the View menu to see your specific allowances or ask your administrator for more information.

**QEL00002:**
The Maximum Rows to Fetch limit has been exceeded. {0} rows were retrieved so far.

**Explanation:** The Maximum Rows to Fetch limit has been reached. The indicated number of rows have been retrieved so far.

**User response:** Check your Maximum Rows to Fetch resource limit by selecting Resource Limits from the View menu to see your specific allowances or ask your administrator for more information.

**QEL00003:**
The Maximum Bytes to Fetch limit has been exceeded. {0} bytes were retrieved so far.

**Explanation:** The Maximum Bytes to Fetch limit has been reached. The indicated number of bytes have been retrieved so far.

**User response:** Check your Maximum Bytes to Fetch resource limit by selecting Resource Limits from the View menu to see your specific allowances or ask your administrator for more information.

**QEL00004:**
The length of a LOB column exceeds the maximum allowable length.

**Explanation:** The length of a LOB column in the query results exceeds the maximum allowable length.

**User response:** Check your Maximum LOB column size by selecting Resource Limits from the View menu to see your specific allowances or ask your administrator for more information.

**QEL00005:**
You are not allowed to use the "{0}" SQL verb.

**Explanation:** You are not allowed to use the indicated SQL verb.

**User response:** Check your SQL Verbs limits by selecting Resource Limits from the View menu to see your specific allowances or ask your administrator for more information.

**QEL00006:**
You are not allowed to save data to database.

**Explanation:** You are not allowed to save data to the database.
User response: Check your Save Data resource limits by selecting **Resource Limits** from the **View** menu to see your specific allowances or ask your administrator for more information.

**QEN00001:**
*Query was cancelled.*

**Explanation:** The active query was cancelled.

**User response:** Rerun the query.

**QEN00003:**
*Unexpected token: {0}*

**Explanation:** An unexpected token was encountered while parsing the query that calls the stored procedure.

**User response:** Check the query's syntax for errors.

**QEN00004:**
*Unexpected token: {0} is expected but {1} was found.*

**Explanation:** An unexpected closed bracket was encountered while parsing the query that calls the stored procedure.

**User response:** Check the query's syntax for errors.

**QEN00005:**
*Unexpected end of statement.*

**Explanation:** An unexpected end of statement was encountered while parsing the query that calls the stored procedure.

**User response:** Check the query's syntax for errors.

**QEN00006:**
*Expression {0} is not a stored procedure.*

**Explanation:** An internal error occurred.

**User response:** Collect support data by selecting **Help → Collect Support Data** and forward it to customer support.

**QEN00007:**
*End of statement is expected but {0} was found*

**Explanation:** Tokens were encountered after the end of statement while parsing the query that calls the stored procedure.

**User response:** Check the query's syntax for errors.

**QEN00008:**
*Unsupported parameter data type: {0}*

**Explanation:** An unsupported parameter was encountered while parsing the query that calls the stored procedure.

**User response:** Check the query's syntax for errors.

**QEN00009:**
*{0} is invalid value for this type: {1}*

**Explanation:** When prompted for a value for a stored procedure parameter, the user entered a value that is not valid for that parameter's type.

**User response:** Specify a valid value for the parameter.

**QRL00001:**
*You are not authorized to access this data source in user interface mode.*

**Explanation:** You are not authorized to access this data source in user interface mode.
**User response:** Check your Options limits by selecting **Resource Limits** from the **View** menu to see your specific allowances or ask your administrator for more information.

---

**QRL00002:**
**You are not authorized to access this data source in automation mode.**

**Explanation:** You are not authorized to access this data source in automation mode.

**User response:** Check your Options limits by selecting **Resource Limits** from the **View** menu to see your specific allowances or ask your administrator for more information.

---

**QRL00003:**
**You are not authorized to run unsaved queries.**

**Explanation:** You are not authorized to run unsaved queries.

**User response:** Check your Options limits by selecting **Resource Limits** from the **View** menu to see your specific allowances or ask your administrator for more information.

---

**QRL00004:**
**You are not authorized to save data from this data source to a file.**

**Explanation:** You are not authorized to save data from this data source to a file.

**User response:** Check your Save Data limits by selecting **Resource Limits** from the **View** menu to see your specific allowances or ask your administrator for more information.

---

**QRL00005:**
**You are not authorized to save data from this data source to the database.**

**Explanation:** You are not authorized to save data from this data source to the database.

**User response:** Check your Save Data limits by selecting **Resource Limits** from the **View** menu to see your specific allowances or ask your administrator for more information.

---

**QRL00006:**
**You are not authorized to retrieve LOB data.**

**Explanation:** You are not authorized to retrieve LOB data.

**User response:** Check your LOB Options limits by selecting **Resource Limits** from the **View** menu to see your specific allowances or ask your administrator for more information.

---

**QRL00008:**
**No resource group for user {0} was found.**

**Explanation:** No resource group for the indicated user was found.

**User response:** Specify a valid resource group for the user.

---

**QRL00009:**
**The operation cannot be performed because the processing of LOB data is disabled.**

**Explanation:** The operation cannot be performed because the processing of LOB data is disabled.

**User response:** Check you LOB Options limits by selecting **Resource Limits** from the **View** menu to see your specific allowances or ask your administrator for more information.

---

**QRY0999:**
{0}

**Explanation:** This is a wrapped message.

**User response:** Resolve the indicated error and try again.

---

**QRY1001:**
Query was cancelled.
<table>
<thead>
<tr>
<th>Message ID</th>
<th>Message</th>
<th>Explanation</th>
<th>User Response</th>
</tr>
</thead>
<tbody>
<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>QRY1019:</td>
<td>A column on which to join tables is in more than one of the tables in the query.</td>
<td>A join column can only exist in one table when multiple tables are joined in a single query.</td>
<td>Ensure that any join columns in the query exist in only one table.</td>
</tr>
<tr>
<td>QRY1020:</td>
<td>A column on which to join tables is not in any of the tables in the query.</td>
<td>A specified join column is not present in any of the tables in the query.</td>
<td>Check the join column name for errors, or specify a different join column.</td>
</tr>
<tr>
<td>QRY1021:</td>
<td>Type of join columns for joined tables mismatch.</td>
<td>A columns that join two or more tables must all be of the same column type.</td>
<td>Ensure that all of the columns that make up a single join are of the same column type.</td>
</tr>
<tr>
<td>QRY1022:</td>
<td>The expression contains a syntax error: {0}</td>
<td>The indicated expression contains a syntax error.</td>
<td>Correct the syntax error and rerun the query.</td>
</tr>
<tr>
<td>QRY1023:</td>
<td>An invalid connector for a row condition was encountered.</td>
<td>An invalid connector for a row condition was encountered.</td>
<td>Check all row conditions for errors and rerun the query.</td>
</tr>
<tr>
<td>QRY1024:</td>
<td>An invalid expression type in a row condition was encountered.</td>
<td>An invalid expression type in a row condition was encountered.</td>
<td>Check all expression types in the row conditions for errors and rerun the query.</td>
</tr>
<tr>
<td>QRY1025:</td>
<td>An invalid operator in a row condition was encountered.</td>
<td>An invalid operator in a row condition was encountered.</td>
<td>Check all operators in the row conditions for errors and rerun the query.</td>
</tr>
<tr>
<td>QRY1026:</td>
<td>The selected relational operator is not valid for numeric data.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Explanation: An invalid operator in a row condition was encountered for numeric data.

User response: Check all of the operators in the row conditions for errors and rerun the query.

QRY1027:
An invalid sort direction was encountered.

Explanation: An invalid sort direction was encountered. Valid directions are Ascending or Descending.

User response: Check all instances where sort directions are applied and change them to either ascending or descending.

QRY1029:
The table {0} does not exist.

Explanation: The indicated table is called in the SQL code, but does not exist in the data source.

User response: Check the table name for errors, or enter a different table to be called.

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.

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.

QRY1044:
Only simple LIKE statements are supported.

Explanation: Only simple LIKE statements are supported.

User response: Ensure that the specified query only contains simple LIKE statements.

QRY1045:
Only simple row conditions are supported.

Explanation: Only simple row conditions are supported.

User response: Ensure that the specified query only contains simple row conditions.

QRY1046:
Expression cannot be empty.

Explanation: An error occurred while constructing the query.

User response: Check the query's syntax for errors and try again.

QRY1047:
Invalid type for saving the query.

Explanation: The type that was set for the query is invalid.

User response: Specify a valid type for the query or collect support data by selecting Help → Collect Support Data and forward it to customer support.

QRY1048:
You cannot run an empty query.

Explanation: The specified query did not contain any SQL code.

User response: Populate the query with at least a SELECT statement and rerun.
<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
<th>Explanation</th>
<th>User response</th>
</tr>
</thead>
<tbody>
<tr>
<td>QRY1049</td>
<td>Some substitution variables are not defined.</td>
<td>Some of the substitution variables in the query were not given values when the query was run.</td>
<td>Ensure that all of the substitution variables in the query are defined.</td>
</tr>
<tr>
<td>QRY1053</td>
<td>Only SELECT statements are supported in compound queries.</td>
<td>Only SELECT statements are supported in compound queries.</td>
<td>Ensure that only SELECT statements are called in the specified compound query.</td>
</tr>
<tr>
<td>RCO0000</td>
<td>Internal error</td>
<td>An internal processor error occurred.</td>
<td>Collect support data by selecting Help → Collect Support Data and forward it to customer support.</td>
</tr>
<tr>
<td>RCO0001</td>
<td>Type mismatch</td>
<td>An unexpected error occurred while processing the *.rco file.</td>
<td>Collect support data by selecting Help → Collect Support Data and forward it to customer support.</td>
</tr>
<tr>
<td>RCO0002</td>
<td>Function {0} is not implemented yet</td>
<td>An unexpected error occurred while processing the *.rco file.</td>
<td>Collect support data by selecting Help → Collect Support Data and forward it to customer support.</td>
</tr>
<tr>
<td>RCO0003</td>
<td>Undefined variable {0}</td>
<td>An unexpected error occurred while processing the *.rco file.</td>
<td>Collect support data by selecting Help → Collect Support Data and forward it to customer support.</td>
</tr>
<tr>
<td>RCO0004</td>
<td>Missing variable name</td>
<td>An unexpected error occurred while processing the *.rco file.</td>
<td>Collect support data by selecting Help → Collect Support Data and forward it to customer support.</td>
</tr>
<tr>
<td>RCO0005</td>
<td>Syntax error in &quot;{0}&quot;</td>
<td>An unexpected error occurred while processing the *.rco file.</td>
<td>Collect support data by selecting Help → Collect Support Data and forward it to customer support.</td>
</tr>
<tr>
<td>RCO0006</td>
<td>Unknown function {0}</td>
<td>An unexpected error occurred while processing the *.rco file.</td>
<td>Collect support data by selecting Help → Collect Support Data and forward it to customer support.</td>
</tr>
<tr>
<td>RCO0007</td>
<td>Unknown object type {0}</td>
<td>An unexpected error occurred while processing the *.rco file.</td>
<td>Collect support data by selecting Help → Collect Support Data and forward it to customer support.</td>
</tr>
</tbody>
</table>
RCO0008:
Wrong parameters in "\{0\}"
Explanation: An unexpected error occurred while processing the *.rco file.
User response: Collect support data by selecting Help → Collect Support Data and forward it to customer support.

RCO0009:
Database object {0} not found
Explanation: An unexpected error occurred while processing the *.rco file.
User response: Collect support data by selecting Help → Collect Support Data and forward it to customer support.

RCO0010:
Database error {0}
Explanation: An unexpected error occurred while processing the *.rco file.
User response: Collect support data by selecting Help → Collect Support Data and forward it to customer support.

RCO0011:
Unknown database server type {0}
Explanation: An unexpected error occurred while processing the *.rco file.
User response: Collect support data by selecting Help → Collect Support Data and forward it to customer support.

REP01000:
{0}
Explanation: The indicated error occurred while accessing the repository.
User response: Resolve the indicated error and try again.

REP01001:
An input/output error occurred: {0}
Explanation: The indicated error occurred in the repository.
User response: Resolve the indicated error and try again.

REP01002:
Character encoding is not supported: {0}
Explanation: The indicated encoding error occurred while the data was being processed.
User response: Ensure that UTF-8 encoding is used in the repository storage database.

REP01003:
Internal error has occurred: {0}
Explanation: The indicated error occurred.
User response: Resolve the indicated error and try again.

REP01004:
The following error has occurred while accessing database: {0}
Explanation: The indicated error occurred while the database was being accessed.
User response: Resolve the indicated error and retry to connect to the database.

REP01005:
Operation is not supported.
Explanation: The specified operation is not supported by the repository.
User response: Specify a different operation and try again.
### REP01006
Object with such identifier not found.

**Explanation:** An error occurred while searching for the object.

**User response:** Ensure that the specified object identifier is correct.

### REP01007
Object or file with URL "{0}" not found.

**Explanation:** There are no objects or files with the indicated URL.

**User response:** Check the URL for errors or specify a different URL.

### REP01008
Not enough permissions to perform operation.

**Explanation:** Your user ID does not have permission to perform the specified operation.

**User response:** Contact your administrator to check your specific permissions or specify a different operation to perform.

### REP01009
Operation cancelled due to invalid object identifier: {0}

**Explanation:** The operation was cancelled because the indicated object identifier is invalid.

**User response:** Specify a different object identifier and retry the specified operation.

### REP01010
Cannot open connection: {0}

**Explanation:** The indicated error occurred while establishing a connection to the repository database.

**User response:** Ensure that the repository database is accessible.

### REP01011
At least one line of the document is longer than 79 characters, which is the maximum supported length when saving objects in the database.

**Explanation:** The maximum character length allowed for documents saved to the database is seventy-nine characters. The document contains at least one line that is longer than seventy-nine characters.

**User response:** Ensure that all of the lines of the document are less than seventy-nine characters and retry to save the document.

### REP01012
{0} already exists and has a different type.

**Explanation:** The indicated object name already exists as a different type of object. Each object in the database must have a different name.

**User response:** Specify a different name for the object and retry to save the object.

### REP01013
{0} does not exist.

**Explanation:** The indicated object could not be found because it does not exist in the database.

**User response:** Specify a different object name to search for in the database.

### REP01014
{0} already exists.

**Explanation:** The indicated object could not be saved because it already exists in the database.

**User response:** Specify a different name for the object and retry to save the object.
<table>
<thead>
<tr>
<th>REP01015:</th>
<th>The indicated object is of an unsupported type.</th>
</tr>
</thead>
<tbody>
<tr>
<td>{0} has unsupported type and cannot be accessed.</td>
<td>User response: Ensure that all of the necessary plug-ins are installed and available.</td>
</tr>
<tr>
<td>REP01016:</td>
<td>The specified operation could not be performed because the specified object is too large.</td>
</tr>
<tr>
<td>Operation is impossible because object is too large.</td>
<td>User response: Specify a different object or a different operation and try again.</td>
</tr>
<tr>
<td>REP01017:</td>
<td>An internal error occurred in the repository.</td>
</tr>
<tr>
<td>Illegal state or illegal operation error.</td>
<td>User response: Collect support data by selecting Help → Collect Support Data and forward it to customer support.</td>
</tr>
<tr>
<td>REP01018:</td>
<td>You can not perform the specified operation, because you are not connected to the repository.</td>
</tr>
<tr>
<td>User is not logged on repository.</td>
<td>User response: Connect to the repository and try again.</td>
</tr>
<tr>
<td>REP01019:</td>
<td>The indicated error occurred while attempting to access a non-implemented functionality.</td>
</tr>
<tr>
<td>Not implemented.</td>
<td>User response: Ensure that you are using the latest version of the product.</td>
</tr>
<tr>
<td>REP01020:</td>
<td>An internal error occurred in the repository.</td>
</tr>
<tr>
<td>Unexpected error occurred.</td>
<td>User response: Collect support data by selecting Help → Collect Support Data and forward it to customer support.</td>
</tr>
<tr>
<td>REP01021:</td>
<td>An error occurred while the repository was being initialized.</td>
</tr>
<tr>
<td>Initialization error occurred &quot;{0}&quot;.</td>
<td>User response: Check the repository initialization parameters and try again.</td>
</tr>
<tr>
<td>REP01022:</td>
<td>An error occurred while the repository was being initialized.</td>
</tr>
<tr>
<td>Invalid initialization parameters were specified &quot;{0}&quot;.</td>
<td>User response: Check the repository initialization parameters and try again.</td>
</tr>
<tr>
<td>REP01027:</td>
<td>An error occurred while trying to access a non-initialized repository.</td>
</tr>
<tr>
<td>A repository instance hasn't been initialized.</td>
<td>User response: Initialize the repository and try again.</td>
</tr>
<tr>
<td>REP01028:</td>
<td>The indicated repository could not be found.</td>
</tr>
<tr>
<td>Repository &quot;{0}&quot; could not be found.</td>
<td>User response: Check the repository name for errors or specify a different repository and try again.</td>
</tr>
</tbody>
</table>
Appendix B: Messages

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.

REP01034:
A repository with name {0} already exists.
Explanation: The repository could not be created, because a repository of the same name already exists.
User response: Specify a different name for the new repository and try again.

REP01035:
Unknown object type.
Explanation: An error occurred while trying to access an unsupported object.
User response: Ensure that all of the required plug-ins are installed and available.

REP01036:
Object {0} cannot be added to this folder.
Explanation: The indicated object can not be added to the specified folder.
User response: Specify a different folder in which to add the indicated object.

REP01037:
Repository could not be found.
Explanation: The specified repository could not be found.
User response: Check the repository name for errors or specify a different repository name and try again.

REP01038:
Undefined or wrong repository identifier
Explanation: An error occurred while trying to initialize a repository with deprecated initialization parameters.
User response: Change the repository connection properties through the Edit Repository Connection dialog and try again.

REP01042:
You are not authorized to use objects with Owner = {0}.
Explanation: Your user ID is not authorized to use objects with the indicated owner.
User response: Specify an object that has a different owner or contact your administrator for more information.

REP01043:
This data source cannot be accessed by CorVu NG.
Explanation: The application can not access this data source.
User response: Check your user license and try again.

**REP01046:**
Cannot find connection: {0}
Explanation: The indicated repository connection could not be found.
User response: Specify a different repository connection and try again.

**REP01047:**
Cannot find data source: {0}
Explanation: The indicated data source could not be found.
User response: Specify a different data source and try again.

**REP01049:**
Object name cannot exceed [0] characters.
Explanation: The indicated object name exceeds the maximum character limit.
User response: Shorten the indicated object name and try again.

**REP01050:**
Object path cannot exceed [0] characters.
Explanation: The indicated object path exceeds the character limit.
User response: Shorten the indicated object path and try again.

**REP01051:**
Cannot connect to repository "{0}" because it is used by another instance of CorVu NG. Select different repository.
Explanation: You can not connect to the indicated repository because it is being used by another instance of the application.
User response: Select a different repository connection and try again.

**REP01052:**
Data Source plug-in "{0}" was not properly initialized.
Explanation: The indicated plug-in was not properly initialized.
User response: Change the plug-in parameters.

**RGEN1000:**
An error has occurred while generating the report: {0}
Explanation: The indicated error occurred while the report was being generated.
User response: Resolve the indicated error and retry to generate the report.

**RGEN1001:**
The action was cancelled.
Explanation: The specified action was cancelled
User response: Rerun the specified action.

**RGEN1002:**
The number of columns in the form does not match the number of columns in the query.
Explanation: The number of columns in the form does not match the number of columns in its associated query.
User response: Edit the number of columns in the form to match the number of columns its associated query.

**RGEN1003:**
Internal error.
Appendix B: Messages

**Explanation**: An internal error occurred.

**User response**: Collect support data by selecting Help → Collect Support Data and forward it to customer support.

---

**RGEN1004**: The number of columns in the form does not match the number of columns in the query.

**Explanation**: The number of columns in the form does not match the number of columns in its associated query.

**User response**: Edit the number of columns in the form to match the number of columns its associated query.

---

**RGEN1005**: The data source object is not specified.

**Explanation**: The data source object for the report is not specified.

**User response**: Specify a data source object for the report and try again.

---

**RLE00002**: Cannot retrieve resource limits of type {0}.

**Explanation**: The resources of the indicated type can not be retrieved.

**User response**: Collect support data by selecting Help → Collect Support Data and forward it to customer support.

---

**RLE00003**: An error has occurred while retrieving resource limits from remote site: {0}

**Explanation**: The indicated error occurred while resource limits were being obtained from a remote site.

**User response**: Resolve the indicated error and try again.

---

**RLE00004**: Unknown resource limits key is used to retrieve limits.

**Explanation**: An error occurred while retrieving resource limits.

**User response**: Collect support data by selecting Help → Collect Support Data and forward it to customer support.

---

**RLE00005**: An error has occurred while obtaining resource limits: {0}

**Explanation**: The indicated error occurred while resource limits were being obtained.

**User response**: Resolve the indicated error and try again.

---

**RLE00006**: Cannot retrieve resource limits because resource limits key content is corrupted. {0}

**Explanation**: The indicated error occurred while attempting to retrieve resources limits.

**User response**: Resolve the indicated error and try again.

---

**RLE00007**: The following error has occurred while retrieving resource limits: {0}

**Explanation**: The indicated error occurred while resource limits were being retrieved.

**User response**: Resolve the error and try again.

---

**RLE00008**: Data source cannot be accessed.

**Explanation**: Your user ID does not have permission to access the data source.

**User response**: Check your resource limits by selecting Resource Limits from the View menu or contact your administrator for more information.
<table>
<thead>
<tr>
<th>Message ID</th>
<th>Description</th>
<th>Explanation</th>
<th>User Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>RLE00009:</td>
<td>LOB data saving is denied.</td>
<td>Your user ID does not have permission to save objects that contain LOB data.</td>
<td>Check your LOB Options resource limits by selecting Resource Limits from the View menu or contact your administrator for more information.</td>
</tr>
<tr>
<td>RLE00010:</td>
<td>You are only authorized to run saved queries at this server.</td>
<td>You are only allowed to run saved queries while you are connected to the current server.</td>
<td>Check you Save Data resource limits by selecting Resource Limits from the View menu or contact your administrator for more information.</td>
</tr>
<tr>
<td>RLE00011:</td>
<td>You are not authorized to save objects at this server.</td>
<td>You are not authorized to save objects while you are connected to the current server.</td>
<td>Check your Save Data resource limits by selecting Resource Limits from the View menu or contact your administrator for more information.</td>
</tr>
<tr>
<td>RLE00012:</td>
<td>You are not authorized to access data source {0}.</td>
<td>Your user ID is not authorized to access the indicated data source.</td>
<td>Try to connect to another data source, or contact your administrator for information regarding specific allowances.</td>
</tr>
<tr>
<td>RLE00013:</td>
<td>You are not authorized to access data source {0}.</td>
<td>Your user ID is not authorized to access the indicated data source.</td>
<td>Try to connect to another data source, or contact your administrator for information regarding specific allowances.</td>
</tr>
<tr>
<td>RLW00001:</td>
<td>{0} rows were retrieved. The Maximum Rows to Fetch warning limit has been exceeded.</td>
<td>The indicated number of rows were retrieved and the warning limit for the Maximum Rows to Fetch limit has been reached.</td>
<td>Click OK to continue running the query. Click Abort to abort the query. Click Ignore to ignore the warning.</td>
</tr>
<tr>
<td>RLW00002:</td>
<td>{0} bytes were retrieved. The Maximum Bytes to Fetch warning limit has been exceeded.</td>
<td>The indicated number of bytes were retrieved and the warning limit for the Maximum Bytes to Fetch limits has been reached.</td>
<td>Click OK to continue running the query. Click Abort to abort the query. Click Ignore to ignore the warning.</td>
</tr>
<tr>
<td>RLW00003:</td>
<td>There has been no response from {0}.{1} for {2} seconds.</td>
<td>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.</td>
<td>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.</td>
</tr>
</tbody>
</table>
Appendix B: Messages

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 CorVu NG.

---

**RPT00044:**

**Unable to read visual dashboard format file.**

**Explanation:** {CONFIG_DIR}/converter/repository/world.xml can not be found or read.

**User response:** Reinstall CorVu NG.
Appendix B: Messages

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 CorVu NG.

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 to file. - Offline visual reports can not be saved to a file.
User response:
1. Reload the visual project.
2. Convert the visual report to an online report before saving to a file.

RPT10001: Source Visionary World file has not been specified.
Explanation: The Visionary World file can not be found.
User response: Ensure that the Visionary World file exists and that it can be read.

RPT10002: Configuration directory has not been specified.
Explanation: The configuration directory for the converter can not be found.
User response: Reinstall CorVu NG.

RPT10003: Source Visionary World file has not been found.
Explanation: The Visionary World file can not be found.
User response: Ensure that the Visionary World file exists.

RPT10004: Configuration directory has not been found.
Appendix B: Messages

**Explanation:** The configuration directory for the converter can not be found.
**User response:** Reinstall CorVu NG.

**RPT10005:**
Configuration file {0} has not been found.
**Explanation:** The configuration file for the converter can not be found.
**User response:** Reinstall CorVu NG.

**RPT10006:**
Unable to locate Visionary Connection file(s) in the folder {0}.
**Explanation:** The Visionary Connection files do not exist in the proper locations.
**User response:** Check that all Visionary World files are saved in the correct locations.

**RPT10007:**
Unable to read Visionary Query file {0}.
**Explanation:** The Visionary World file that is specified in the query can not be found.
**User response:** Check that all Visionary World files are saved in the correct locations.

**RPT10008:**
Unable to locate Visionary Scene file {0}.
**Explanation:** The indicated scene file is not saved in the correct location.
**User response:** Check that all Visionary World files are saved in the correct locations.

**RPT10009:**
Unable to read an image file {0}.
**Explanation:** The image file that is specified by the indicated name can not be found.
**User response:** Check that all Visionary World files are saved in the correct locations.

**RPT10010:**
Picture file {0} is too large.
**Explanation:** The indicated picture file is too long.
**User response:** Specify a smaller picture file to be used in the visual project.

**RPT10011:**
XML parsing error.
**Explanation:** An XML parsing error occurred while attempting to convert the Visionary World.
**User response:** Save the Visionary World file in Visionary and try again.

**RPT10012:**
XML reading error.
**Explanation:** An input/output error occurred while attempting to read the XML data.
**User response:** Try to convert the XML data again.

**RPT10013:**
Unable to create Visual Dashboard object.
**Explanation:** An error occurred while attempting to convert the Visionary World file.
**User response:** Try to convert the Visionary World file again.

**RPT10014:**
Cannot parse configuration file {0}.
Explanation: An error occurred while attempting to parse the converter configuration files.
User response: Reinstall CorVu NG.

RPT10015:
Visual Dashboard for a connection is not specified.
Explanation: The Visionary World file was not loaded properly.
User response: Reload the Visionary World file.

RPT10016:
Invalid color description has been found.
Explanation: Corrupted tags were encountered while loading a ColorMap or ColorSequence.
User response: Reload the Visionary World file.

RPT10017:
Invalid color map description has been found.
Explanation: Corrupted tags were encountered while loading a ColorMap.
User response: Reload the Visionary World file.

RPT10018:
Unknown tag: {0}
Explanation: Unknown tags were encountered while attempting to convert the Visionary World file.
User response: Try to convert the Visionary World file again.

RPT10019:
Invalid scene template description.
Explanation: An error occurred while loading the Visionary World scene file.
User response: Reload the Visionary World scene file.

RPT10020:
Corrupted property value has been found.
Explanation: An incorrect format or a corrupted value was specified in a property tag.
User response: Reload the Visionary World file.

RPT10021:
Invalid URL or path format: "'{0}'"
Explanation: The indicated URL or path format is invalid.
User response: Check the URL for errors or specify a different URL.

RPT10022:
Unknown or unsupported event type
Explanation: A corrupted or unknown event type was encountered.
User response: Reload the Visionary World file.

RPT10023:
Function name in expression is missing
Explanation: A function name in an expression is missing.
User response: Reload the Visionary World file.

RPT10024:
Cannot parse function argument
Appendix B: Messages

Explanation: The function argument could not be parsed while loading the Visionary World file.
User response: Reload the Visionary World file.

RPT10025:
Invalid scene template description.
Explanation: An error occurred while loading the Visionary World scene file.
User response: Reload the Visionary World file.

RPT10026:
Unable to parse a number: {0}
Explanation: While loading the Visionary World file, a string that was supposed to contain a number value was encountered without digit characters.
User response: Reload the Visionary World file.

RPT10027:
PointSet component XML is corrupted.
Explanation: While loading the Visionary World file, a string that was supposed to contain a point value was encountered with characters that could not be parsed into PointSet value.
User response: Reload the Visionary World file.

RPT10028:
Unknown expression type {0}.
Explanation: An unknown expression type was encountered while loading the Visionary World file.
User response: Reload the Visionary World file.

RPT10029:
One or more queries do not have alias.
Explanation: A query alias tag that does not contain a name property was encountered while converting the Visionary World file.
User response: Reload the Visionary World file.

RPT10031:
Unable to parse a point value.
Explanation: While loading the Visionary World file, a string that was supposed to contain a point value was encountered with characters that could not be parsed into PointSet value.
User response: Reload the Visionary World file.

RPT10032:
Visionary World XML has been corrupted.
Explanation: An error occurred while attempting to load the Visionary World file.
User response: Reload the Visionary World file.

RPT10033:
Internal error: unable to access Visionary World.
Explanation: An internal error occurred.
User response: Collect support data by selecting Help → Collect Support Data and forward it to customer support.

RPT10034:
Unable to add parameter {0} to a Visual Dashboard
Explanation: An internal error occurred.
User response: Collect support data by selecting Help → Collect Support Data and forward it to customer support.

RPT20001:
Unable to access repository during Visual Report conversion.
Explanation: The repository could not be accessed while the Visual Report was being converted.
User response: This could possibly be because you were not connected to the repository. Make sure you are connected and try again.

RPT20002:
Source visual report file has not been specified.
Explanation: The source visual report file has not been specified.
User response: Specify the source visual report file.

RPT20003:
Source visual report file has not been found.
Explanation: The source visual report file could not be found.
User response: Specify a valid source visual report file.

RPT20004:
Visual Report name has not been specified.
Explanation: The visual report name has not been specified.
User response: Specify a name for the visual report.

RPT20005:
Unable to create visual report.
Explanation: An error occurred while creating the visual report.
User response: Reload the visual report.

RPT20006:
Unable to open an embedded image.
Explanation: An embedded image file was not found.
User response: Reload the visual report.

RPT20007:
Unknown section type: {0}
Explanation: The structure of a section set and type was not found while loading the visual report.
User response: Reload the visual report.

RPT20008:
Unable to create Visual Report page component.
Explanation: An error occurred while parsing and creating a page component.
User response: Reload the visual report.

RPT20009:
Internal error: unable to access Visual Report.
Explanation: An internal error occurred.
User response: Collect support data by selecting Help → Collect Support Data and forward it to customer support.

RPT20011:
Unable to set a line style for a line control.
Appendix B: Messages

**Explanation:** An undefined line style was encountered while loading the visual report.

**User response:** Reload the visual report.

---

**RPT20012:**
Invalid linked picture alias has been found in Visual Report.

**Explanation:** A link to a picture without a name was encountered while parsing the visual report data.

**User response:** Reload the visual report.

---

**RPT20013:**
Unable to create Visual Report control.

**Explanation:** An error occurred while loading a primitive or control object.

**User response:** Reload the visual report.

---

**RPT20015:**
Invalid parameter name "{'0}'' has been found.

**Explanation:** The indicated parameter name is invalid.

**User response:** Specify a different name for the indicated parameter.

---

**RPT20016:**
Invalid parameter value "{'0}'' has been found.

**Explanation:** The indicated parameter value is invalid.

**User response:** Specify a different value for the indicated parameter.

---

**RPT30001:**
Navigate to Embedded Scene: Embedded Scene not found for name "{'0}''.

**Explanation:** The Navigate to Embedded Scene action failed because the indicated embedded scene could not be found.

**User response:** Correct the name of the embedded scene that is associated with the Navigate to Embedded Scene action, or specify another embedded scene.

---

**RPT30002:**
Navigate back: can`t execute exit action (maybe history is empty).

**Explanation:** The Navigate Back action failed, possibly because the navigation history is empty.

**User response:** Reload the visual project and try again.

---

**RPT30021:**
DesignReporter: LevelTemplate is null.

**Explanation:** An internal error occurred.

**User response:** Collect support data by selecting Help → Collect Support Data and forward it to customer support.

---

**RPT30022:**
DesignReporter: unknown component {0}.

**Explanation:** An unknown component was encountered during runtime rendering.

**User response:** Reload the visual project.

---

**RPT30024:**
The template is invalid. FillStyle or LineStyle property is null!

**Explanation:** An error occurred while running the substitution rule to change the Wedge template.

**User response:** Check the FillStyle and LineStyle properties for errors.
RPT30025:
The template is invalid. Height == {0}!
Explanation: An error occurred while running the substitution rule to change a template.
User response: Check the Height property for errors.

RPT30026:
The template is invalid. Width == {0}!
Explanation: An error occurred while running the substitution rule to change a template.
User response: Check the Width property for errors.

RPT30028:
Cannot load resources for package {0} and locale {1}.
Explanation: The bundle for the resource can not be found.
User response: Restart the application. If this does not fix the problem, reinstall the application.

RPT30029:
Request for resource string ":{0}" from package {1} failed.
Explanation: An error occurred while requesting a string resource.
User response: Restart the application. If this does not fix the problem, reinstall the application.

RPT30035:
No OutputComponentCreator in map for {0}.
Explanation: An internal error occurred.
User response: Collect support data by selecting Help → Collect Support Data and forward it to customer support.

RPT30038:
Could not find {0}.
Explanation: One of the objects in the Connections folder could not be found.
User response: Reload the project or delete the connection and try again.

RPT30040:
Chart cannot read query result set.
Explanation: An error occurred while creating a ClusterGraph, OrganizationChart, or TreeChart.
User response: Reconnect to the repository and reload the visual project.

RPT30042:
Error - axisLength is infinite. totalPercentages: {0} m_sceneRuntimePanel.getZoomFactor() {1}.
Explanation: The value of an axis's PercentageLength property is invalid.
User response: Check the PercentageLength property value of the axis for errors.

RPT30044:
Unknown checkbox value: {0}.
Explanation: A value other than UNCHECKED, CHECKED, or INDETERMINATE was specified.
User response: Ensure that any action that sets the value of a checkbox sets it to UNCHECKED, CHECKED, or INDETERMINATE.

RPT30045:
Unknown m_style: {0}.
Explanation: An invalid style property was specified for a ComboBox object.
User response: Check the style property for the ComboBox object for errors.

RPT30047:
Line is not horizontal or vertical.
Explanation: A segment in an ElbowConnector object is set to neither vertical or horizontal.
User response: Reload the visual project.

RPT30048:
Unsupported segment type: {0}.
Explanation: An unsupported segment type was encountered for an ElbowConnector object.
User response: Reload the visual project.

RPT30049:
Could not get corner type last: {0} start: {1} end: {2}.
Explanation: The start and end coordinates of a line segment do not define a corner in an ElbowConnector object.
User response: Reload the visual project, or delete and recreate the ElbowConnector object.

RPT30050:
Should not have gotten here.
Explanation: An internal error occurred.
User response: Collect support data by selecting Help → Collect Support Data and forward it to customer support.

RPT30052:
Unknown horizontal align: {0}.
Explanation: A value other than LEFT, RIGHT, or CENTER was encountered in the HorizontalAlign property of an object.
User response: Ensure that any action that sets the value of the HorizontalAlign property of an object sets it as LEFT, RIGHT, or CENTER.

RPT30053:
Unknown vertical align: {0}.
Explanation: A value other than TOP, BOTTOM, or CENTER was encountered in the VerticalAlign property of an object.
User response: Ensure that any action that sets the value of the VerticalAlign property of an object sets it as TOP, BOTTOM, or CENTER.

RPT30054:
Unknown location anchor: {0}.
Explanation: A value other than BOTTOMLEFT, LEFTCENTER, TOPLEFT, BASELINELEFT, BOTTOMCENTER, CENTER, TOPCENTER, BASELINECENTER, BOTTOMRIGHT, RIGHTCENTER, or BASELINERIGHT was encountered in the LocationAnchor property of an object.
User response: Ensure that any action that sets the value of the LocationAnchor property of an object sets it as BOTTOMLEFT, LEFTCENTER, TOPLEFT, BASELINELEFT, BOTTOMCENTER, CENTER, TOPCENTER, BASELINECENTER, BOTTOMRIGHT, RIGHTCENTER, or BASELINERIGHT.

RPT30055:
Unknown arrowheadStyle: {0}.
Explanation: A value other than NONE, TRIANGLE, CIRCLE, DIAMOND, or SWEPTPOINTER was encountered in the MarkerShape property of an arrow object.
User response: Ensure that any action that sets the value of the MarkerShape property of an arrow object sets it as NONE, TRIANGLE, CIRCLE, DIAMOND, or SWEPTPOINTER.
RPT30056:
Unknown marker shape: {0}.

Explanation: A value other than HOLLOWCIRCLE, HOLLOWDIAMOND, HOLLOW SQUARE, HOLLOWTRIANGLE, HOLLOWTRIANGLE2, HOLLOWSTAR, HOLLOWPLUS, SOLIDCIRCLE, SOLIDDIAMOND, SOLIDSQUARE, SOLIDTRIANGLE, SOLIDTRIANGLE2, SOLIDSTAR, SOLIDPLUS, CROSS, PLUS, BOXCROSS, or BOXPLUS was encountered in the MarkerShape property of a data point object.

User response: Ensure that any action that sets the value of the MarkerShape property of a data point object sets it as HOLLOWCIRCLE, HOLLOWDIAMOND, HOLLOW SQUARE, HOLLOWTRIANGLE, HOLLOWTRIANGLE2, HOLLOWSTAR, HOLLOWPLUS, SOLIDCIRCLE, SOLIDDIAMOND, SOLIDSQUARE, SOLIDTRIANGLE, SOLIDTRIANGLE2, SOLIDSTAR, SOLIDPLUS, CROSS, PLUS, BOXCROSS, or BOXPLUS.

RPT30060:
Unknown attachmentStyle: {0}.

Explanation: A value other than TOPLEFT, TOPCENTER, TOP, TOPRIGHT, CENTERLEFT, CENTER, CENTERRIGHT, BOTTOMLEFT, BOTTOMCENTER, BOTTOM, or BOTTOMRIGHT was encountered in the AttachmentStyle property of a connector object.

User response: Ensure that any action that sets the value of the AttachmentStyle property of a connector object sets it as TOPLEFT, TOPCENTER, TOP, TOPRIGHT, CENTERLEFT, CENTER, CENTERRIGHT, BOTTOMLEFT, BOTTOMCENTER, BOTTOM, or BOTTOMRIGHT.

RPT30061:
Unknown border type for table: {0}.

Explanation: An invalid value was encountered in the BorderType property of an object.

User response: Ensure that any action that sets the BorderType property of an object sets a valid value.

RPT30062:
Unknown m_horizontalAlign: {0}.

Explanation: An invalid value was encountered in the HorizontalAlign property of a table cell.

User response: Ensure that any action that sets the HorizontalAlign property of a table cell sets a valid value.

RPT30063:
Unknown m_verticalAlign: {0}.

Explanation: An invalid value was encountered in the VerticalAlign property of a table cell.

User response: Ensure that any action that sets the VerticalAlign property of a table cell sets a valid value.

RPT30064:
Unknown border type for table: {0}.

Explanation: An invalid value was encountered in the Border property of a table.

User response: Ensure that any action that sets the Border property of a table sets a valid value.

RPT30065:
Unknown scrollBarStyle: {0}.

Explanation: An invalid value was encountered in the ScrollBarStyle property of an object.

User response: Ensure that any action that sets the ScrollBarStyle property of an object sets a valid value.

RPT30066:
Unknown AlignmentStyle: {0}.

Explanation: An invalid value was encountered in the AlignmentStyle property of a textbox object.

User response: Ensure that any action that sets the AlignmentStyle property of a textbox object sets a valid value.
Appendix B: Messages

RPT30068:
Unknown lineStylePattern: {0}.
Explanation: An invalid value was encountered in the LineStylePattern property of an object.
User response: Ensure that any action that sets the LineStylePattern property of an object sets a valid value.

RPT30069:
Unknown gradient type: {0}.
Explanation: An invalid value was encountered in the GradientType property of an object.
User response: Ensure that any action that sets the GradientType property of an object sets a valid value.

RPT30070:
Unknown borderType: {0}.
Explanation: An invalid value was encountered in the BorderType property of a textbox or embedded scene object.
User response: Ensure that any action that sets the BorderType property of a textbox or embedded scene object sets a valid value.

RPT30071:
Unknown imageAnchor: {0}.
Explanation: An invalid value was encountered in the ImageAnchor property of an object.
User response: Ensure that any action that sets the ImageAnchor property of an object sets a valid value.

RPT30072:
Embedded Scene: Unknown scrollBarStyle.
Explanation: An invalid value was encountered in the ScrollBarStyle property of an embedded scene object.
User response: Ensure that any action that sets the ScrollBarStyle property of an embedded scene object sets a valid value.

RPT30073:
Embedded Scene: unknown typeOfFind: {0}.
Explanation: An internal error occurred.
User response: Collect support data by selecting Help → Collect Support Data and forward it to customer support.

RPT30076:
The cell chosen by client not found.
Explanation: The action can not be performed when using the HTML rendering mode.
User response: Reload the visual project.

RPT30077:
Grid cell's click event could not be dispatched.
Explanation: The action can not be performed when using the HTML rendering mode.
User response: Reload the visual project.

RPT30079:
Unknown page type - {0}.
Explanation: An internal error occurred.
User response: Collect support data by selecting Help → Collect Support Data and forward it to customer service.

RPT30082:
Parameters group cloning problem {0}.
Explanation: An internal error occurred.
Messages

User response: Collect support data by selecting Help → Collect Support Data and forward it to customer support.

RPT30083:
No class in map for {0}.
Explanation: An internal error occurred.
User response: Collect support data by selecting Help → Collect Support Data and forward it to customer support.

RPT30088:
Unable to serialize an object: {0}.
Explanation: An internal error occurred.
User response: Collect support data by selecting Help → Collect Support Data and forward it to customer support.

RPT30089:
Value must be positive: {0}.
Explanation: The indicated value must be a positive number.
User response: Ensure that the indicated value is a positive number.

RPT30090:
Level was null when zoom was {0}.
Explanation: No level is specified for the indicated zoom percentage.
User response: Ensure that a level is specified for the indicated zoom percentage.

RPT30091:
Chart can't find parent column: {0}.
Explanation: The indicated parent column could not be found in the result set.
User response: Ensure that a valid column named is specified in the DataLocator property of the group.

RPT30092:
Chart can't find child column: {0}.
Explanation: The indicated child column could not be found in the result set.
User response: Ensure that a valid column named is specified in the DataLocator property of the group.

RPT30095:
Hierarchy chart must have one DataTemplate child.
Explanation: A hierarchy chart must have one DataTemplate child.
User response: Check the Project Explorer and ensure that the hierarchy chart only has one DataTemplate child.

RPT30096:
CurrentRowIndex < 0.
Explanation: The specified row index is less than zero. The row index must be an integer that is equal to or greater than zero.
User response: Ensure the specified row index is an integer that is equal to or greater than zero.

RPT30097:
Unknown interval: {0}.
Explanation: An invalid value was specified for the DateAxis property of an object.
User response: Ensure that any action that sets the value of the DateAxis property of an object sets a valid value.

RPT30098:
Unhandled interval: {0}.
Appendix B: Messages

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.

RSE01000:
{0}
Appendix B: Messages

**Explanation:** The indicated error occurred while accessing the repository security provider.

**User response:** Resolve the indicated error and try again.

---

**RSE04000:**
*System internal error occurred, please check the error log.*

**Explanation:** A system internal error occurred.

**User response:** Check the error log by selecting **Output** from the context menu of the **Window** menu.

---

**RSE04001:**
*Cannot find User {0}*

**Explanation:** The indicated user can not be found.

**User response:** Check the name of the user for errors and try again.

---

**RSE04002:**
*User {0} already exists;*

**Explanation:** The indicated user already exists.

**User response:** Specify a different user.

---

**RSE04003:**
*Group {0} already exists;*

**Explanation:** The indicated group already exists.

**User response:** Specify a different group.

---

**RSE04004:**
*Cannot find group {0}*

**Explanation:** The indicated group can not be found.

**User response:** Specify a different group.

---

**RSE04005:**
*Not logged in with security provider.*

**Explanation:** You attempted to access the repository without logging in.

**User response:** Login to the repository and try again.

---

**RSE04006:**
*The user ID and password are invalid*

**Explanation:** The user ID and password that were entered are invalid.

**User response:** Check the user ID and password for errors and try again.

---

**RSE04007:**
*Name "{0}" is in conflict with the existing name.*

**Explanation:** The new name that was entered already exists.

**User response:** Select a different name.

---

**RSE04008:**
*Object access denied*

**Explanation:** Access to the specified object is denied.

**User response:** Select a different object to access or contact your administrator for more information.

---

**RSE04011:**
*Cannot remove User {0} from Group {1}*
Explanation: The indicated user can not be removed from the indicated group.
User response: Select another user to be removed from the group.

RSE04012:
Cannot delete User {0}
Explanation: The indicated user can not be deleted.
User response: Select another user to be deleted.

RSE04013:
Cannot delete Group {0}
Explanation: The indicated group can not be deleted.
User response: Select another group to be deleted.

RSE04014:
Cannot initialize security provider,{0}
Explanation: The indicated error occurred while initializing the repository.
User response: Resolve the indicated error and try again.

RSE04015:
This security provider is a read-only one.
Explanation: You can not change the principals of a read-only security provider.
User response: Choose another security provider to modify.

RSE04016:
Cannot find any name associates with UID {0} in Repository
Explanation: The indicated repository identifier is invalid.
User response: Specify a valid repository identifier and try again.

RSE04017:
AuthType is not anonymous("none"),please specify principal and credentials
Explanation: An error occurred while initializing the LDAP security provider.
User response: Change the repository initialization parameters and try again.

RSE04018:
LDAP Configuration contains illegal values or some required values are missing, please check the preferences({0})
Explanation: An error occurred while initializing the LDAP security provider.
User response: Change the repository initialization parameters and try again.

RSE04019:
Cannot connect to the LDAP server. Communication broken or Directory Service is not available.
Explanation: An error occurred while connecting to the LDAP directory.
User response: Check the network connection and try again.

RSE04020:
Cannot read User's password attribute (Make sure you have enough permission or PasswordAttribute in configuration is correct.)
Explanation: An error occurred while reading data from the LDAP directory.
User response: Ensure that the LDAP service is accessible.
<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
<th>Explanation</th>
<th>User response</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSE04021</td>
<td>Invalid search filter in LDAP configuration</td>
<td>The specified search filter in the LDAP configuration is invalid.</td>
<td>Specify a valid search filter for the LDAP configuration.</td>
</tr>
<tr>
<td>RSE04022</td>
<td>LDAP server returned error</td>
<td>An error occurred while reading data from the LDAP directory.</td>
<td>Ensure that the LDAP service is accessible.</td>
</tr>
<tr>
<td>RSE04023</td>
<td>Cannot find the specified name {0}</td>
<td>The indicated user name can not be found.</td>
<td>Specify a different user name.</td>
</tr>
<tr>
<td>RSE04024</td>
<td>Not enough permission to perform the LDAP operations.</td>
<td>You user ID does not have permission to perform LDAP operations.</td>
<td>Contact your administrator for more information concerning specific permissions and allowances.</td>
</tr>
<tr>
<td>RSE04025</td>
<td>Cannot find a attribute of entry {0} specified in LDAP configurations.</td>
<td>An error occurred while reading attribute data from the LDAP configuration.</td>
<td>Correct the configuration parameters and try again.</td>
</tr>
<tr>
<td>RSE04026</td>
<td>UUID {0} mapping to an invalid distinguished name {1}.({2})</td>
<td>The indicated distinguished name is invalid.</td>
<td>Specify a valid distinguished name and try again.</td>
</tr>
<tr>
<td>RSE04028</td>
<td>Incorrect operation.</td>
<td>An error occurred while editing the principals of the security provider.</td>
<td>Collect support data by selecting Help → Collect Support Data and forward it to customer support.</td>
</tr>
<tr>
<td>RSE04029</td>
<td>Cannot register security provider.</td>
<td>An error occurred while registering the security provider in the provider registry.</td>
<td>Ensure that the registered name of the provider is unique.</td>
</tr>
<tr>
<td>RSE04030</td>
<td>Security provider not initialized.</td>
<td>An error occurred while accessing the security provider service because the service is not initialized.</td>
<td>Initialize the security provider and try again.</td>
</tr>
<tr>
<td>RSE04031</td>
<td>Security provider not registered.</td>
<td>An error occurred while accessing the security provider because the security provider is not registered.</td>
<td>Ensure that the registered name of the provider is unique.</td>
</tr>
</tbody>
</table>
User response: Register the security provider and try again.

RSE04032:  
You can not remove default Administrator.
Explanation: The default administrator can not be deleted.
User response: Specify a different administrator to be deleted.

RSE04033:  
You can not change the default Administrator's membership.
Explanation: The membership of the default administrator can not be changed.
User response: Specify the membership of a different administrator to change.

RSE04035:  
You can not rename the Administrators group.
Explanation: The group of the default administrator can not be renamed.
User response: Specify the group of a different administrator to be renamed.

RSE04036:  
Password has expired and is no longer valid.
Explanation: The specified password has expired and is no longer valid.
User response: Specify an updated password and try again.

RSE04037:  
Group could not became its own parent.
Explanation: A group may not be its own parent. You must select a different group as the parent.
User response: Select a different group to be the parent of the specified group.

SPL01001:  
Cannot find objectClass attribute of {0}
Explanation: The indicated object does not have an objectClass attribute. Every object in the LDAP directory must contain this attribute.
User response: Specify an objectClass attribute for the indicated object.

SPL01002:  
The returned entry {0} is not acceptable for LDAP Security Provider, its objectClass is not {1} or {2}
Explanation: The indicated object is defined as neither a user or group in the LDAP security provider.
User response: Edit the objectClass attribute of the indicated object to specify whether it is a user or group.

SPL01003:  
Cannot read the Group's name (attribute: {0})
Explanation: The LDAP directory does not contain an object that has the required attribute.
User response: Edit the groupName attribute so that it will point to an existing object in the LDAP directory.

SPL01004:  
Cannot read the User's name (attribute: {0})
Explanation: The LDAP directory does not contain an object that has the required attribute.
User response: Edit the userName attribute so that it will point to an existing object in the LDAP directory.

SPL01005:  
LDAP server host or port was not specified.
**Explanation:** The LDAP server host or port was not specified.

**User response:** Specify the LDAP server and host.

---

**SPL01006:**

**The base distinguish name of the LDAP server must be present.**

**Explanation:** The base distinguished name of the LDAP server was not specified.

**User response:** Specify the base distinguished name of the LDAP server.

---

**SPL01007:**

**The distinguish name of user principal must be present.**

**Explanation:** The distinguished name of the principal user was not specified.

**User response:** Specify the distinguished name of the principal user or use anonymous authentication.

---

**SPL01008:**

**The credentials of user entry must be specified.**

**Explanation:** The credentials of the entry user were not specified.

**User response:** Specify the credentials of the entry user or use anonymous authentication.

---

**SPL01009:**

**Cannot create LDAP entry for administrator user: {0}**

**Explanation:** The user who is associated with the administrative account could not be found in the LDAP directory.

**User response:** Specify the administrative user of the LDAP directory.

---

**SPL01010:**

**Cannot create LDAP entry for administrators group: {0}**

**Explanation:** The group that is associated with the administrative account could not be found in the LDAP directory.

**User response:** Specify the administrative group of the LDAP directory.

---

**SPL01011:**

**Cannot create LDAP entry for everyone group: {0}**

**Explanation:** The group that is associated with the everyone account could not be found in the LDAP directory.

**User response:** Specify the everyone group of the LDAP directory.

---

**SPL01012:**

**Cannot create LDAP entry for root group: {0}**

**Explanation:** The object that is associated with the Base DN could not be found in the LDAP directory.

**User response:** Specify the Base DN object in the LDAP directory.

---

**SPL01013:**

**Cannot find the JNDI extension package or Sun's LDAP Service Provider.**

**Explanation:** The JNDI extension package or Sun's LDAP Service Provider was not located in the CLASSPATH.

**User response:** Ensure that the CLASSPATH includes both the JNDI extension package and Sun's LDAP Service Provider.

---

**SPL01014:**

**The catalog object is missing.**

**Explanation:** The catalog object is not set.

**User response:** Specify the catalog object and try again.
SPL01015:
Empty name is not allowed.
Explanation: No name was specified as an RDN.
User response: Specify a name as an RDN.

SPL01016:
Illegal RDN of security object (offending string: "{0}")
Explanation: The RDN is invalid. The RDN must be one or more LDAP components that are separated by commas.
User response: Specify valid components for the RDN.

SPL01017:
Unknown attribute name "{0}" in RDN.
Explanation: An unknown attribute name was encountered while parsing the RDN.
User response: Check the attributes of the RDN for errors.

SPL01101:
Currently Client Authentication is not supported.
Explanation: Client Authentication is not supported.
User response: Use anonymous authentication.

SPL01103:
SSL is not correctly initialized.
Explanation: The secure socket protocol factory is not initialized.
User response: Restart the application and try again.

SPL01104:
Attribute {0} is absent.
Explanation: The indicated attribute was not found while reading the LDAP object.
User response: Specify a valid value for the indicated attribute.

SQM02001:
An input/output error occurred while processing package "{0}". {1}
Explanation: The indicated error occurred while processing the indicated package.
User response: Resolve the indicated error and try again.

SQM02002:
Error in SQLJ customization: {0}
Explanation: The indicated error occurred while processing the SQLJ customization.
User response: Resolve the indicated error and try again.

SQM02003:
No SQLJ installed
Explanation: The SQLJ library could not be loaded.
User response: Add sqlj.zip to the driver definition.

SQM02004:
An error occurred while retrieving profile data for package "{0}". {1}
Explanation: The indicated error occurred while the profile data for the indicated package was being loaded.
User response: Resolve the indicated error and try again.
Appendix B: Messages

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 the class with the implementation of the indicated package was being loaded.
User response: Reinstall the application and try again.

SQM02007:
A database access error occurred while processing package "{0}". {1}
Explanation: The indicated database access error occurred while the package was being processed.
User response: Resolve the indicated database access error and try again.

SQM02008:
A database access error occurred: {1}
Explanation: The indicated database access error occurred.
User response: Resolve the indicated error and try again.

SQM02009:
User has no rights to execute queries of package "{0}". Original error follows: {1}
Explanation: The user does not have permission to access the package because of the indicated reason.
User response: Contact your database administrator.

SQM02010:
An error occurred while deleting profile for package "{0}". {1}
Explanation: The indicated error occurred while the profile for the indicated package was being deleted.
User response: Resolve the indicated error and try again.

SQM02011:
Empty customizer call pattern
Explanation: Package customization failed because of an unexpected error.
User response: Collect support data by selecting Help → Collect Support Data and forward it to customer support.

SQM02013:
An error occurred while loading data for package "{0}".
Explanation: An error occurred while the data for the indicated package was being loaded.
User response: Resolve the indicated error and try again.

SQM02014:
Wrong customizer call pattern: {0}
Explanation: Package customization failed because of an unexpected error.
User response: Collect support data by selecting Help → Collect Support Data and forward it to customer support.

SQM02016:
Cannot create collection "{0}" due to the following: {1}
Explanation: The indicated collection could not be created due to the indicated error.
User response: Resolve the indicated error and try again.

SQM02017: Profile for package "\{0\}" is too long to be stored at database
Explanation: The profile for the indicated package is too long to be stored at the database.
User response: Collect support data by selecting Help ➔ Collect Support Data and forward it to customer support.

SQM02018: No collection ID (profile owner) is specified. Cannot proceed in static mode.
Explanation: The SQLJ package can not be loaded if no collection ID is specified.
User response: Collect support data by selecting Help ➔ Collect Support Data and forward it to customer support.

SQM02019: Cannot prepare statement "\{0\}" of package "\{1\}" due to the following error: "\{2\}"
Explanation: The indicated statement of the package can not be prepared due to the indicated error.
User response: Resolve the indicated error and try again.

SQM02020: While checking user rights on package "\{0\}" the following error has occurred: "\{1\}"
Explanation: The indicated error occurred while the user rights on the indicated package were checked.
User response: Resolve the indicated error and try again.

SQM02021: Inconsistent profile data for queries package "\{0\}" were detected. Rebind package.
Explanation: Inconsistent profile data for the indicated queries package was detected.
User response: Rebind the package.

SQM02100: The required parameter "\{0\}" is missing.
Explanation: The indicated required parameter is missing.
User response: Collect support data by selecting Help ➔ Collect Support Data and forward it to customer support.

SQM02101: An error occurred while initializing the connection. \{0\}
Explanation: The indicated error occurred while the connection was initialized.
User response: Resolve the indicated error and try again.

SQM02102: \{0\}
Explanation: The indicated database access error occurred.
User response: Resolve the indicated error and try again.

SQM02103: An error occurred while loading package description class "\{0\}". \{1\}
Explanation: The indicated error occurred while the package description class was loaded.
User response: Resolve the indicated error and try again.

SQM02104: An error occurred while loading the profile for package "\{0\}". \{1\}
Explanation: The indicated error occurred while the profile for the indicated package was loaded.
Appendix B: Messages

**User response:** Resolve the indicated error and try again.

**SQM02105:**
Static mode profile for package "{0}" is not found.
*Explanation:* The static mode for the indicated package was not found.
*User response:* Reinstall the application and try again.

**SQM02106:**
An error occurred while creating directory "{0}".
*Explanation:* The temporary directory for the customization process could not be created.
*User response:* Ensure that the application has permissions to create the directory and try again.

**SQM02107:**
An error occurred while creating profiles directory "{0}".
*Explanation:* An error occurred while the indicated profile's directory was created.
*User response:* Ensure that the application has permissions to create the directory and try again.

**SQM02108:**
An error occurred while saving the profile for package "{0}". {1}
*Explanation:* The indicated error occurred while the profile for the indicated package was saved.
*User response:* Resolve the error and try again.

**SQM02109:**
An error occurred while saving the packages description file. {0}
*Explanation:* The indicated error occurred while the package description file was saved.
*User response:* Resolve the error and try again.

**SQM02110:**
An error occurred while starting customization. {0}
*Explanation:* The indicated error occurred while customization was started.
*User response:* Resolve the error and try again.

**SQM02111:**
An error occurred while customizing package "{0}". {1}
*Explanation:* The indicated error occurred while the indicated package was customized.
*User response:* Resolve the indicated error and try again.

**SQM02112:**
An error occurred while creating the profiles data table. {0}
*Explanation:* The indicated error occurred while the data table of the profile was created.
*User response:* Resolve the indicated error and try again.

**SQM02113:**
No user ID list for GRANT/REVOKE operation was specified.
*Explanation:* No user ID list for GRANT/REVOKE operation was specified.
*User response:* Specify a user ID list for the GRANT/REVOKE operation.

**SQM02115:**
An error occurred while uninstalling the packages. {0}
*Explanation:* The indicated error occurred while the packages were uninstalled.
User response: Resolve the indicated error and try again.

SQM02116:  
File with packages description was not found. {0}  
Explanation: The indicated error occurred while preparing the packages descriptions based on the previously build packages description file.  
User response: Collect support data by selecting Help → Collect Support Data and forward it to customer support.

SQM02117:  
An error occurred while reading the packages description file. {0}  
Explanation: The indicated error occurred while the packages description file was being read.  
User response: Collect support data by selecting Help → Collect Support Data and forward it to customer support.

SQM02119:  
An error occurred while installing the package {0}. {1}  
Explanation: The indicated error occurred when the indicated package was being installed.  
User response: Resolve the indicated error and try again.

SQM02120:  
An error occurred while uninstalling the package {0}. {1}  
Explanation: The indicated error occurred when the indicated package was being uninstalled.  
User response: Resolve the indicated error and try again.

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.

SYNERRCD0009:
Object length greater than maximum allowed.
Explanation: An internal error in the driver code occurred.
User response: Send an error report to the developers.

SYNERRCD0010:
Object length less than minimum required.
Explanation: An internal error in the driver code occurred.
User response: Send an error report to the developers.

SYNERRCD0011:
Object length not allowed.
Explanation: An internal error in the driver code occurred.
User response: Send an error report to the developers.

SYNERRCD0012:
Incorrect large object extended length field.
Explanation: An internal error in the driver code occurred.
User response: Send an error report to the developers.

SYNERRCD0013:
Object codepoint index not supported.
Explanation: An internal error in the driver code occurred.
User response: Send an error report to the developers.

SYNERRCD0014:
Required object not found.
Explanation: An internal error in the driver code occurred.
User response: Send an error report to the developers.

SYNERRCD0015:
Too many command data objects sent.
Explanation: An internal error in the driver code occurred.
User response: Send an error report to the developers.

SYNERRCD0016:
Mutually-exclusive objects present.
Explanation: An internal error in the driver code occurred.
User response: Send an error report to the developers.

SYNERRCD0017:
Too few command data objects sent.
Explanation: An internal error in the driver code occurred.
User response: Send an error report to the developers.

SYNERRCD0018:
Duplicate object present.
Explanation: An internal error in the driver code occurred.
User response: Send an error report to the developers.

SYNERRCD0019:
Invalid request correlator specified.
Explanation: An internal error in the driver code occurred.
User response: Send an error report to the developers.

SYNERRCD0020:
Required value not found.
Explanation: An internal error in the driver code occurred.
User response: Send an error report to the developers.

SYNERRCD0021:
Reserved value not allowed.
Explanation: An internal error in the driver code occurred.
User response: Send an error report to the developers.

SYNERRCD0022:
DSS continuation less than or equal to two.
Explanation: An internal error in the driver code occurred.
User response: Send an error report to the developers.

SYNERRCD0023:
Objects not in required order.
Explanation: An internal error in the driver code occurred.
User response: Send an error report to the developers.

SYNERRCD0024:
DSS chaining bit not 1, but DSSFMT bit 3 set to 1.
Explanation: An internal error in the driver code occurred.
User response: Send an error report to the developers.

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.

THM1001:
Cannot copy CSS file.

Explanation: The CSS file could not be copied to the server.
User response: Ensure that the CSS file name is correct and try again.

THM1002:
Cannot extract main theme CSS file from the archive.

Explanation: The specified archive does not contain the main CSS file.
User response: Ensure that the main CSS file is saved to the archive and try again.

THM1003:
Already registered. Try another theme name.

Explanation: A theme with the same name already exists.
User response: Specify a different name for the theme.

THM1004:
Theme root folder already exists. Try another theme name.

Explanation: A theme with the same name already exists.
User response: Specify a different name for the theme.

UIM12001:
Internal error has occurred: {0}

Explanation: The indicated internal error occurred.
User response: Resolve the indicated error and try again.

UIM12004:
An input/output error occurred: {0}
Explanation: The indicated input/output error occurred.
User response: Resolve the indicated error and try again.

UIM12005:
The file does not exist: {0}
Explanation: The indicated file does not exist.
User response: Specify a valid file and try again.

VAL00001:
Variable {0} not found.
Explanation: The indicated variable was not found.
User response: Specify a valid variable.

VAL00003:
Type conversion error. Object of {0} type cannot be converted to {1} type.
Explanation: The indicated object can not be converted to the indicated type.
User response: Specify an object of a different type to be converted, or convert the object to a different type.

VAL00004:
Unit conversion error. Unit {0} cannot be converted to {1}.
Explanation: The indicated unit can not be converted to the indicated type.
User response: Specify a different unit to be converted, or convert the unit to a different type.

VAL00005:
An expression contains a syntax error.
Explanation: An expression that you entered contains a syntax error.
User response: Check the expressions that you typed for syntax errors.

VAL00006:
Wrong expression has been used.
Explanation: An incorrect expression has been used.
User response: Specify a valid expression.

VAL00007:
Name "{{0}}" has invalid character at {1}. The name must start with a letter and should not contain special characters. See the help for a list of invalid characters.
Explanation: The indicated name contains an invalid character at the indicated column.
User response: Check the help for syntax rules and retype the invalid character.

VAL00008:
Name cannot be empty. The name must start with a letter and should not contain special characters. See the help for a list of invalid characters.
Explanation: The name of a control can not be empty.
User response: Specify a correct name for the control.

VAL00009:
Unable to process URL: "{{0}}"
Appendix B: Messages

**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 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.
Appendix B: Messages

**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.

**VDS1017:**
**Virtual table cannot be created because source query returns multiple columns that have the same name.**

**Explanation:** The source table for a virtual table must not return columns that have the same name.

**User response:** Specify a source table that does not return columns that have the same name.
Appendix C: 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. (CorVu Workstation Pro 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.

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.

complex queries

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

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.
Appendix C: Glossary

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 CorVu Web Pro.

Display editor

The Display editor allows you to create dynamic visual representations of your query results in the form of easy to manage charts and graphs.

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 CorVu interface and the database. The CorVu 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.

forecasts

Forecasts allow you to predict future values of historical time series data using existing query results. There are several forecast models that are available to find the best predictive formula.

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.

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.
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

CorVu uses JDBC to connect to all database repositories and data sources. CorVu does not include JDBC drivers. The location of the JDBC drivers that CorVu will use to connect to repositories and data sources must be defined. CorVu administrators are responsible for identifying where each instance of the application interface will find the JDBC drivers.

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.

LOB data

A large object (LOB) is a DB2 for Linux, UNIX®, and Windows data type that houses nontraditional data such as text, multimedia, image, video, photograph, sound, or any very large data file inside a database table. Retrieving or saving LOB data can consume a substantial amount of resources.
network repositories

A connection to a shared repository using direct network access. Network connectivity permits users to access a shared repository without the need for direct database connectivity from the workstation. Connecting to repositories using a network connection allows for consolidation of data access at a central server and removes the need to distribute JDBC drivers across each desktop running CorVu Workstation Pro. This type of connection is not applicable to CorVu Web Pro users.

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 CorVu application interface.

perspectives

An interface used for querying data and formatting results.

Project Explorer view

The Project Explorer view details in a tree structural all the elements and objects of each visual report and each dashboard.

prompt hierarchies

Prompt hierarchies allow the end user to select one or more values from a flat or hierarchical list of options. They are stored independent of a given query, allowing them to be used by multiple query objects.

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.

quick reports

Quick reports allow you to rapidly generate a printable report from a query result set.
regular save

When saving your query results data using a "regular save" method, the interface that you are using (CorVu Workstation Pro or CorVu Web Pro) 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

CorVu 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.

scenarios

Scenarios are interactive series of charts that allows you to display several levels of details of a given query.

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.

shared repositories

Shared repositories are located on a database and can be shared by many users. Only CorVu 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 CorVu to store processing information. Some of these tables store sensitive information, for example the permissions table. By default all users can access these repository storage tables and make changes to the tables. You can choose to protect the repository storage tables. In protection mode, the repository storage tables are accessed using a collection of stored procedures or static SQL packages depending on what the database that...
is hosting the repository storage supports. Users of the repository storage must then be granted permission to run the stored procedures or static SQL packages.

substitution variables

Substitution variables are used to 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 query is running. As a result, only one object can access the substitution variable. The variable will not exist after the object is executed.

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 CorVu application interface session. The Preferences dialog consists of two panes. The left pane displays a hierarchal 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

CorVu 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 CorVu 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 CorVu Workstation Pro, CorVu Web Pro, 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 CorVu Web Pro 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 CorVu Workstation Pro. This type of connection does not apply to CorVu Web Pro.

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 CorVu 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 CorVu Workstation Pro, you can use the Zoom option to view in hexadecimal or binary the contents of a cell in the query results.
about repositories......................................................... 250,349,350
accessibility
  assistive technology................................................... 399
  keyboard equivalents.................................................. 398
  keyboard navigation.................................................. 398
  navigation in WebSphere............................................ 401
  navigation in Workstation.......................................... 399
  operating system..................................................... 398
accessing different components.................................... 253
ACTION........................................................................... 367,375
actions
  action groups................................................................. 177,178,179,179
  execute object
    export query................................................................. 166
    export query to mail..................................................... 167
    export quick report...................................................... 170
    export quick report to mail.......................................... 169
    export visual portfolio.................................................. 174
    export visual portfolio to mail........................................ 173
    export visual report...................................................... 172,174
    export visual report to mail.......................................... 171
    print query................................................................... 169
  resource manipulation
    copy........................................................................... 176
    send mail.................................................................... 177
Ad hoc reports
  creating....................................................................... 194
Add to Favorites.............................................................. 31
Add to Startup..................................................................... 31
adding a Print scene event action
  how to........................................................................... 234
adding a Reload scene event action
  how to........................................................................... 240
adding a Set focus event action
  how to........................................................................... 240
adding a Show message event action
  how to........................................................................... 240
adding an End session event action
  how to........................................................................... 238
adding an Execute JavaScript event
  how to........................................................................... 241
adding an Execute procedure event
  how to........................................................................... 237
adding an Execute SQL statement event
  how to........................................................................... 236
adding an Export to Excel event action
  how to........................................................................... 239
adding an Export to PDF event action
  how to........................................................................... 241
adding objects................................................................... 251,251,252
Adding objects
  Canvas
  Flash mode.................................................................... 208
  adding Send Email event action
    how to........................................................................... 234
  Administrator perspective........................................... 16,16,19
  AeroText knowledge base........................................... 352,353,354
  aggregations................................................................. 128
  analysis paths............................................................... 146
Analytical queries
  Analytical queries....................................................... 80
  append queries.............................................................. 78
  conditional grouping.................................................... 83
  crosstab queries............................................................ 79
  join queries................................................................. 79
  normalize..................................................................... 80
  Appearance preferences.............................................. 22
  append queries.............................................................. 78
  applications.................................................................. 201
  applying
    conditional format.................................................... 136,142
  conditional formatting............................................... 136,142
  associations.................................................................. 68,68,70
  ATTACHMENT................................................................ 368
  AutoFit........................................................................... 110
  Average.......................................................................... 110
  axes chart format options........................................... 119
B
  best fit models............................................................. 152
  binary.......................................................................... 109
  bind() method............................................................... 255
  binding resource limits............................................... 392
  BIRT reports................................................................... 19
  BODY.............................................................................. 368
  Bookmarks view............................................................ 16,19
  Boolean.......................................................................... 220
  Border........................................................................... 225
  BOTTOM......................................................................... 359
  breakpoints................................................................... 320
C
  cache resource limits.................................................. 396
  caching
    repository data........................................................... 51
  calculated columns..................................................... 91,92,93,247
  calculated expressions................................................. 271
  Calculator viewCorVu................................................... 16,19
  Cancel........................................................................... 63
  canvas
    editing linked objects................................................... 213
    linked objects............................................................. 213
    placing......................................................................... 206
    sizing.......................................................................... 206
  Canvas
    Built-in.......................................................................... 200,200
    layouts......................................................................... 200
<table>
<thead>
<tr>
<th>Function/Operation</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Export quick report operation</td>
<td>170</td>
</tr>
<tr>
<td>Export quick report to email operation</td>
<td>169</td>
</tr>
<tr>
<td>Export to excel</td>
<td>239</td>
</tr>
<tr>
<td>Export to Excel</td>
<td>228</td>
</tr>
<tr>
<td>Export to PDF</td>
<td>241</td>
</tr>
<tr>
<td>Export visual portfolio operation</td>
<td>174</td>
</tr>
<tr>
<td>Export visual portfolio to mail operation</td>
<td>173</td>
</tr>
<tr>
<td>Export visual report operation</td>
<td>172</td>
</tr>
<tr>
<td>Export visual report to mail operation</td>
<td>171</td>
</tr>
<tr>
<td>Expression design</td>
<td>112</td>
</tr>
<tr>
<td>Events</td>
<td></td>
</tr>
<tr>
<td>Expiration schedules</td>
<td></td>
</tr>
<tr>
<td>daily</td>
<td>327</td>
</tr>
<tr>
<td>hourly</td>
<td>327</td>
</tr>
<tr>
<td>monthly</td>
<td>328</td>
</tr>
<tr>
<td>specific dates</td>
<td>328</td>
</tr>
<tr>
<td>weekly</td>
<td>327</td>
</tr>
<tr>
<td>External Event events</td>
<td>231</td>
</tr>
<tr>
<td>Events view</td>
<td></td>
</tr>
<tr>
<td>Execute</td>
<td></td>
</tr>
<tr>
<td>CommandLine</td>
<td>363</td>
</tr>
<tr>
<td>Execute JavaScript</td>
<td>237</td>
</tr>
<tr>
<td>Execute procedure</td>
<td>236</td>
</tr>
<tr>
<td>Execute shell command</td>
<td>228</td>
</tr>
<tr>
<td>Execute SQL statement</td>
<td>228</td>
</tr>
<tr>
<td>Execute SQL statement</td>
<td>227</td>
</tr>
<tr>
<td>forecast trends</td>
<td></td>
</tr>
<tr>
<td>forecast chart properties</td>
<td>163</td>
</tr>
<tr>
<td>forecast cyclerity</td>
<td>162</td>
</tr>
<tr>
<td>forecast grid properties</td>
<td>163</td>
</tr>
<tr>
<td>Forecast performance</td>
<td></td>
</tr>
<tr>
<td>cumulative forecast error</td>
<td>161</td>
</tr>
<tr>
<td>mean absolute deviation</td>
<td>161</td>
</tr>
<tr>
<td>mean absolute percent error</td>
<td>161</td>
</tr>
<tr>
<td>mean square error</td>
<td>161</td>
</tr>
<tr>
<td>tracking signal</td>
<td>161</td>
</tr>
<tr>
<td>forecast seasonality</td>
<td></td>
</tr>
<tr>
<td>global average</td>
<td>162</td>
</tr>
<tr>
<td>moving average</td>
<td>162</td>
</tr>
<tr>
<td>forecasting models options</td>
<td></td>
</tr>
<tr>
<td>Curve fitting</td>
<td>159</td>
</tr>
<tr>
<td>Double exponential smoothing</td>
<td>158</td>
</tr>
<tr>
<td>Holt-Winters method</td>
<td>158</td>
</tr>
<tr>
<td>Multiplicative decomposition</td>
<td>159</td>
</tr>
<tr>
<td>Neural network</td>
<td>156</td>
</tr>
</tbody>
</table>
Index

HTML5.................................................................................................. 265
I
icons
Create Cyclicality........................................................................ 152
Create Performance...................................................................... 152
Create Seasonality...................................................................... 152
Create Trend.............................................................................. 152
Run Query................................................................................ 152
Show Forecast Outline Tree..................................................... 152
IDENTIFIER.................................................................................. 361
images.......................................................................................... 318,336
import CSV
result set structure................................................................. 106
separators options................................................................ 105
import DBF.............................................................................. 104
import IXF.............................................................................. 104
import TXT
result set structure................................................................. 106
separators options................................................................ 105
import XML............................................................................... 104
IMPORT
ACTION...................................................................................... 367
COMMENT................................................................................. 367
CONFIRM.................................................................................... 367
FileName................................................................................... 367
LANGUAGE................................................................................ 367
LOBSFROM............................................................................. 367
ObjectName............................................................................. 367
SHARE...................................................................................... 367
import CSV
result set structure................................................................. 106
separators options................................................................ 105
import DBF.............................................................................. 104
import IXF.............................................................................. 104
import TXT
result set structure................................................................. 106
separators options................................................................ 105
import XML............................................................................... 104
information zooming
layout objects........................................................................ 222
scenes..................................................................................... 223
Informix.......................................................................................... 15
inner join.................................................................................... 62
Insert Query window.............................................................. 215,217
Integer.......................................................................................... 220
introduction................................................................................ 349
Invalidated Query Cache.......................................................... 244
iterator........................................................................................ 179
J
JAR........................................................................................................ 25,42
JavaScript
event actions........................................................................ 257
functions..................................................................................... 263
module......................................................................................... 263
JavaScript functions.................................................................... 93
JavaScript modules....................................................................... 318
JavaScript Tables....................................................................... 341,341,345
JDBC................................................................................................. 15
JDBC drivers.............................................................................. 15,41
JDBC libraries............................................................................ 25,42
JDBC library settings file.......................................................... 41
JDBC preferences...................................................................... 25,42
join queries............................................................................. 79
Jump to new location.............................................................. 228,229,229,276
K
knowledge base........................................................................... 350,350,350
knowledge base dictionaries................................................... 351,351
L
labels chart format options......................................................... 121
LANGUAGE................................................................................ 367,367,375
Last............................................................................................. 110
Launch LOB............................................................................. 245
Layers view............................................................................... 16,19,184,317
layout objects
Bar chart................................................................................... 290
BarChart
extended............................................................................. 294
Candlestick chart...................................................................... 290
CandlestickChart....................................................................... 298
Cluster chart........................................................................... 290
ClusterGraph............................................................................. 299
Column chart........................................................................... 290
ColumnChart
extended............................................................................. 294
data symbols
graphic objects........................................................................ 315
data templates
new........................................................................................... 316
default connectors..................................................................... 315
Dial.............................................................................................. 305
Event Band chart...................................................................... 290
EventBandChart....................................................................... 299
extending functionality............................................................ 314
Grid.............................................................................................. 290,293
Horizon...................................................................................... 290
information zooming.............................................................. 222
inserting..................................................................................... 291
Linear map............................................................................... 290
LinearMaps............................................................................... 299
Matrix.......................................................................................... 290,301
multiple query results............................................................. 315
Multivariate chart...................................................................... 290
MultivariateChart...................................................................... 296
nesting......................................................................................... 317
Organization chart.................................................................... 290
OrganizationChart..................................................................... 301
Pie chart..................................................................................... 290
PieChart
extended.................................................................................. 295
queries......................................................................................... 314
query parameters....................................................................... 314
Scatter chart............................................................................. 290
ScatterChart............................................................................... 296
Simple form............................................................................... 290
SimpleForm............................................................................... 300
Q
queries creating.................................................................57
OLAP
building offline..........................................................65
building online..........................................................66
filtering cube data.......................................................67
modifying.................................................................65
opening...........................................................................65
retrieving cube data....................................................64
profiling...........................................................................89
relational.........................................................................57
-specifying in visual applications.................................259
-specifying in visual dashboards......................................259
-specifying in visual reports...........................................259
query connections adding................................................262
editing.............................................................................262
Query Editor..................................................................37
query environment.........................................................387
Query menu commands
Cancel............................................................................63
Manage Prompts.........................................................63
Prepare............................................................................63
Run..................................................................................63
Set Data Source..........................................................63
Set Font............................................................................63
Set Row Limit..................................................................63
Set User Information....................................................63
Transfer To......................................................................63
query parameters............................................................922
query results created columns........................................91,92
-contents of a cell.........................................................109
-customizing display.......................................................95
-exporting.................................................................99
-exporting to a database...............................................99
-exporting to a file.........................................................100
-file export options.......................................................102
-filtering
-compound condition.....................................................113
-free style condition.......................................................114
-grouping and aggregation.............................................94
-LOB data.......................................................................108
-sorting..........................................................................112
Query Structure tree.......................................................64,65
quick reports changing data source.................................143
column headers.............................................................134
create............................................................................135
creating........................................................................131
creating column groups...............................................134
data format options.......................................................141
detail columns...............................................................132
details columns formatting..........................................140
displaying details columns............................................132
displaying grouping columns........................................132
editing............................................................................143
editing data source.......................................................144
grouping columns........................................................138
grouping columns footer...............................................138

R
range chart format options..........................................122
Recently Used folder.....................................................36
retrieving cube data......................................................64
profiling...........................................................................89
relational queries...........................................................57
Relate scene....................................................................240
remote scheduled tasks...............................................183
remote server..................................................................24
Remove from Favorites.................................................36
Remove from Startup.....................................................36
Rename............................................................................31
Report Designer perspective.........................................19
reports
visual.............................................................................184
repositories
connections
import............................................................................43
creating connections
-file-based.................................................................46
-network........................................................................47
-shared.........................................................................45
-web service...............................................................48
creating new objects.....................................................55
creating web links
data objects...........................................................49
-managing connections...............................................48
-personal........................................................................39
-personal repositories..................................................39
repository connection settings......................................15
repository connections...............................................40,43
Repository Connections view........................................16,19
repository objects
Analytical queries.........................................................55
complex prompts.........................................................55
Drill-down Paths..........................................................55
ER diagrams...............................................................55
folders............................................................................55
forecasts.........................................................................55
links...............................................................................55
personal repositories....................................................55
 queries.........................................................................55
quick reports...............................................................55
repository connections...............................................55
visual reports...............................................................55
workspaces....................................................................55
repository objects caching..........................................51
requirements..................................................................351,351
### Index

<table>
<thead>
<tr>
<th>PageFooter</th>
<th>190</th>
</tr>
</thead>
<tbody>
<tr>
<td>PageHeading</td>
<td>190</td>
</tr>
</tbody>
</table>

**W**

- web service repositories................................. 44,48
- weighted moving average.................................. 157
- Weighted Moving Average.................................. 152
- WIDTH...................................................................... 364

**workspace properties**

- Connection Parameters................................. 34
- LDAP...................................................................... 34
- Login mapping................................................... 34
- Plug-ins............................................................... 34
- Resource Limits.................................................. 34
- Security................................................................. 34

**workspaces**

- adding a new workspace.................................... 36
- adding content................................................... 35
- creating................................................................. 36
- discovering contents.......................................... 30
- filtering data source objects................................ 33
- folders..................................................................... 32
- linking to objects
  - repository......................................................... 33
  - workspace.......................................................... 33
- moving objects.................................................... 35
- properties............................................................... 34
- refreshing contents............................................. 34

- Workspaces view................................................. 16,19,30

**Workspaces view menu commands**

- Add to Favorites................................................ 31
- Add to Startup .................................................... 31
- Copy......................................................................... 31
- Delete........................................................................ 31
- Explore....................................................................... 31
- New.......................................................................... 31
- Open.......................................................................... 31
- Open From............................................................. 31
- Paste........................................................................ 31
- Paste Link............................................................... 31
- Properties............................................................... 31
- Refresh...................................................................... 31
- Rename...................................................................... 31
- Run........................................................................... 31
- Set Name Filter...................................................... 31
- Set User Information.............................................. 31
- Show in Repository................................................ 31

**X**

- X-Axis chart format options.............................. 120
- Xlocation............................................................... 224
- XMLA connectivity.................................................. 15

**Y**

- Y-Axis chart format options.............................. 120
- YLocation............................................................... 224
Z

zoom............................................................ 109
ZoomPct......................................................... 225