## Contents

**Documentation Accessibility** .......................................................... 17

**Chapter 1. Introducing Interactive Reporting** ........................................ 19
  Overview .................................................................................. 19
  Interactive Reporting Features .................................................. 19
  How Interactive Reporting Works .............................................. 20
  Relational Databases .................................................................. 20
  Multidimensional Databases .................................................... 20
  About Interactive Reporting Documents ..................................... 20
  Data Source Connections ........................................................... 21
  Data Models ............................................................................. 22
  Queries ..................................................................................... 22
  Analysis and Reporting .............................................................. 23
  Turning Data into Information .................................................... 24

**Chapter 2. Interactive Reporting: A Tutorial** ........................................ 25
  Starting Interactive Reporting .................................................... 25
  Opening and Saving a Sample File .............................................. 25
  Looking at a Simple Query ........................................................ 26
    Viewing Database Tables ....................................................... 26
    Adding Topics to a Query ....................................................... 27
  Viewing Results ......................................................................... 27
  Reordering Columns ................................................................. 27
  Sorting Columns ...................................................................... 27
  Filtering Data ............................................................................ 28
  Calculating Data ....................................................................... 28
  Pivoting Data ............................................................................ 28
  Calculating Totals and Subtotals ............................................... 29
  Drilling Down ........................................................................... 29
  Hiding Data .............................................................................. 30
  Adding Color ............................................................................ 30
  Charting Data ........................................................................... 30
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query Processing Order</td>
<td>95</td>
</tr>
<tr>
<td>Saving Queries</td>
<td>96</td>
</tr>
<tr>
<td>Cancelling Queries</td>
<td>96</td>
</tr>
<tr>
<td>Building Subqueries</td>
<td>96</td>
</tr>
<tr>
<td>Regular Subqueries</td>
<td>97</td>
</tr>
<tr>
<td>Modifying Subqueries</td>
<td>97</td>
</tr>
<tr>
<td>Correlated Subqueries</td>
<td>98</td>
</tr>
<tr>
<td>Derived Tables</td>
<td>99</td>
</tr>
<tr>
<td>Derived Tables Rules and Behavior</td>
<td>100</td>
</tr>
<tr>
<td>Derived Tables and SQL</td>
<td>102</td>
</tr>
<tr>
<td>Working with Query Section Data</td>
<td>104</td>
</tr>
<tr>
<td>Processing Results to a Database Table</td>
<td>105</td>
</tr>
<tr>
<td>Estimating Query Size</td>
<td>106</td>
</tr>
<tr>
<td>Displaying Database Remarks</td>
<td>106</td>
</tr>
<tr>
<td>Preaggregating Data Using Functions</td>
<td>107</td>
</tr>
<tr>
<td>Appending Queries</td>
<td>109</td>
</tr>
<tr>
<td>Using Local Results</td>
<td>111</td>
</tr>
<tr>
<td>Using Stored Procedures</td>
<td>112</td>
</tr>
<tr>
<td>Setting Query Options</td>
<td>113</td>
</tr>
<tr>
<td>Editing with SQL</td>
<td>114</td>
</tr>
<tr>
<td>Editing with SQL</td>
<td>114</td>
</tr>
<tr>
<td>Importing SQL Files</td>
<td>115</td>
</tr>
<tr>
<td>Reviewing the SQL Log</td>
<td>117</td>
</tr>
<tr>
<td>Importing Data Files</td>
<td>117</td>
</tr>
<tr>
<td>Setting Data Type Properties</td>
<td>117</td>
</tr>
<tr>
<td>Picture (BLOB Image) Support</td>
<td>119</td>
</tr>
<tr>
<td>Adding Pictures in Query</td>
<td>119</td>
</tr>
<tr>
<td>Adding Tooltips to Pictures</td>
<td>120</td>
</tr>
<tr>
<td>Working with Pictures in Results and Tables</td>
<td>120</td>
</tr>
<tr>
<td>Working with Pictures and Computed Items</td>
<td>120</td>
</tr>
<tr>
<td>Working with Pictures in Reports</td>
<td>121</td>
</tr>
<tr>
<td>Resource Manager</td>
<td>121</td>
</tr>
<tr>
<td>Picture General Properties</td>
<td>124</td>
</tr>
<tr>
<td>Query Menu Command Reference</td>
<td>124</td>
</tr>
<tr>
<td>Insert Query</td>
<td>125</td>
</tr>
<tr>
<td>Process Results to Database Table</td>
<td>127</td>
</tr>
<tr>
<td>Chapter 5. Results</td>
<td>129</td>
</tr>
<tr>
<td>Results Section</td>
<td>129</td>
</tr>
</tbody>
</table>
Chapter 12. Computed Items ......................................................... 323

About Computed Items .......................................................... 323
  Computing New Data .......................................................... 323
  Computed Items in Sections .................................................. 324
  Computed Items and Data Functions ....................................... 328
  Pivot Options ........................................................................ 329
Adding Computed Items .......................................................... 329
  Operators ............................................................................. 330
Referencing Topics in Computed Items ....................................... 333
Using Functions ...................................................................... 333
  Scalar Functions .................................................................. 334
  Data Functions ..................................................................... 342
  Teradata Standard Functions ................................................ 344
  Teradata OLAP Functions ...................................................... 346
  ANSI SQL Functions ............................................................ 351
  Functions for Returning the Day of the Week .......................... 355
Common Computed Item Examples ............................................ 356
  Math Functions ................................................................. 357
  Central Tendency ............................................................... 357
  Calculated Averages ........................................................... 357
  Percentile ........................................................................... 358
  Rank .................................................................................... 359
  Project Sales ........................................................................ 360
Scalar Function Examples .......................................................... 360
Chapter 13. Applying Sorts ........................................................ 401
  Sorting Data ........................................................................ 401
  Simple Sorts ....................................................................... 401
  Sort Lines .......................................................................... 402
  Complex Sorting .................................................................. 402
    Complex Sorting in the Query, Results, and Table Sections .... 402
    Complex Sorting in Chart, Pivot, and OLAPQuery Reports ... 404
Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.
Overview

Oracle Hyperion Interactive Reporting connects business users to data and gives them a complete set of tools to support business decisions including ad hoc client/server querying, reporting, and analysis all in one application. Interactive Reporting provides the following capabilities:

- Data extraction and analysis
- Reporting and distribution
- Platform development

Interactive Reporting is an all-in-one query, data analysis, and reporting tool. The interface is highly intuitive and provides an easy-to-navigate environment for data exploration and decision making. With a consistent design paradigm for query, pivot, charting, and reporting, users at any level move fluidly through cascading dashboards—finding answers fast. Trends and anomalies are automatically highlighted, and robust formatting tools enable users to easily build free-form, presentation-quality reports for broad-scale publishing across their organization.

Interactive Reporting Features

Interactive Reporting features include:

- Support for all industry-standard databases.
- A point-and-click interface for intuitive custom query and report building.
- Support for Microsoft Windows and Motif (UNIX), with complete file compatibility between the platforms.
- Easy, nonprocedural navigation between query and reporting sections.
- An extensive online help system that provides assistance for features and document construction.
● A drag-and-drop data layout tool for developing reports and analyzing data.

● Interactive pivot reporting that lets you perform unrestricted drill-down analysis of different data relationships.

● Extensive formatting tools for creating compelling data presentations.

● An easy-to-use, interactive charting utility for graphically displaying and drilling-down into data.

How Interactive Reporting Works

Interactive Reporting enables you to access and analyze information stored in different company data sources. It connects you to data and supplies a complete set of tools that enable you to build queries quickly and intuitively—by clicking icons and manipulating objects. Interactive Reporting automatically builds a query to your specifications, send it to the database, and displays the retrieved data as a table of results.

There are many types of data sources, but the most prevalent are relational databases and multidimensional databases.

Relational Databases

A relational database is a collection of data items organized as a set of formally described tables from which data can be accessed or reassembled in many different ways without having to reorganize the database tables.

The definition of a relational database results in a table of metadata or formal descriptions of the tables, columns, domains, and constraints. Metadata is literally “data about data.”

Multidimensional Databases

A multidimensional database is a data cube that provides multidimensional views of business data. Multidimensional databases are OLAP servers that enable you to easily and selectively extract and view data from different points of view. Multidimensional databases consider each data attribute as a separate dimension and allow you to create hierarchies within a dimension.

OLAP (On-Line Analytical Processing) designates a category of applications and technologies that allow the collection, storage, manipulation, and reproduction of multidimensional data, with the goal of analysis. OLAP provides for the fast analysis of multidimensional shared information.

About Interactive Reporting Documents

Interactive Reporting documents (BQYs) are files you create and use to retrieve information from a database, analyze the information, and build reports. Because Interactive Reporting is an integrated query, analysis, and reporting tool, Interactive Reporting documents have multiple
sections, each of which governs one part of the query and reporting process. You create sections progressively: first you query a database, then retrieve results, and last generate reports.

Interactive Reporting documents can contain data from any number of relational databases queries, multidimensional database queries, and/or from imported data. Documents usually include one or more of the following items:

- A **data model**, which is a visual representation of actual database tables
- A **query** or multiple queries for retrieving a subset of data from the database
- **Join** options, including local joins between different data sets within a single document, local join filters, and optional join path generation
- A **results** set displayed in a table-style format
- **Reports** presenting customized hierarchical views of your data
- Multidimensional **pivot** tables that permit drill-down analysis of data results
- **Charts** that graphically display your query results and allow different angles of vision on the data.

All Interactive Reporting documents usually have at least one Query section and one Results section. From the Results section, you can create multiple Pivot, Chart, Table, and Report sections to analyze and present data. Developers can also create Dashboard sections, which provide an automated push-button interface to a document for use by other users across the enterprise.

### Data Source Connections

For Interactive Reporting users, the process of creating a new document and connecting to a database is simple. You select an Interactive Reporting database connection file (.oce) for the database server and enter your database password. You can select either a new or an existing Interactive Reporting database connection file.

The way you choose an Interactive Reporting database connection file depends on the data model, document, or edition:

- When a data model is available for the Query, Interactive Reporting automatically prompts you for connection information. You connect to a database when you download a data model and process a query to retrieve a data set. In addition, you must be connected to show values for a server filter, to use server functions to create a computed item, or to schedule a document.

- When you open Interactive Reporting to begin a work session (for example, by downloading a data model from an Oracle Hyperion Reporting and Analysis repository, or creating a new data model), you must select the correct connection for the targeted database.

Interactive Reporting database connections retain all the information necessary to log onto a specific configuration of database and connection API software. They also retain DBMS-specific connection preferences and specifications for automatic access to metadata. This simplifies the connection process for company personnel by transparently handling host and configuration
information. Each user can substitute a database user name when using the Interactive Reporting
database connection, which enforces security measures and privileges that are centralized at the
database server.

Interactive Reporting database connections have significant advantages in network
environments with many database users. One connection can be created for each Interactive
Reporting database connection file in the environment and shared with each end-user. Because
passwords are not saved with the Interactive Reporting database connection, distribution does
not provide unauthorized access to any user who receives the wrong Interactive Reporting
database connection file or acquires it from other sources.

Data Models

After connecting to a database, Interactive Reporting presents subsets of the database contents
in the Query section through custom views called data models, which are visual representations
of actual database tables. You use a data model to interact with a database to create queries which
fetch data from the database.

Data models make the database more accessible by:

- Substituting descriptive names for arcane database table and column names.
- Creating custom views of the data.
- Adding computed fields for performing calculations on the retrieved data.

Standard data models derived from database tables enable you to create metatopics—virtual
views independent of the actual database. Metatopics standardize complex calculations and
simplify views of the underlying data with intuitive topics customized for business needs.

Users can create their own data models, or use prebuilt data models stored in the centralized
Reporting and Analysis repository.

You can also provide a document that contains a master data model from which other users can
build one or more queries. This master data model allows your users to concentrate on specific
data, not how to set up the data access. Any data model can be promoted to a master data model.

You can offer users a raw look at the table schema, or hide the complexity by first creating one
or more metatopics and then promoting the data model. Each time the user adds a new query,
Interactive Reporting asks if the query should be linked to the master data model. Any linked
queries inherit changes made to the master data model, but the data model is locked and cannot
be modified. (Only the master data model can be changed.)

Queries

A query is a request for information from a database. Queries take the form of a command
language that lets you select, insert, update, find out the location of data, and so forth.

The standard command language for getting information from and updating a relational
database is Structured Query Language (SQL). SQL statements are used both for interactive
queries for information from a relational database and for gathering data for reports.
Multidimensional databases also require a language that allows you to express multidimensional queries; however, to date, there is no standard. MDX (Multidimensional Expression Language) is used by Microsoft’s OLE DB for OLAP API and OLAP Services. For Oracle Essbase, Interactive Reporting uses MaxL (Multidimensional Access Language). MDSQL (Multidimensional Query Language) is yet another query language.

With Interactive Reporting, you do not need to know SQL or any multidimensional query languages to create powerful database queries. You build queries by choosing the data to retrieve from a visual representation of the database.

Interactive Reporting offers two query methods.

- **Query**—Displays the structure of the relational database as tables (or topics), which are used to create a data model—the visual representation of the database tables. An Interactive Reporting document can have more than one data model.

- **OLAPQuery**—Displays the structure of the multidimensional database as a hierarchical tree. OLAP queries are displayed in a form similar to a pivot table, except the data comes straight from the OLAP server.

- **CubeQuery**—Is the Interactive Reporting interface with Essbase 7.x, and 9.x databases. It provides access to the cube in which multidimensional data can be analyzed and complex calculations can be applied before and after the queries are processed. The resulting data set can be shown with relational data side-by-side in a dashboard report, or downloaded to a results set and joined to relational sources.

An Interactive Reporting document can contain one or more relational Query sections, as well as one or more OLAPQuery sections. This allows users access to information in organizations that have both types of databases.

**Analysis and Reporting**

Once a query is processed and data results are returned to the desktop, you can use the Interactive Reporting powerful reporting and analysis tools to create custom views, cross-sections, and drill-downs to slice and dice data and view multidimensional relationships.

You can create multiple views of the data or display the information in any form and from any angle possible. At any time, you can reconnect to the server and update your reports and charts with fresh data from the database.

You can also use Interactive Reporting to work autonomously with data after disconnecting from the server. Even without a database connection, you can continue to analyze data and produce reports. You can save results in the desired format for additional refinement in Interactive Reporting, or export the data to other applications for further analysis.

Interactive Reporting enables you to create a wide variety of reports:

- **Tables**—Columnar arrangements of data. Tables are used as building blocks in other reporting sections. You can apply filters to tables, add computed items, include subtotals and grand totals, as well as summary totals such as sum, count or average.
- **Pivot tables**—Interactive tables that quickly summarize or cross-tabulate large amounts of data. You can rotate rows and columns to see different summaries of data or display the details for areas of interest. A pivot table summarized data by using a summary function, such as Sum, Count, or Average. You can include subtotals and grand totals automatically, or use your own formulas by adding computed items.

- **Charts**—Visual display of information; fully interactive, three-dimensional views of data. Interactive Reporting displays data from results sets as bars, lines, columns, pie slices, or other shapes in the chart. When you create a chart, the values from the worksheet are automatically represented in the chart. Charts are linked to the data they are created from and are updated when data is changed.

- **Custom reports**—Creates free-form presentation-quality reports with graphic objects, predefined fields, band-style report data from multiple data sources and computed fields, charts, and pivots; Smart reports enable you to embed charts and pivot tables and show only the data that is relevant to the section in which they are placed.

### Turning Data into Information

Data is meaningless unless it can be analyzed and interpreted. Analysis depends on consolidating and summarizing data through mathematical operations that reveal meaningful relationships, also called aggregation. The result is a summary of the data at a higher level, which summarizes and consolidates data from a lower level.

Aggregation is a critical feature of data analysis. Successful and rapid interpretation of data requires an easy method of aggregating data and representing it for easy interpretation. Aggregation techniques are easy to master. With the simple drag-and-drop of a data item, you can reorganize your data. Remove an item or drill down into your data, and you disaggregate your data.

Interactive Reporting provides a great deal of flexibility in how you choose to aggregate your data. One possibility is to aggregate your data at the time of your query, called server aggregation or preaggregation. In this case, the database server actually performs the aggregation for you.

An advantage of server aggregation is the network traffic is reduced and takes less time because the database server. If you preaggregate data at the server, you might not reach any substantial level of data because detailed data was excluded data by preaggregating. If you do not know your database tables, you could eliminate data that might be important for analysis. Also, preaggregation requires more server processing resources.

Alternatively, you can aggregate data on your desktop without involving the server. Interactive Reporting automatically aggregates your data for you in report sections.
Starting Interactive Reporting

To start Interactive Reporting, select Start, then Programs, then Hyperion, then Reporting and Analysis, then Interactive Reporting, and then Studio.

Whenever you start Interactive Reporting, you must create a new Interactive Reporting document file or open one that has already been created.

Opening and Saving a Sample File

During the installation process, Interactive Reporting installs several sample files. This tutorial uses the document named Sample1mod.bqy to familiarize you with many Interactive Reporting features.

To open Sample1mod.bqy:

1. In the Welcome dialog box, select Recent Documents from the Open Existing Documents area.
2. Click Browse and navigate to the Samples folder (for example, C:\Hyperion\BIPlus\docs\samples).

Note: If you cannot locate the sample files, please contact your administrator.

3. Select Sample1mod.bqy and click Open.

The content area is displayed and shows the sample document with the Dashboard active.

4. Select File, then Save Options, and then Save Query Results with Document.

The Save Query Results With Document dialog box is displayed.
5. Make sure that all of the items in the Query, Results, and Computed Columns lists are selected, and then click **OK**.

6. Select **File**, then **Save As** to open the Save File dialog box.

7. Type a new name for the sample document (for example, *practice.bqy*) and click **Save**.

### Looking at a Simple Query

The Query section is the foundation of an Interactive Reporting document. It is the space where you build questions for the database.

The Query section in the sample document is labelled *SalesQuery*. It is a simple query constructed for a fictitious company named Books, Movies, and Video (BMV). BMV distributes books, movies, and videos to a number of retail stores. To predict trends and locate strengths and weaknesses in its distribution techniques, BMV warehouses extensive data.

➢ To view the SalesQuery section, select **SalesQuery** in the Sections pane.

### Viewing Database Tables

In *Sample1.bqy*, there are four tables, also called *topics*, in the Content pane. These four topics represent tables in the BMV database. The database tables are listed in the Catalog pane, located to the left of the Content pane.

➢ To view all the database tables:

1. **In the Catalog pane, click the + (plus) sign to the left of Tables**.

   If you are not connected to the sample database, the Hyperion Sample1.oce dialog box is displayed and prompts you for a user name and password.

   An Interactive Reporting database connection file is a file that enables you to connect to a database. The following icons on the Status bar indicate your connection status:

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="icon" alt="Connected" /></td>
<td><strong>Connected</strong>—You must be connected to a database to work in the Query section.</td>
</tr>
<tr>
<td><img src="icon" alt="Disconnected" /></td>
<td><strong>Disconnected</strong>—You do not need to be connected to a database for many Interactive Reporting tasks.</td>
</tr>
</tbody>
</table>

   For the sample database, you do not need to enter any information in this dialog box.

2. **Click **OK** to connect to the database**.

   The Tables tree in the Catalog pane expands to show all of the tables in the database.
Adding Topics to a Query

- To include data from a particular table in your query, drag and drop the table from the Catalog pane to the Content pane.

Each topic contains a list of topic items that represent fields or rows of data in the database. In Sample1.bqy, the topics included in the query are Periods, Sales Fact, Products, and Stores.

You build queries by adding topics from the Content pane to the Request line. You can drag and drop any topic item to the Request line. When you process a query, Interactive Reporting returns data for all the topic items present on the Request line. In the sample, several topic items from each table have already been dragged to the Request line (for example, Unit Sales, Amount Sales, Year, Quarter, and so on).

You can also add filters to the data, or specify columns by which to sort the data. You can apply filters and sorts in either the Query section or the Results section.

In the Query section, filters instruct the database server to filter unwanted information from the requested data. Sorts instruct the database server to retrieve data to your desktop in a particular order.

Viewing Results

Data returned from a query is displayed in the Results section. Each column of results corresponds to items on the Request Line in the Query section. Request items are listed in the Catalog pane.

- To view the Results section, select SalesResults in the Sections pane.

Use the arrow buttons on the Section title bar to compare the items in the Query and Results sections.

Reordering Columns

- To reorder columns in the Results section, drag one column to the left or right of another column.

You can also move column labels in the data layout.

Sorting Columns

- To sort columns of data:
  1. If the Sort line does not show, click Sort(0) on the Section title bar.
  2. Drag Product Line from the Catalog pane to the Sort line.
  3. Click Sort Now to group items by Product Line.
Filtering Data

Use the Filter line to limit the data displayed in a column.

- To filter data:
  1. If the filter line does not show, click filters(0) on the Section title bar.
  2. Drag Amount Sales from the Catalog pane to the filter line.
     The Filter dialog box is displayed.
  3. Select \( \geq \text{Greater or Equal} \) from the filter drop-down list.
  4. Click Custom Values, type 100000 in the field provided, and click OK.

All entries with sales amounts less than $100,000 dollars are dropped from the Results section.

- To remove a data filter, delete Amount Sale from the Filter line.
  The data is redisplayed when you remove the filter.

Calculating Data

You can perform calculations on columns of numeric data.

- To sum up numeric data:
  1. Select the Amount Sales column.
     The Insert Grand Total dialog displays.
  2. Select Results, then Grand Total and then click OK.
  3. Scroll down to the bottom of the table to view the total amount of sales.

BMV total sales for 1999-2000 are $132,881.

If the figure does not match $132,881, remove any filters imposed on the results set. Delete any filters from the Filter line and view the total again. The sum automatically adjusts.

If you see ########, the number is too large to fit in the designated space.

- To resize the column, select Format, then Column, and then Auto Size Width.

Pivoting Data

In Sample1.bqi, the SalesPivot section is a simple example of a pivot table. Pivot tables provide multiple angles on your data.

- To view the Pivot section, select SalesPivot in the Section pane:
  The SalesPivot section is displayed.
Pivot tables allow you to quickly summarize data in the Results section and immediately see the relationships between different dimensions of your data. These reports pivot to provide fresh angles of vision on your data.

➤ To create a pivot table:

1. Select SalesQuery in the Section pane.
2. Select Insert, then New Pivot.
3. If the data layout is not already visible, click Data Layout on the Section title bar.
   Drag one or more items from the Catalog pane into each of the data layout panes. Remove items from the data layout panes and add new ones. Use the Top Labels pane and Side Labels pane for text. Use the Facts pane for numeric values.

➤ To pivot views:

1. Click the dimension tab at the end of the row labels.
2. Drag the tab down and left to turn your row into a column.
   The same data is displayed but with a different angle on the data.
3. Take the tab of the newly formed column and drag it so it becomes a row again.

### Calculating Totals and Subtotals

To make effective use of data, you may need to generate totals or subtotals.

➤ To calculate totals and subtotals (as displayed above):

1. Drag Product Line and Region to the Row pane in data layout.
2. Drag Year to the Column pane in data layout.
3. Drag AmountSales to the Facts pane in data layout.
4. Select the dimension handle for Region (click at the bottom of the Region column).
5. Select Pivot, then Add Totals.
   A row is added that shows the total number of product line sales for all regions.
6. Select the handle for Product Line and select Pivot, then Add Totals.
   A row is added showing the subtotals (also know as break totals) for each product line by region.

### Drilling Down

More data is available for analysis than is currently visible in your pivot table.
To drill anywhere, select the Region column and select **Pivot**, then **Drill Anywhere**, and then **Country**.

A column is added to your pivot table that shows countries within region.

To restore the original pivot table without the Unit Sales column, select the Country column and select **Pivot**, then **Drillup**.

**Hiding Data**

You can temporarily hide data.

To hide an item, select a label such as the Americas label and select **Pivot**, then **Hide Items**.

To focus on an item, select a label such as the Americas label and select **Pivot**, then **Focus On Items**.

To restore your excluded items, select **Pivot**, then **Show All Items**.

**Adding Color**

Use the Format toolbar to add color to emphasize aspects of your pivot table.

**Note:** If the Format toolbar is not visible, select **View**, then **Toolbars**, and then **Formatting**.

To add a line color:

1. **Click a label, dimension handle, or column.**
2. On the Format toolbar, open the **Line Color** list and select a color from the palette.

To add a fill color:

1. **Click a label, dimension handle, or column.**
2. On the Format toolbar, open the **Fill Color** list and select a color from the palette.

To add a text color:

1. **Click a label, dimension handle, or column.**
2. On the Format toolbar, open the **Text Color** list and select a color from the palette.

**Charting Data**

Charting features make graphic analysis of data and powerful presentations simple. In **Sample1.bqy**, the **AllChart** section is a chart based on the data from the original query.
To view the Chart section, select Pivot, then Show All Items.
The AllChart section is displayed.

To change the chart type and format, select a chart format from the Chart list.
As you shift from one chart type to another, data may be shifted to different axes.

To create a new chart:
1. In the Section pane, select SalesQuery.
2. Select Insert, then New Chart.
   Chart is displayed in the Section pane.
3. In the Section pane, double-click Chart.
   The Section Label dialog box is displayed.
4. Delete Chart and type: Unit Sales Region.
5. Click OK.
6. If the data layout is not already visible, click Data Layout on the Section title bar.
   Numeric values (facts) are placed in the y pane in the data layout. Non-numeric data (dimensions) are placed in the x pane and z pane in the data layout.
7. Drag Unit Sales to the x pane in data layout.
8. Drag Region to the z pane in data layout.
10. Click a Legend box to change the distribution and patterns of colors.

Sorting Charts
It is often useful to order the bars of a chart sequentially.

To sort your chart:
1. If the Sort line is not visible, click Sort on the Section title bar.
   Sort provides drop-down menus to select sort criteria. Experiment sorting.
2. Click the Ascending or Descending sort icon on the Standard toolbar.
   The data on the chart is rearranged in ascending or descending order.

Inserting Text Labels
You can insert a text label anywhere within the chart to further explain or emphasize a chart component.
To insert a text label:

1. On the shortcut menu, click Insert Text.

   The Set Inserted Text dialog box is displayed.

2. Type the text you want to insert and click OK.

3. Drag the text box to any position on the chart.

**Designing Reports**

The Report Designer is another way to analyze and present data and offers a great deal of formatting flexibility. You can embed either a pivot table or a chart directly in a report.

To view a Report section, in the Section pane, select RegionReport.

The RegionReport section is displayed.

To create a new tabular report:

1. Select Insert, then New Report.

   Blank columns are displayed in the Content pane.

2. Click Groups and Table on the Section title bar to see all panes in the data layout.

   In the Report section, create reports by dragging items from the Catalog pane to the Groups data layout and the Table data layout.

   The Groups data layout allows you to drag non-quantifiable items and create separate tables for each label in a report group. In other words, when you designate a results column to serve as a report group, Interactive Reporting organizer data in repeating collections of records according to the Report group.

   The Table data layout is divided into the Dimensions and Facts panes. The Facts pane allows you to drag quantifiable items to show quantity. Totals are generated automatically in the report body. The Dimensions pane allows you to drag non-quantifiable items to itemize the facts.

3. In the Catalog pane, double-click the SalesQuery folder, and click the plus sign to the left of the SalesResults folder.

4. Drag Year and Quarter into the Dimensions pane in Table data layout.

5. Drag Amount Sales into the Facts pane in Table data layout.


   The new report has visible section boundaries. Results columns added to the Facts pane are automatically summed and the totals are displayed in the tabular report of the report table column.

   **Note:** You can disable automatic totaling.
Creating Smart Reports

Smart reports allow you to embed charts and pivot tables into a report body. These reports show only the data that is relevant to the report section in which they are placed. For example, if the report is grouped by year, and you insert a chart in the report body, the chart replicates automatically so that there is a chart for each year of data in the report. Each year for the chart contains data specific to that year.

To create a Smart report:

1. In the Catalog pane, click the plus sign to the left of the SalesQuery folder.
2. Drag a Chart icon (use either Unit Sales Region or AllChart) and position it just above the table in the report body.

   Allow some white space between the embedded chart and the table.

   In the newly-created Smart report, the chart changes to reflect the data specific to each country within a territory.

Sorting Columns

To sort table columns, select a table column, and drag it to the Sort line. Multiple table columns can be added to the Sort line to create a nested sort.

Setting Up a Report Page

To set up a report page:

1. Select Report, then Report Setup.

   The Report Page Setup dialog box is displayed.

2. On the Margin page, set margin sizes (top, bottom, left, and right) and click OK.

3. On the Column page, specify the number of columns on a page and the default column width and spacing,

4. Click OK.

To insert a page break, select a Report Group header (be careful not to select the Report Group label) and on the shortcut menu, select Page Break After or Page Break Before.

Page breaks can be inserted before and after a report body or before and after a Report Group label.

To remove a page break, select a Report Group header with a page break applied, and on the shortcut menu, select Page Break After or Page Break Before to remove the check mark.
Interactive Reporting Basics

In This Chapter

- Maintaining Interactive Reporting Document Files ...................................................... 35
- Exporting to Microsoft Office HTML File Formats ........................................................ 48
- Using Edit Commands ..................................................................................... 65
- Changing Views ............................................................................................ 66
- Inserting Sections and Breaks ............................................................................ 66
- Working with the Tools Menu ............................................................................. 66
- Formatting Numeric Data Types .......................................................................... 66
- Working with Document Sections ........................................................................ 68
- Setting Interactive Reporting Options .................................................................... 71
- Export Properties ........................................................................................... 84
- Export Document as Web Page ........................................................................... 84
- Export to HTML Wizard ..................................................................................... 85
- Conditional Formatting .................................................................................... 87

Maintaining Interactive Reporting Document Files

Interactive Reporting documents are files created by Interactive Reporting that you store on your personal computer. Each Interactive Reporting document file consists of one or more sections. Additionally, the Interactive Reporting document file can be used to retrieve information from a database, analyze the information, and build reports. Maintaining Interactive Reporting document files involves the tasks described in these topics:

- "Creating Interactive Reporting Document Files" on page 36
- "Opening Interactive Reporting Document Files" on page 36
- "Closing Interactive Reporting Document Files" on page 36
- "Saving Documents" on page 36
- "Specifying Save Options" on page 37
- "Save Query Results With Document" on page 37
- "Compressing Interactive Reporting Document Files" on page 37
- "Saving Interactive Reporting Documents to the Reporting and Analysis Repository in Interactive Reporting Web Client" on page 39
- "Password Protecting Interactive Reporting Document Files" on page 38
Creating Interactive Reporting Document Files

To create an Interactive Reporting document file:

1. **Select File, then New, or click the New icon on the Standard toolbar.**
   The New File dialog box is displayed.

2. **Select an Interactive Reporting database connection file.**
   For more information about Interactive Reporting database connection files, see “Data Source Connections” on page 21.

3. **Enter your user name and password and click OK.**

Opening Interactive Reporting Document Files

To open an Interactive Reporting document file:

1. **Select File, then Open.**
   The Open File dialog box is displayed.

2. **Select the Interactive Reporting document file and click OK.**

   **Tip:** When opening an Interactive Reporting document file with your browser commands, you may have to adjust the type of field displayed. By default, browsers show files with HTM extensions only. To view Interactive Reporting documents, set the file type to “all files or “.bqy”.

Closing Interactive Reporting Document Files

**Note:** Document shutdown scripts are only executed when an Interactive Reporting document file is closed.

To close an Interactive Reporting document file, select **File, then Close.**
You are prompted to save the changes.

Saving Documents

When you save a document, you save the formatting and layout of all sections it contains. You can also use the **Save Query Results With Document** option to work offline.
Note: The Save and Save As commands in Oracle Hyperion Enterprise Performance Management Workspace do not execute the document shutdown scripts when saving the Interactive Reporting document file. These scripts only execute when an Interactive Reporting document file is closed.

To save an Interactive Reporting document file, click the Save button on the standard toolbar, or select File, then Save.

To save the Interactive Reporting document file under another name:
1 Select File, then Save As.
Save File is displayed.
2 Enter an Interactive Reporting document name in the File Name field and click Save.

Specifying Save Options

To select specific save options, select File, then Save Options and choose an option:

- Specify Save Options
- Work Offline in Interactive Reporting Web Client
- Compressing Interactive Reporting Document Files
- Password Protecting Interactive Reporting Document Files
- Password Protect Designer Mode

Save Query Results With Document

This option saves the results of a query and any computed columns (from Results) with an Interactive Reporting document file and allows you to analyze and generate reports using the results set without being connected to the database. Computed columns can be saved as snapshots, or with the document by section.

Compressing Interactive Reporting Document Files

The Compress Document option condenses the number of bytes in a document, and saves the Interactive Reporting document file in a compressed file format. It enables you to quickly transmit the Interactive Reporting document file and saves valuable storage space. Compression concentrates the number of bytes by removing empty data fields and unnecessary information. Interactive Reporting automatically decompresses the file when you reopen it.

Tip: To configure Interactive Reporting to compress all Interactive Reporting document files when saving them, select Compress All Documents on the General tab. (See “General Options” on page 74.)
Password Protecting Interactive Reporting Document Files

Use the Password Protect Document option to restrict access to an Interactive Reporting document file and ensure the confidentiality of sensitive documents. This option requires users to input a correct password for document access.

To assign a password to an Interactive Reporting document file:

1. Select File, then Save Options, and then Password Protect Document.

   Password Protect Document is displayed.

2. In the Password field, type the password to be assigned to the Interactive Reporting document file.

   Passwords can contain up to 38 alphanumeric characters and are case sensitive.

3. In the Verify Password field, retype the password and click OK.

   If the password and the verification do not match, an error message is displayed.

   **Caution!** Keep a list of each Interactive Reporting document file and the specific password needed to access it in a secure place.

Password Protect Designer Mode

Use the Password Protect Designer Mode option to:

- Restrict access to a document script and the scripting environment. This ensures the integrity and confidentiality of sensitive scripts in a Dashboard section and when you are working with document scripts.
- Enable encryption for startup, shutdown, document, and Dashboard scripts associated with Interactive Reporting document files.

You must supply a valid password to encrypt scripts. (Encrypted scripts are not functional in releases prior to 6.6.3 or 8.1.) You can still password protect an Interactive Reporting document file without encrypting any scripts associated with it.

You can define a default encryption state on the General options of Designer Options. When you create an Interactive Reporting document file, the default encryption state matches the one in Designer Options. Despite the default setting, scripts are only encrypted if the Interactive Reporting document file has a valid Design mode password.

**Note:** Password Encryption accommodates user credentials that contain characters from a language other than the Windows machine locale. Password encryption extends to database passwords specified in Interactive Reporting database connection files, Interactive Reporting document files and sections.

To password protect a script:

1. Select File, then Save Options, and then Password Protect Designer Mode.
Password Protect Designer Mode is displayed.

2 In the **Password** field, enter the password required to open the document script.
Passwords can contain up to 38 alphanumeric characters and are case sensitive.

3 In the **Verify Password** field, retype the password and click **OK**.
If the password and the verification do not match, an error message is displayed.

To open a password protected script:

1 Select one action:
   - Select **File**, then **Document Scripts**.
   - Click the Design/Run icon in the Dashboard section run mode.
     Unlock Design Mode is displayed.

2 Enter the password required to open the script and click **OK**.

To encrypt a script:

1 In the **Password** field, enter the design mode password required to open the script.
Passwords can contain up to 38 alphanumeric characters and are case sensitive.

2 In the **Verify Password** field, retype the password and click **OK**.
If the password and the verification do not match, an error message is displayed.

3 Select **Encrypt Dashboard scripts**.

4 Click **OK**.

### Saving Interactive Reporting Documents to the Reporting and Analysis Repository in Interactive Reporting Web Client

When you modify an Interactive Reporting document or job, you can save the changed document to the Reporting and Analysis repository. The Reporting and Analysis repository is an efficient way to manage documents and distribute documents over a wide network for end-user query and reporting.

Saving the document to the Reporting and Analysis repository can be made either by saving the document with changes to the original document using the Save to Repository option. To save the original document under a new name or when you do not have permission to overwrite a document, use the Save As to the Repository.
To save an document to the Repository, select **File**, then **Save to Repository**.

**Saving As to the Reporting and Analysis Repository in Interactive Reporting Web Client**

When saving to the Repository, use the Save As to the Repository to save an original Interactive Reporting document under a new name, or when you need to save a document, but do not have permission to overwrite the original.

To save the document under a another name:

1. **Select File**, then **Save to Repository As**.

   The Save As To Repository File dialog box is displayed.

2. **Navigate to the folder in which to save the Interactive Reporting document file in the Look in field**:
   - To select an existing folder, select a folder from the list show by double clicking the folder. You can drill down into a sub-folders within a folder also by double clicking the selected folder.
   - To move up one level in the Look in field, click **Go up**.
   - To create a new folder, click **New Folder**. When the Create New Folder dialog box is displayed, type the name of the folder in which to save the document.
   - To show hidden folders, click **Show Hidden**.

3. **Type a name for the document in the Name field**.

4. **Optional**: Enter a document description in the Description field.

5. **Select** Interactive Reporting document **from the Type drop-down list**.

6. **Click Save**.

**Create New Folder**

Use the Create New Folder to add a new folder in which to save Interactive Reporting document files in the repository.

- Type the name of new folder in the **Please Enter the Name** field and click **Save**.

To nest the new folder within a parent folder, navigate to the parent folder before creating the new folder.

**Work Offline in Interactive Reporting Web Client**

This option enables you to save an Interactive Reporting document file locally so that you can work with it offline if you have an installed version of Oracle Hyperion Interactive Reporting Web Client. In this case, you are not prompted to connect to the EPM Workspace when opening.
the document. You set this option in the Oracle Hyperion Interactive Reporting Studio or Interactive Reporting Web Client.

**Note:** Once the Interactive Reporting document file is imported to the EPM Workspace, a user with a Viewer/Dynamic Viewer and Explorer role can open the saved locally file without a logon. A user with Explorer Role cannot save the Interactive Reporting document file back to the repository. As a result there is no option to work offline in Interactive Reporting Web Client.

### Working with Data Sources

You can use a prebuilt data model or you can import files from other applications and use the data as the source for your document. As you work with data sources, you can use the commands described in these topics:

- “Open From Repository in Interactive Reporting Studio” on page 41
- “Save To Repository in Interactive Reporting Studio” on page 42
- “Import Data File” on page 43
- “Import SQL” on page 44

### Open From Repository in Interactive Reporting Studio

Use Open From Repository command to select Interactive Reporting database connection file (.oce) and repository object (data models, standard queries, and standard queries with reports) in Interactive Reporting Studio.

Repository objects ensure that your documents maintain a standard look and feel. If you are not experienced with database access, repository objects also help you get the data you need quickly and easily.

To open a document using a repository object:

1. **Select File, then Open From Repository, and then Select.**

   The Select Connection dialog box is displayed.

   **Note:** You can also use the Interactive Reporting database connection file currently in use if there is one. Current are listed below the Select menu item.

2. **Select the Interactive Reporting database connection file (.oce) to use and click OK.**

3. **In the Password dialog box, type your user name and password and click OK.**

   The Open From Repository dialog box is displayed and displays information about the selected object.

   - **Unique Name**—Name of repository object
   - **Creator**—Creator of the repository object
• **Created**—Date on which the repository object was created
• **Description**—General description of the repository object, its contents, and the type of information that can be queried

4. **Navigate through the repository tree to select the repository object to use and click Open.**

The repository object is downloaded to the appropriate section.

---

**Save To Repository in Interactive Reporting Studio**

Designers can use the Save To Repository command to upload repository objects (data models, standard queries, and standard queries with reports) for version-controlled distribution to networked Interactive Reporting Studio users, and to describe the object to users browsing the repository.

**Note:** When you store objects in the Interactive Reporting repository, make the Interactive Reporting database connection file available to users.

➢ To upload an object to the Interactive Reporting repository:

1. **With the repository object to upload open in Interactive Reporting Studio, select File, then Save to Repository.**

2. **Click Select to launch the Select Connection File dialog box.**

   The Save to Repository dialog box is displayed and the Model Tab defaults.

3. **If you have modified a Data Model, built a standard query or a standard query with a report, the Model Type field shows the types of objects that can be saved to the Repository at this time. Select the repository type. Available types are:**
   • Data Model
   • Standard Query
   • Standard Query with Reports

4. **Enter a descriptive name for the object in the Unique Name field.**

5. **Type your name as the object creator in the Creator field.**

   The date on which the object was saved to the Repository defaults in the Created field.

6. **Unlock the repository object to allow users to make modifications to it by selecting Locked/Linked Object (Required for ADR) field so that no check mark appears in the field.**

   If the repository object is subject to Automatic Distributed Refresh, the object must be locked (the Locked/Linked Object field must be checked).

7. **Click Prompt for Sync on Download to prompt users to make a copy of an object that has been saved locally if a new version of the object exists in the repository.**

8. **Enter a description of the object, its attributes, and what it can be used for in the Description field. You can add a description of up to 255 characters in length.**

9. **Click OK.**
If you did not enter a unique name for the object in the Unique Name field, Interactive Reporting prompts: “Model name exists in this group. Do you want to enter a Unique Name. In this case, click Yes.

If you click No, prompts: ”Do you want to replace the existing file?” If you click OK to replace the existing file, the saved object overwrites any existing object with the same name.

11 Click OK.

The Groups tab is displayed. Groups associated with the owned repository appear in the Groups panel. The PUBLIC groups is also included by default.

12 Use the arrow buttons to grant access to Repository groups by adding them from the Available list to the Selected Groups list.

Tip: You must move the PUBLIC group to the Groups List panel if you want to provide general, unrestricted access to the repository object.

13 Click OK to save the object to the Repository.

14 Distribute the Interactive Reporting database connection file to end-users as need to access both the object source database, and if necessary, the Interactive Reporting hub used to store the object.

Import Data File

You can import data from other applications and use the data as the source for your document. Use the Import Data File command to use data stored in Microsoft Excel, tab-delimited, or comma-delimited file formats. The data is displayed in the Results section. You can then use the imported data to build reports and perform data analysis.

To import a data file:

1 Select File, then Import Data File, and then Data File.

The Import File dialog box is displayed.

2 Navigate to the location of the file to import.

3 Select a file type from the Files of Type drop-down list to make the file easier to find:
   • Tab delimited (*.txt)
   • Comma Delimited (*.csv)
   • Excel (*.xls) (the Import feature supports style and font information and Unicode strings [Excel 2000, Excel 2003 file formats]).

   Note: Interactive Reporting does not support the import of the Office 2007 format of Excel documents. As a workaround, save the document in an older Excel format and import the document again.
   • All Files (*.*)
Note: If an Excel (.xls) file to be imported is not in table format, you receive the following error message: “Warning: Data you are trying to import is not in a tabular format. Do you still want to import it? You can either reformat the data in Excel to a table format, or continue with the import. If you continue the format, the data does not map in the Interactive Reporting document.

4 Select the file to import and click OK.

The data from the imported file is displayed as a table in the Results section.

5 Select the encoding for the file from the Encoding drop-down.

The encoding parameter sets the output file encoding. It works for the file formats that allow you to use code page encoding like TXT, CSV and XLS. XLS supports a subset of encodings.

Import SQL

The Import SQL command takes a complete SQL statement from a text file, imports it into an existing query, and retrieve the data set from the database server. Use this feature to take advantage of SQL statements you have already written.

Before importing SQL files, make sure that the following conditions are true:

- The SQL file to import begins with a SELECT statement.
- The Query section is active.
- The connection to the database is active.
- The Content pane does not contain any tables.

You also need to know the number of columns to display in the Results section.

After you import the SQL file into the Query section, you cannot:

- edit it
- drag items from the table to the Request line
- use the custom SQL feature
- display its properties

However, you can specify a user-friendly name for the Request item and identify its data type.

To import SQL files:

1 Select File, then Import Data File, and then SQL.

The Import File dialog box is displayed.

2 Navigate to the location of the file.

3 Select the file to import and click OK.

Interactive Reporting prompts you for the number of data columns. The number that is by default in the dialog box is an estimate.

4 Type the number of columns and click OK.
Interactive Reporting inserts the SQL statement directly into the content, nested between the header and footer “Imported SQL Statement.” If the statement is larger than the visible Content pane, use the scroll keys to view it.

**Exporting Data**

Interactive Reporting enables you to export data to other file formats for use with non-Interactive Reporting applications. Review these topics for information on exporting data:

- “Defining Export Properties for .HTML in the Results and Table Sections” on page 45
- “Defining Export Properties for HTML in the Pivot and OLAP Sections” on page 46
- “Defining Export Properties for HTML in the Report Section” on page 47
- “Exporting a Section” on page 47
- “Exporting to Microsoft Office HTML File Formats” on page 48
- “Exporting a Document as a Web Page” on page 61
- “Browser and HTML Restrictions and Limitations” on page 62
- “Using the Export to HTML Wizard” on page 63
- “Exporting SQL” on page 63
- “Exporting a Query Log” on page 63
- “Exporting Scripts To a Text File” on page 64

**Defining Export Properties for .HTML in the Results and Table Sections**

Before exporting data to a .HTML file, use the Export Properties dialog box to specify export properties for data in the Results and Table sections.

1. **Select Format, then Export Properties.**
   
   The Export Properties dialog box is displayed.

2. **Specify the number of vertical rows to include on an .HTML page before starting a new page (file) by checking the Vertical Page Break checkbox and entering the number of rows.**
   
   The default setting is 100 rows per .HTML page. (Select Pixels to specify the number of pixels instead of rows).
   
   To export to one file, or to have no page break, leave the Vertical Page Break checkbox blank.

3. **Select the desired export properties.**
   
   Check the Export with Style Sheet (CSS) option to create a Cascading Style Sheet file separate from the .HTML file. If this field is unchecked, the style sheet information is embedded in the .HTML file itself.

4. **Select the Export MS Excel formulas check box to export Interactive Reporting computed items to Excel.**
   
   To export a raw table (without formulas), leave the check box blank.
By default the Export MS Excel formulas feature is enabled. For more information about including formulas in a section to be exported to Excel, see “Formula Mappings” on page 55.

5 For data exported to tab-delimited text files, select the Export Without Quotes check box to exclude double quotation marks around real column/cell values in the exported files and click OK.

Defining Export Properties for HTML in the Pivot and OLAP Sections

Before exporting data to HTML, use the Export Properties dialog box to specify export properties for data in the Pivot section.

➢ To specify export properties:

1 Select Format, then Export Properties.

   The Export Properties dialog box is displayed.

2 Specify the number of vertical rows to include on an HTML page before starting a new page (file) by checking the Vertical Page Break check box and entering the number of row or pixels.

   The default setting is 100 rows per HTML page. (Select Pixels to specify the number of pixels instead of rows). To export to one file, or to have no page break, leave the Vertical Page Break check box blank.

3 Specify the number of horizontal columns to include on an HTML page before starting a new page (file) by checking the Horizontal Page Break check box and entering the number of columns or pixels.

   The default setting is 100 rows per HTML page. (Select Pixels to specify the number of pixels instead of rows). To export to one file, or to have no page break, leave the Horizontal Page Break check box blank.

4 Select the desired export properties.

   Check the Export with Style Sheet (CSS) option to create a Cascading Style Sheet file separate from the HTML file. If this field is unchecked, the style sheet information is embedded in the HTML file itself.

5 For data exported to tab-delimited text files, select the Export Without Quotes check box to exclude double quotation marks around real column/cell values in the exported files.

6 Click OK.

Defining Export Properties for HTML in the Chart Section

Before exporting a chart section to an HTML file, use the Export Properties dialog box to specify export properties for data.

➢ To specify export properties:

1 Select Format, then Export Properties.

   The Export Properties dialog box is displayed.
To match the maximum number of bars displayed (X and Z directions) for HTML renderings (static or ) with the corresponding values on the Label Axis tab of the Chart Properties dialog (that is, use the Designer settings for the chart), check the Sync with Chart Properties field.

To manually set the maximum number of bars to display per view, uncheck the Sync with Chart Properties field. When the X-Axis Maximum Bars Displayed and Z-Axis Maximum Bars displayed are active, you can set the maximum number of bars to display for both the X axis and the Z axis. The horizontal scrollbar controls scrolling of the X axis, and the Vertical scrollbar controls scrolling of the Z axis.

To render chart boundaries, click Chart Boundaries.

The default chart boundaries are 504 pixels in height and 838 pixels in width. To specify other chart boundaries pixel sizes, enter the size in the Height and Width edit boxes. To use the default sizes, click the Reset Size.

Click OK.

Defining Export Properties for HTML in the Report Section

Before exporting data to static HTML, use the Export Properties dialog box to specify export properties for data in the Report section.

To specify export properties:

1. Select Format, then Export Properties.
   The Export Properties dialog box is displayed.
2. Select the desired export properties.
   Check the Export with Style Sheet (CSS) option to create a Cascading Style Sheet file separate from the HTML file. If this field is unchecked, the style sheet information is embedded in the HTML file itself.
3. Click OK.

Exporting a Section

After processing a query, you can export the data contents of the Results, Pivot, Chart, Table, or Report sections for use in other applications. If you export from a Results section, the data is raw and unaggregated. If you export from a report section, the data is already aggregated.

In addition, you can export the contents of a Dashboard section to a JPEG file.

To export the current section to a file:

1. Select File, then Export, and then Section.
   The Export Section dialog box is displayed.
2. Specify the location where you want to save the file.
3. Type a name for the section to be exported in the File Name field.
Select a file format from the Save As Type drop-down list. Select from:

- Microsoft Office 2000 HTML (*.html)
- Microsoft Office Web Archive (*.mhtml)
- Excel (*.xls)
- Excel 2007 (*.xlsx)
- Lotus 1-2-3 (*.wks)
- Text (Tab delimited) (*.txt)
- Text (Comma Delimited) (*.csv)
- HTML (*.htm)
- PDF (*.pdf)

The available export file formats change depending on which section you export.

**Note:** Select **Format**, then **Export Properties** to set properties for files exported to text or HTML.

**Note:** You cannot export Report sections using the Excel 2007 format (*.xlsx).

### Notes on Exporting a Section to PDF

By default, headers and footers of the Results, Table, Pivot and Chart sections are mapped to the dialog box font, MS Sans. Because this font is not presented as a single file in the system, but as a series of fonts that may be used to show specific charters, the PDFLib can export only a certain amount of characters using the export mechanism for multiple page PDF export. To avoid the lack of double byte characters in the PDF export, change the header and footer font to Arial Unicode MS or Andale WT.

In a Pivot report exported to PDF using the Portrait setting, the total side label width may truncate. To remedy this issue, reduce the number of side labels to decrease the width, or change the page setup to Landscape, and then export the report again.

### Exporting to Microsoft Office HTML File Formats

You can export and deploy an Interactive Reporting section on an intranet or internet, and work back and forth between the HTML file and Microsoft Office 2000 Excel. This is achieved by using the Microsoft Office 2000 HTML (*.html) or Microsoft Office Web Archive (*.mhtml) file formats. These file formats preserve and recognize formatting attributes, formulas, and Visual Basic for Applications macros (VBA) created in the original section.

This feature uses Microsoft Office XML (extended markup language) tags in addition to the HTML. When the exported section is opened in Microsoft Office (specifically Excel), the formatting and formula from the original section are preserved and recognized. You can modify
the file again and save it as an HTML file without the loss of any HTML code. If you modify the file and save it in Office Excel file format, the HTML tags are not retained.

A section exported to a Microsoft Office 2000 HTML (.html) file format has the main web page saved in one folder and all the graphics and other related information (such as VBA macros) in another.

Microsoft Office 2000 Web Archive (.mhtml or .mhtml) refers to MS HTML, which is a standard for including objects in the same file as the HTML code (for example .gif or .jpeg files). Objects are encoded using the MIME HTML Internet standard. You might use this file format if you plan to email the HTML as a single file.

**Note:** Any changes made to the section through the HTML Wizard are not recognized when it is exported to Microsoft Office 2000 HTML.

### Types of HTML File Formats

Interactive Reporting supports three HTML file formats to which a section can be exported:

- Microsoft Office 2000 HTML (*.html)
- Microsoft Office Web Archive (*.mhtml)
- Standard HTML (*.html)

The export features are supported in the following sections:

- Tables and Results
- Chart
- Pivot
- Report

The Dashboard section cannot be exported.

To export a section in Microsoft Office file format:

1. **Select File, then Export, and then Section**
   
   The Export Section dialog box is displayed.

2. **Specify the name of the section in the File name field.**

3. **Select the HTML file format for the section to be exported in the Save as type field.**
   
   Valid HTML file formats are:
   
   - Microsoft Office 2000 HTML (*.html)
   - Microsoft Office Web Archive (*.mhtml)
   - Standard HTML (*.html)

4. **Click Save.**
### Comparison of HTML File Format Types

The following table shows which export attributes are supported by the HTML file format.

<table>
<thead>
<tr>
<th>Section</th>
<th>Office HTML Export to Office Excel</th>
<th>Standard HTML Export to Excel</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Results/Table</strong></td>
<td>Formatting is preserved.</td>
<td>Formatting is not preserved. Formulas are not supported.</td>
</tr>
<tr>
<td></td>
<td>Formulas are supported. (See Formula Mappings).</td>
<td></td>
</tr>
<tr>
<td><strong>Pivot</strong></td>
<td>Export preserves formatting.</td>
<td>Export does preserve formatting. Formulas are not supported.</td>
</tr>
<tr>
<td></td>
<td>Cell merging is supported.</td>
<td>Export does not support formulas. Cell margins are not supported.</td>
</tr>
<tr>
<td><strong>Chart</strong></td>
<td>The Chart section is written to two worksheets: the first worksheet contains the Chart data in a table, and the second worksheet contains the Chart image. Exported Chart data is aggregated in the same way as in the original Chart section. <strong>Tip:</strong> When a Chart section has been exported to Excel, you can use the data in the table as a data source and launch the Excel Chart Wizard to create the actual Chart. The rendered Chart looks the same as the one created in Interactive Reporting Studio.</td>
<td>The Chart data is written to a simple table.</td>
</tr>
<tr>
<td><strong>Report</strong></td>
<td>The Report data is written to a simple table.</td>
<td>The Report data is written to a simple table.</td>
</tr>
<tr>
<td></td>
<td>If a report has only one table or one pivot (no matter how many pictures and or text labels are in the report), column widths are preserved. Otherwise column widths are not maintained. Column width, in this case, is resized to fit data from any table. Formatting is preserved. Report elements including: text, labels, images, tables, pivots, and charts are placed one after another vertically. A vertical gap of a certain fixed height exists between two elements. If elements overlap vertically by a small value, these elements are shown with no gap between them. This is to accommodate any design that includes stacked elements. The gap is one row high. Report headers/footers are output. Graphical elements such as lines, circles are not supported. Overlapping tables are not supported and are computed consecutively. Page headers/footers are not supported.</td>
<td></td>
</tr>
<tr>
<td><strong>Dashboard</strong></td>
<td>Not supported except as an image in the export feature.</td>
<td>Not supported</td>
</tr>
<tr>
<td><strong>EPM Workspace</strong></td>
<td>Export to Office 2000 is not supported by way of Toolbar or Menu</td>
<td>Not supported by way of Toolbar or Menu</td>
</tr>
</tbody>
</table>
Exporting a Section to Microsoft Office Excel Worksheets

Each section exported to Microsoft Office HTML creates one worksheet in the Microsoft Excel workbook even when there are multiple sections, unless you use the Microsoft Excel 2007 format (*.xlsx). Exported sections of the same Interactive Reporting document do not reference each other in the Microsoft Excel worksheet. For example, if a Table section is created from a Results section, both exported sections are displayed as independent worksheets in Microsoft Excel.

Exported sections of the same Interactive Reporting document do not reference each other in the Microsoft Excel worksheet. For example, if a Table section is created from a Results section, both exported sections are displayed as independent worksheets in Microsoft Excel.

An Internet Explorer browser can show more than 65,536 rows, and an Excel spreadsheet has 256 columns and a maximum of 65,536 rows. If an exported section has more rows than the Microsoft Excel row maximum, all rows are exported to the source rows available, but only 65,536 rows are processed for Microsoft Excel to process any formulas. In this case, Microsoft Excel truncates the exported data in the section.

Number and Date Formats

Number and date formats in Interactive Reporting sections are exported based on the following predicates:

- Any text within Reporting and Analysis format strings is enclosed double-quotes (““).
- Interactive Reporting fills the number formats for positive, negative, and zero numbers to achieve compatibility with Excel’s construction. Excel number format definition consists of the above number formats, delimited with a semi-colon.
- Date/time formats are exported as is except for the am/pm format, which receives a special format.
- Thousands and decimal separators are converted to Excel special symbols based on the default Reporting and Analysis locale format. When the number format is evaluated, the decimal separator converts to "." (a period) and the thousand separator becomes “,” (a comma) in the target Excel format. Excel treats these special symbols as locale independent separators and replaces them with actual separators from the user’s locale at run-time.
- A value in an Excel cell has two aspects: the value formatted for displaying and the original value. Similarly when a value is placed in a HTML table, a special attribute x:num=”<actual value>” is used to preserve the original value. The pre-formatted value is placed in a cell of the HTML table, and it can be shown in the web browser. Formulas operate with the original values. The x:str attribute is used as a global designator in an html <table> in order to tell Excel that all the values which do not have x:num attribute specified should be treated as strings. This is useful when there is mixed string and numeric data in a string type column. The third component associated with an HTML table cell is a style class. It is used to specify the custom numeric format for a cell value. Date and time values are represented as numbers. These values are saved in the x:num attribute, and the style class definition specifies the format for date.
**Styles**

Exported sections support styles (CSS) that have been applied to the section including: font name and size, and bold/italic properties.

Overline and double overline properties are not supported in Excel, because Excel has no equivalent styles.

The simple overline style is supported in a browser; the double overline is not. In addition, the overline applies only to the text, not the whole cell as in Interactive Reporting Studio.

**Colors**

Excel supports a fifty-six color palette of which sixteen are used exclusively for Charts, and forty for cells in a spreadsheet. Colors can be defined by using RGB notation, and are shared among all worksheets in a workbook.

All colors displayed in the sections being exported are saved as a custom palette and saved in the HTML file (in a hidden xml section). By default Excel uses the color palette as a baseline for creating a custom palette. The colors that match Excel’s colors retain their positions. Unused positions are filled by default Excel colors.

Excel automatically determines which index in the color palette to use for a particular color occurring in CSS.

If the total number of colors is more than forty, Excel determines the color placement.

**Text Wrapping**

If text wrapping is disabled in Interactive Reporting Studio and the section is exported, Excel sizes the column to display all values within a row. Words that do not fit into the cell are wrapped to the next line and are not visible. In the browser, the text is truncated.

**Chart as Image**

A worksheet containing a chart image shows no cell grid.

Charts larger than a single image (chart scrolling is enabled) are placed on a single worksheet, and page breaks are provided. Each image is printed on its own page.

**Headers and Footers**

Headers and footers are converted into appropriate Excel equivalents, and are displayed in a printed document. In addition, the format template is converted to the Excel format. Multiple headers and footers are supported, but they are converted into a single multi-line header or footer.

Headers and footers are not displayed when the exported section is opened in a browser.
**Printing Improvements**

Results, Table, and Pivot sections are exported so that the table headings and pivot top and side labels are marked. These marks allow headings and labels to be duplicated on each page when they are printed from Excel.

**Suppress Duplicates for Results or Table**

The Suppress Duplicate feature hides duplicate values in a selected column but includes them in calculations. Hidden values can be seen and changed only by clicking the cell in Excel. Typing any new value does not change the hidden status of the cell unless the user changes the cell format.

**Formula Generation**

An exported Interactive Reporting document table section can have the following kinds of JavaScript expressions:

- Grand/Break Total calculation (for example, Sum(Units, Breaks) + 10)
- Grand/Break Total labels (for example, "Total "+ ToChar(Store))
- Computed Item expressions (for example, Units*2 + 10)

When exporting a table with expressions, the following steps are performed:

- Values that are calculated using formulas are exported to HTML, and available to be shown in a web browser.
- Excel formulas are generated for each cell in the table and written to HTML. This file can be opened in Excel and all values are re-calculated. An end-user can change values in the cells and results are re-calculated.

**JavaScript Expressions**

JavaScript expressions used with the Export to HTML feature can refer to columns in the table and invoke computed items. They can also refer to any JavaScript class like Math and String. Those computed items which do not have direct equivalent among the built-in Excel functions are simulated by custom VB functions. See “Formula Mappings” on page 55 for computed item to Excel built-in function mappings.

If the expression refers to a JavaScript class not supported by the Export feature, #NAME? is displayed in the Excel cell (and as a valid value in the web browser) and the Error Office html tag attribute is used. In addition, a comment is added to the cell explaining the error. In Excel, this comment is displayed near the cell where the error occurred. In the browser, the comment is displayed when you hover the mouse pointer over a cell with a small red-colored corner.
Mapping JavaScript to Excel

Interactive Reporting uses a translator to distinguish between the range types used to substitute a column reference in a computed item expression. There are three types of ranges:

- Continuous range corresponding to the whole column (for example, Units)
- Aggregate range – this is a continuous range that is broken by Break Totals (for example Units_Agg). The only purpose of this range is as a data source for aggregate functions. The translator uses Units in all kind of expressions that refer to the column, but it uses Units_Agg range for aggregate functions, for example, SUM(Units_Agg).
- Continuous sub-range used in break total computation.
- Individual cell references (for example, A1, B5)

The table below outlines how the JavaScript expression operations are mapped to Excel.

<table>
<thead>
<tr>
<th>JavaScript</th>
<th>MS Excel</th>
</tr>
</thead>
<tbody>
<tr>
<td>*, /, &lt;=, =&gt;, -, + (arithmetical plus operator)</td>
<td>Uses the same symbols.</td>
</tr>
<tr>
<td>%</td>
<td>MOD() built-in function</td>
</tr>
<tr>
<td>!=</td>
<td>&lt;&gt;</td>
</tr>
<tr>
<td>==</td>
<td>=</td>
</tr>
<tr>
<td>+ (concatenation operator)</td>
<td>&amp;</td>
</tr>
<tr>
<td>Exp1? Exp2: Exp3 (temary operator)</td>
<td>IF(Exp1, Exp2, Exp3)</td>
</tr>
<tr>
<td>Comma operator (e.g. exp1, exp2, ..., expN)</td>
<td>Replaced with the last expression. For example, (exp1, exp2, ..., expN) -&gt; is replaced with expN in resulting Excel formulas. Expressions should be of the same type (expression or string)</td>
</tr>
<tr>
<td>!</td>
<td>NOT() built-in function</td>
</tr>
<tr>
<td>&amp;&amp;</td>
<td>AND() built-in function</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Character literal (e.g. ‘a’)</td>
<td>Converts to string literal (for example, “a”)</td>
</tr>
<tr>
<td>Hexadecimal and octal number (e.g. 0x10, 010)</td>
<td>Converts to a decimal number (for example, 16, 8)</td>
</tr>
<tr>
<td>Logical constants (true and false)</td>
<td>Converts to Excel’s TRUE and FALSE</td>
</tr>
<tr>
<td>Expressions with ‘null’</td>
<td>Expressions with ‘null’ have limited support. Column values can be compared with null. Computed Items should not produce null values.</td>
</tr>
<tr>
<td>\n\t\r\a\b and \xHH in string literals. Example: “a\na”</td>
<td>Replaced with CHAR(x) in the output string. For example: “a” &amp; CHAR(10) &amp; “a”</td>
</tr>
</tbody>
</table>

The JavaScript ‘+’ operator can be applied to strings and denotes a string concatenation. Excel’s string concatenation operator is ‘&’. Interactive Reporting distinguishes between arithmetic ‘+’
and string ‘+’ by tracking the type of arguments in expressions. If one of the arguments is a string literal, a concatenation operator is used. The translator tracks the return type of Object Model functions and columns, so it can infer a final expression type. There are three supported types: numeric, string, and date.

**Formula Mappings**

Most of the computed items that comprise computed item expressions can be mapped directly to Excel functions. Certain functions have different numbers of arguments or argument order and require additional processing.

The following table shows which Interactive Reporting computed items map to Excel formulas.

<table>
<thead>
<tr>
<th>Interactive Reporting Computed Item</th>
<th>Microsoft Excel Function</th>
<th>Implementation</th>
<th>Aggregation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decode</td>
<td>H_Decode</td>
<td>VBA</td>
<td></td>
</tr>
<tr>
<td>Nvl</td>
<td>Limited support: Nvl(column, expr).</td>
<td>Excel</td>
<td></td>
</tr>
<tr>
<td>AddMonth</td>
<td>H_AddMonth</td>
<td>VBA</td>
<td></td>
</tr>
<tr>
<td>DayOfMonth</td>
<td>DAY</td>
<td>Excel</td>
<td></td>
</tr>
<tr>
<td>LastDay</td>
<td>H_LastDay</td>
<td>VBA</td>
<td></td>
</tr>
<tr>
<td>MonthsBetween</td>
<td>H_MonthsBetween</td>
<td>VBA</td>
<td></td>
</tr>
<tr>
<td>NextDay</td>
<td>H_NextDay</td>
<td>VBA</td>
<td></td>
</tr>
<tr>
<td>Sysdate</td>
<td>NOW</td>
<td>Excel</td>
<td></td>
</tr>
<tr>
<td>ToChar(value, format)</td>
<td>Limited Support: Excel’s built-in TEXT(val, format) function is used and the format argument is translated from the Reporting and Analysis format into Excel’s on the formula generation time. Consequently, the format should always be a constant string literal.</td>
<td>Excel</td>
<td></td>
</tr>
<tr>
<td>ToChar(column)</td>
<td>When the argument for ToChar is a column reference, it is converted to TEXT(column, format), where format is one of real, int, or date_format depending on the column type.</td>
<td>Excel</td>
<td></td>
</tr>
<tr>
<td>ToDate(x)</td>
<td>(DATEVALUE(x) + TIMEVALUE(x))</td>
<td>Excel</td>
<td></td>
</tr>
<tr>
<td>ToMonth</td>
<td>H_ToMonth</td>
<td>VBA</td>
<td></td>
</tr>
<tr>
<td>ToQtr</td>
<td>H_ToQtr</td>
<td>VBA</td>
<td></td>
</tr>
<tr>
<td>ToYear</td>
<td>YEAR</td>
<td>Excel</td>
<td></td>
</tr>
<tr>
<td>Abs</td>
<td>ABS</td>
<td>Excel</td>
<td></td>
</tr>
<tr>
<td>Atn</td>
<td>ATAN</td>
<td>Excel</td>
<td></td>
</tr>
<tr>
<td>Interactive Reporting Computed Item</td>
<td>Microsoft Excel Function</td>
<td>Implementation</td>
<td>Aggregation</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>--------------------------</td>
<td>----------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Cell</td>
<td>H_Ceil</td>
<td>VBA</td>
<td></td>
</tr>
<tr>
<td>Cos</td>
<td>COS</td>
<td>Excel</td>
<td></td>
</tr>
<tr>
<td>Cosh</td>
<td>COSH</td>
<td>Excel</td>
<td></td>
</tr>
<tr>
<td>Exp</td>
<td>EXP</td>
<td>Excel</td>
<td></td>
</tr>
<tr>
<td>Floor</td>
<td>H_Floor</td>
<td>Excel</td>
<td></td>
</tr>
<tr>
<td>Ln</td>
<td>LN</td>
<td>Excel</td>
<td></td>
</tr>
<tr>
<td>Log</td>
<td>LOG</td>
<td>Excel</td>
<td></td>
</tr>
<tr>
<td>Log10</td>
<td>LOG10</td>
<td>Excel</td>
<td></td>
</tr>
<tr>
<td>Max</td>
<td>MAX</td>
<td>Excel</td>
<td></td>
</tr>
<tr>
<td>Min</td>
<td>MIN</td>
<td>Excel</td>
<td></td>
</tr>
<tr>
<td>Mod</td>
<td>H_MOD</td>
<td>VBA</td>
<td></td>
</tr>
<tr>
<td>Power</td>
<td>POWER</td>
<td>Excel</td>
<td></td>
</tr>
<tr>
<td>Round</td>
<td>ROUND</td>
<td>Excel</td>
<td></td>
</tr>
<tr>
<td>Sign</td>
<td>SIGN</td>
<td>Excel</td>
<td></td>
</tr>
<tr>
<td>Sin</td>
<td>SIN</td>
<td>Excel</td>
<td></td>
</tr>
<tr>
<td>Sinh</td>
<td>SINH</td>
<td>Excel</td>
<td></td>
</tr>
<tr>
<td>Tan</td>
<td>TAN</td>
<td>Excel</td>
<td></td>
</tr>
<tr>
<td>Tanh</td>
<td>TANH</td>
<td>Excel</td>
<td></td>
</tr>
<tr>
<td>Trunc</td>
<td>TRUNC</td>
<td>Excel</td>
<td></td>
</tr>
<tr>
<td>Avg</td>
<td>H_Avg</td>
<td>VBA</td>
<td>*</td>
</tr>
<tr>
<td>AvgNonNull</td>
<td>AVERAGE</td>
<td>Excel</td>
<td>*</td>
</tr>
<tr>
<td>Chr</td>
<td>H_Chr</td>
<td>VBA</td>
<td></td>
</tr>
<tr>
<td>ColMax</td>
<td>H_ColMax</td>
<td>VBA</td>
<td>*</td>
</tr>
<tr>
<td>ColMin</td>
<td>H_ColMin</td>
<td>VBA</td>
<td>*</td>
</tr>
<tr>
<td>Count</td>
<td>H_Count</td>
<td>VBA</td>
<td>*</td>
</tr>
<tr>
<td>CountDistinct</td>
<td>H_CountDistinct</td>
<td>VBA/Excel</td>
<td>*</td>
</tr>
</tbody>
</table>

Limited Support: Can be used in a break or grand totals only. There may be performance considerations associated with this computed item because it uses heavy computations.
<table>
<thead>
<tr>
<th>Interactive Reporting Computed Item</th>
<th>Microsoft Excel Function</th>
<th>Implementation</th>
<th>Aggregation</th>
</tr>
</thead>
<tbody>
<tr>
<td>CountNonNull</td>
<td>COUNTA</td>
<td>Excel</td>
<td>*</td>
</tr>
<tr>
<td>CountNull</td>
<td>H_CountNull</td>
<td>VBA</td>
<td>*</td>
</tr>
<tr>
<td>Cume</td>
<td>Computed Items with expressions that have exactly one call to Cume() and nothing else. Example: “Cume(Units)”</td>
<td>Excel/VBA</td>
<td>Aggregation and cross references in rows</td>
</tr>
<tr>
<td>Next</td>
<td>Function call is converted to appropriate cell reference. Limited support: Cannot be used in a Break or Grand Total expression.</td>
<td>Excel/VBA</td>
<td>Cross references in rows</td>
</tr>
<tr>
<td>Prior</td>
<td>Function call is converted to appropriate cell reference. Limited support: Cannot be used in a break or grand total expression.</td>
<td>Excel</td>
<td>Cross references in rows</td>
</tr>
<tr>
<td>Sum</td>
<td>Use Excel’s SUM() for non-string columns. String columns exception: H_SumStr() is used. Can be used in break and grand totals only. There may be performance considerations associated with this computed item because it uses heavy computations.</td>
<td>Excel/VBA</td>
<td>*</td>
</tr>
<tr>
<td>Median</td>
<td>MEDIAN</td>
<td>Excel</td>
<td>*</td>
</tr>
<tr>
<td>Mode</td>
<td>H_Mode</td>
<td>VBA</td>
<td>*</td>
</tr>
<tr>
<td>Percentile</td>
<td>H_Percentile</td>
<td>VBA</td>
<td>*</td>
</tr>
<tr>
<td>Rank(column)</td>
<td>H-Rank</td>
<td>VBA</td>
<td>*</td>
</tr>
<tr>
<td>RankAsc(column)</td>
<td>H-RankAsc</td>
<td>VBA</td>
<td>*</td>
</tr>
<tr>
<td>StdDev</td>
<td>H-StdDev</td>
<td>VBA</td>
<td>*</td>
</tr>
<tr>
<td>StdDevp</td>
<td>H_StdDevp</td>
<td>VBA</td>
<td>*</td>
</tr>
<tr>
<td>Var</td>
<td>Var</td>
<td>VBA</td>
<td>*</td>
</tr>
<tr>
<td>Varp</td>
<td>Varp</td>
<td>VBA</td>
<td>*</td>
</tr>
<tr>
<td>Interactive Reporting Computed Item</td>
<td>Microsoft Excel Function</td>
<td>Implementation</td>
<td>Aggregation</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>--------------------------</td>
<td>----------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Ascil</td>
<td>CODE</td>
<td>Excel</td>
<td></td>
</tr>
<tr>
<td>Concat</td>
<td>CONCATENATE</td>
<td>Excel</td>
<td></td>
</tr>
<tr>
<td>Initcap</td>
<td>H_Initcap</td>
<td>VBA</td>
<td></td>
</tr>
<tr>
<td>Instr</td>
<td>H_Instr</td>
<td>VBA</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>LEN</td>
<td>Excel</td>
<td></td>
</tr>
<tr>
<td>Lower</td>
<td>LOWER</td>
<td>Excel</td>
<td></td>
</tr>
<tr>
<td>Ltrim</td>
<td>H_Ltrim</td>
<td>VBA</td>
<td></td>
</tr>
<tr>
<td>Replace</td>
<td>H_Replace</td>
<td>VBA</td>
<td></td>
</tr>
<tr>
<td>Rtrim</td>
<td>H_Rtrim</td>
<td>VBA</td>
<td></td>
</tr>
<tr>
<td>Substr</td>
<td>MID</td>
<td>Excel</td>
<td></td>
</tr>
<tr>
<td>Translate</td>
<td>H_Translate</td>
<td>VBA</td>
<td></td>
</tr>
<tr>
<td>Upper</td>
<td>UPPER</td>
<td>Excel</td>
<td></td>
</tr>
<tr>
<td>MovingAvg</td>
<td>H_MovingAvg</td>
<td>VBA</td>
<td>Cross references in rows</td>
</tr>
<tr>
<td>MovingDiff</td>
<td>H_MovingDiff</td>
<td>VBA</td>
<td>Cross references in rows</td>
</tr>
<tr>
<td>MovingMax</td>
<td>H_MovingMax</td>
<td>VBA</td>
<td>Cross references in rows</td>
</tr>
<tr>
<td>MovingMed</td>
<td>H_MovingMed</td>
<td>VBA</td>
<td>Cross references in rows</td>
</tr>
<tr>
<td>MovingMin</td>
<td>H_MovingMin</td>
<td>VBA</td>
<td>Cross references in rows</td>
</tr>
<tr>
<td>MovingSum</td>
<td>H_MovingSum</td>
<td>VBA</td>
<td>Cross references in rows</td>
</tr>
<tr>
<td><strong>All aggregate functions with two and three arguments:</strong></td>
<td>When two argument functions are used in break or grand total expressions and the second argument is ‘Breaks’, most of the functions are converted to their built-in Excel equivalents. Otherwise, the function is substituted with the appropriate H_xxx() VBA function.</td>
<td>Excel/VBA</td>
<td></td>
</tr>
</tbody>
</table>

### Custom Formulas

Interactive Reporting has a number of computed items that do not map directly to an equivalent Excel function. This section describes the methods the Export feature uses to construct a mapping from the computed item to an Excel function.
Adding VBA functions to the Exported Excel File

Interactive Reporting helper functions implemented in Visual Basic are saved to the exported file. In this instance, the Excel file becomes self-contained. It can be passed to any user as is and does not require installing additional components such as Excel add-ins. All the functions are implemented within this file.

Since VBA functions are treated in Excel as macros, Interactive Reporting signs them digitally in order to prevent a security warning dialog box in Excel if the macro security level is medium or high. When the macros are signed, you can observe certification information and choose to trust the sources. Once the certificate is marked as trusted, all subsequent openings of exported Excel files proceed silently.

Unsupported Formulas

The Excel formula content limit length is 1,024 characters.

Null Values

In Interactive Reporting, a null value is a cell absent of data, but which does not equal zero. Cells that are absent of data are represented by blank cells. Excel does not recognize null values, but has a similar concept. There are also built-in aggregation functions analogs which understand this concept.

Custom Cell Ranges

Named cell ranges are generated to construct Excel formulas for computed items in the spreadsheet. For example, if there is a Table section with Break Totals, then the following types of named ranges are generated:

- A continuous cell range that includes the whole column For example: Units = Results!$C$2:$C$998. This range can be used in all kinds of formula expressions, but cannot be passed to aggregate functions like SUM.

- A compound or broken cell range that includes all the cells with actual data excluding those occupied by Total information For example: Units_Agg = Results!$C$2:$C$4,Results!$C$6:$C$34,Results!$C$36:$C$164. This range can be passed to aggregate functions, which is reflected in its the name.

- A cell range used in break totals calculations For example: Sr_32_2=Results!$C$2:$C$4

Below is a sample spreadsheet generated from the Results section. The column Computed of the section contained Units*2 expression, the column Computed2 contained SUM(Units) expression. The generated Excel formulas are shown in the cells.
For the Excel user, the Auto Outlining feature can be used to create row groups from the Break Totals that Interactive Reporting creates. Excel analyzes the formulas and creates the row groups and outline automatically. The user can then expand or collapse individual groups of rows.

Named cell ranges created in Interactive Reporting can be used for in-sheet formulas and in an external Excel file (worksheet linking). For example, an end user can create a new Excel file and enter a cell formula that refers to a data column of the workbook file exported from Interactive Reporting. For example, an end-user might enter: =SUM(BIExport.xls!Results_Units_Agg), where Results_Units_Agg is an auto-generated range that denotes the Units column of the Results table in the BIExport.xls file.

### Worksheet Names and Range Names Generation

Worksheet names are based on the Interactive Reporting BQY filename. Before exporting a section, consider the following limitations that apply to worksheet names:

- The name of the exported BQY file cannot exceed thirty-one characters.
- The symbols / \ ? * are not allowed and are removed.
- Duplicate worksheet names are appended with a number to ensure the uniqueness of each worksheet.
- A worksheet name can include space symbols. When such a worksheet name is used in formulas, it should be surrounded with single quotation marks.

Generated range names that correspond to Interactive Reporting document table columns are based on a respective column name. The following restrictions apply to generated range names:

- The worksheet name is added to the beginning of the range name to ensure its uniqueness within the whole workbook.
- Spaces within the name are replaced by underscores.
- The symbols ~!@#$%^&*()_+|{}:"<>?`-=\\;\',./ are not allowed and are removed.
- Duplicate range names are appended with a number to ensure the uniqueness of each range name.
Columns must have appropriate identifiers in JavaScript, for instance, an Order Date column can be referred to as Order_Date in a computed item expression. This identifier is used as the main part of a generated range name.

Excel has a limitation on the length of the expression used as a named range. The total length of the references (not including worksheet names) cannot be more than 255 characters. When a named range is generated, its content is broken into smaller chunks, each comprising a smaller named range. The final range is a union of these smaller ranges.

**Exporting to Excel Unicode Support**

When a section is exported to Excel, by default Interactive Reporting uses the Excel97 format (BIFF8) which supports Unicode (UTF-16LE encoding). In this case no encoding selection is necessary when you export a section to Excel on the Export Section dialog box. (The Encoding drop-down on the dialog box is disabled.)

If you want to continue using the Excel2 format (BIFF2), you must enable the Excel2 format manually on the General tab of the Program Options dialog box, and select an encoding when you export a section to Excel on the Export Section dialog box.

You set the default Excel format on the General tab of the Program Options dialog box. See “Selecting Program Options” on page 74.

**Exporting a Document as a Web Page**

Use the Export Document As Web Page command to publish the contents of Interactive Reporting documents as web pages. You can select which sections of the current document to include in the export set.

Documents sections are exported in the same order as they are displayed in the Interactive Reporting document. Exportable document sections include: Results, Table, Pivot, Chart, OLAPQuery, and reports created with the Report Designer. Imported sections that have been added to the document, such as text files and Excel files, can also be exported. Data models and the Query, and Dashboard sections are not exportable. Neither are empty or blank sections. After these files are posted to a Web site, you can access the files individually or use the main HTML file to view the frameset, which lists the sections in the report. The exported Web pages include navigation buttons so you can scroll back and forth between pages, or jump to the beginning or end of a document. In addition, the current page number and total number of pages are included in the report.

**Note:** The Export As HTML and HTML Wizard options in prior versions of Interactive Reporting remain unchanged. However, we recommend that you use the Export As Web Page option.

To export Interactive Reporting documents as Web pages:

1. Select File, then Export, and then Document as Web Page.
The Export Document As Web Page dialog box is displayed.

2 **Select the sections of the document to be exported and click OK.**

To select all sections, click Select All. To clear all selected sections, click Deselect All.

The Save As Type dialog box is displayed.

3 **Navigate to the location where you want to save your Web page.**

   **Tip:** Create a separate folder to store the files created by this export option.

4 **Type a name for the exported Web page in the File Name field, or accept the default name, for example Sample1.htm.**

   The Save As Type field is set to *.htm by default. All of the HTML files that the export process creates begin with the name specified in the File Name field, for example, Sample1Chart.htm, Sample1Pivot.htm, and so on. The HTML file with the exact same name as the name specified in the File Name field is the main HTML page, and it contains the frameset and links to all of the other pages, for example Sample1.htm.

5 **Click Save.**

To view the exported selection, open your Web browser, select **File**, then **Open**, browse to the location of the exported files, and open the main HTML page.

---

**Saving an Excel file to HTML format**

If you plan to import an HTML file that was created from Excel, note:

- Prior to saving an Excel file to HTML format, turn off Excel’s password protect feature.
- Formulas are saved as sheet references in Excel. When an Excel workbook is saved to HTML workbook, formulas linked to a cell in the HTML workbook do not operate.
- Custom view and scenarios can only be used in Excel.
- Add-ins and templates are only available in Excel.

**Browser and HTML Restrictions and Limitations**

Browser and HTML restrictions may affect how graphics and formatting are displayed on Web pages created by exporting document sections. The known restrictions and limitations are:

- Diagonal lines, ovals, round rectangles, and dotted or dashed lines do not export to the Web page. Overline or double-overline text is displayed as regular text.
- The Netscape browser shows data formatted with bold Arial 8 pt. as regular Arial 8 pt.
- Data formatted with Arial 14 pt. is displayed smaller in Interactive Reporting and Netscape than in Microsoft Word and Internet Explorer.
- Border properties (including color properties) are supported in Microsoft Internet Explorer, but not in Netscape. Consequently, border properties for the Results, Table, and Pivot sections are not displayed on Web pages opened with Netscape. If you embed a Results, Table, or Pivot section in a report, Netscape does recognize the border properties.
Raised and sunken borders are displayed as regular borders.

Lines and rectangles are not displayed in reports in UNIX browsers.

The right border of a table embedded in a report sometimes is displayed thicker in Netscape.

The Picture Tile property is not supported by HTML.

Word-wrapped fields contain hard-coded leading spaces for left padding. HTML permits the browser to implement word-wrap, but eliminates extraneous space. In addition, a browser word-wrap feature does not break a word in the middle of a word regardless of its length.

When empty table cells are displayed with a certain font, it is because Interactive Reporting inserts a single blank into each empty cell. HTML requires a single value in each empty cell.

HTML does not recognize vertical text.

**Using the Export to HTML Wizard**

The Export to HTML Wizard helps you create Web pages from charts, reports, and pivot tables.

*Note:* We suggest you use the Export Document As Web Page command (see “Exporting a Document as a Web Page” on page 61) rather than the HTML Wizard.

To use the Export to HTML Wizard, select **File**, then **Export**, and then **HTML Wizard**.

**Exporting SQL**

Use the Export SQL command to export the SQL statement for your query. The file is saved in an SQL format.

To export SQL:

1. **Select File, then Export, and then SQL.**
   
   The Export SQL File dialog box is displayed.

2. **Specify the file name and location and click Save.**

**Exporting a Query Log**

When you process a query, Interactive Reporting translates your request into SQL or a multidimensional database query statement and forwards it to the database server.

To save the contents of the SQL log to a text file:

1. **Select File, then Export, and then Query Log.**
   
   The Export Query Log dialog box is displayed.

2. **Specify the file name and location, and click Save.**
Exporting Scripts To a Text File

Use the Export Scripts To Text File command to export JavaScript code and associated events contained in an Interactive Reporting document to a text file (.TXT). Interactive Reporting categorizes the text file by object name and events, and includes document and custom menu item scripts.

➢ To export a script to a text file:

1. Select File, then Export, and then Script to Text File.
   The Export Script dialog box is displayed.

2. Specify the file name and location, and click Save.

Exporting a Dashboard Section

To be documented.

Printing Documents

Printing functions are available for most document sections. You can specify the page setup for your printer, print directly to a printer, or preview a print job on-screen.

Page Setup

Use the Page Setup command to specify the default printer for the Interactive Reporting document and to define default page properties.

➢ To define page setup information, select File, then Page Setup.

Print Preview

Use the Print Preview command to view on-screen a representation of the printed version of a finished report. The Print Preview command is available for all sections except the Dashboard. In the Report Designer section, the page view of the report is a direct representation of the printed report.

➢ To preview a section before sending it to a printer, select File, then Print Preview.

A preview of the current section is displayed in the Content pane, and a Print Preview menu is added to the Main menu. Use the commands on the Print Preview menu to navigate through the preview pages and to specify starting page numbers.

Print

Use the Print command to print the information displayed in the Content pane of most sections.
To print the information in the document, select **File**, then **Print**.

## Sending Documents

You can attach documents to your electronic mail (email) if you have a MAPI-compliant email system such as Microsoft Mail or Microsoft Exchange.

To email a document:

1. Select **File**, then **Send**.
   
The Mail Document window is displayed.

2. Select the name of the document recipient.
   
   You can send the document with or without the results. Sending an Interactive Reporting document with the results enables the recipient to do further analysis.

   A document sent without the results contains snapshots of the Chart, Pivot, and other reporting sections, but not reports created in the Report section. No further analysis is possible.

3. Copy other recipients or include additional remarks and click **Send It**.

## Quitting Interactive Reporting Studio

If documents remain open, you are prompted to save changes to each document before it shuts down.

To end Interactive Reporting Studio, select **File**, then **Exit**.

## Quitting Interactive Reporting Web Client

If documents remain open, Interactive Reporting Web Client prompts you to save changes to each document before closing it.

To end your work session in Interactive Reporting Web Client, select **File**, then **Close**.

## Using Edit Commands

The Edit menu contains standard editing commands. It also contains commands that allow you to work with document sections.
Changing Views

The View menu allows you to toggle the display of interface elements, such as panes, toolbars, and so on. It also provides commands for working with sections.

Inserting Sections and Breaks

Use the Insert menu to insert new sections in the Interactive Reporting document. You can also insert page headers and footers for use when printing certain sections.

Note: See “Formatting Numeric Data Types” on page 66 for detailed information on adding document sections and customizing the headers and footers in the document sections.

Working with the Tools Menu

Use the options on the Tools menu to:

- Process queries
- Specify connection information
- Launch Dashboard Studio or Dashboard Architect
- Define program and default font options
- Customize menus
- Launch the Resource Manager

Formatting Numeric Data Types

Formatting numeric data types include:

- “Changing Numeric Formatting” on page 66
- “Displaying Numbers in Scientific Notation” on page 67

Changing Numeric Formatting

You can change the formatting properties of numeric data types (real and integer) in the following ways:

- Select an object, right-click to access the Number shortcut menu, and select a numeric formatting option from the menu.
- Use the buttons on the right-hand side of the Formatting toolbar.
Displaying Numbers in Scientific Notation

Interactive Reporting enables you to display numbers in scientific notation. The default scientific notation format is:

0.00E+000

The scientific notation format is displayed using the appropriate decimal separator for the current local. (The above example uses a period as the decimal separator.)

The following table describes the acceptable variations on the default scientific notation format:

<table>
<thead>
<tr>
<th>Variation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>The number of decimal positions after the decimal separator.</td>
<td>For example, 0.000000E+000 displays six digits of precision.</td>
</tr>
<tr>
<td>#</td>
<td>Used after the decimal point to suppress zeros.</td>
</tr>
<tr>
<td></td>
<td>For example, the value .000179 with the format 0.000000E-00 displays as 1.79E000E-04. With the format 0.0###E-00, the same value displays as 1.79E-04.</td>
</tr>
<tr>
<td>E or e</td>
<td>Controls the case in which the exponent designator displays.</td>
</tr>
<tr>
<td>E+0 E+0 E-0 (and the lower case equivalents)</td>
<td>If the plus sign is used, the exponential component always displays with a sign (plus or minus).</td>
</tr>
<tr>
<td></td>
<td>If the minus sign or no sign is used, the sign displays only when negative.</td>
</tr>
<tr>
<td>E+0 to E+000</td>
<td>The number of zeros following the E or e, with or without a sign character, is the minimum number of digits to display for the exponent. For example, the value 179 displays as 1.79E2 with the format 0.00E0 and as 1.79e002 with the format 0.00e000.</td>
</tr>
<tr>
<td></td>
<td>If the minimum number of digits is not adequate to represent the magnitude of the number, additional digits are added as required. For example, the value 1,789,000,000,000 with the format 0.00E0 displays as 1.79E12.</td>
</tr>
<tr>
<td>Leading + or – sign</td>
<td>A leading minus sign (or no sign character) displays a sign only if the precision of the number is negative. The appropriate sign always displays if the plus character leads the string.</td>
</tr>
</tbody>
</table>

Note: You cannot apply scientific notation to non-numeric data types. If you try to do so, the formatting remains the same.

Numeric formatting buttons on the Formatting toolbar are shown in below:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
</table>
### Working with Document Sections

Working with document sections involves the tasks described in these topics:

- “Understanding Document Sections” on page 68
- “Adding Sections” on page 69
- “Viewing Sections” on page 69
- “Moving Between Sections” on page 70
- “Duplicating Sections” on page 70
- “Renaming Sections” on page 70
- “Adding Headers and Footers to Sections” on page 71
- “Deleting Sections” on page 71

### Understanding Document Sections

Documents are divided into multiple sections, each of which governs one step of the query and reporting procedure. You create sections progressively as you query a database, retrieve results, and generate reports.
A document usually includes Query and Results sections. From the Results section, you can create multiple Pivot, Chart, Table and Report sections to analyze and present data. You can also create Dashboard sections, which provide an automated push-button interface to a document.

Each section occupies an independent window and performs distinct operations. You can move back and forth between sections at any time to rebuild your query or alter your result data. You can also position sections side-by-side in multiple windows.

**Note:** It is recommended that an Interactive Reporting document file (.bqy) have no more than two hundred sections to ensure smooth performance.

To see a graphical representation of a document section, click the desired section name in the Section pane.

**Adding Sections**

- To insert a new section in a document, select **Insert**, then **New Section**.

For example, to insert a new Chart, select **Insert**, then **Chart**, to insert a new Table, select **Insert**, then **Table**.

Interactive Reporting inserts the new section and adds a new section label to the Section pane. The section label is based on the type of section added. A sequence number is added to the section label if a section with the same name already exists.

**Viewing Sections**

You can hide sections to simplify your view of the work area. This allows you to concentrate on specific sections.

- To hide a section:
  1. In the **Section pane**, select the section to hide.
  2. Select **View**, then **Hide Section**.

- To view a hidden section:
  1. Select **View**, then **Unhide Section**.
     
     The Unhide Sections dialog box is displayed.
  2. Select the **hidden section** to view and click **OK**.
     
     The section is displayed in the Section pane.
Moving Between Sections

Although each section occupies an independent window and performs distinct operations, you can move back and forth between sections at any time to rebuild your query or alter your results data. You can also position sections side-by-side in multiple windows.

You can easily navigate between sections to work on queries, results, and reports.

To move between sections, use one of the following options:

- Select the desired section in the Section pane.
- Click the arrow keys on the Section title bar.
- Select View, then Go To Section, and then Section.

Duplicating Sections

You can copy Query, OLAPQuery, Chart, Pivot and Dashboard sections if the Duplicatable feature is selected.

To make a section duplicatable, select the section label in the Section pane and select Edit, then Duplicatable.

To duplicate a section, select the section label in the Section pane and select Edit, then Duplicate Section.

Interactive Reporting duplicates the section and adds a new section label to the Section pane. The new section label is based on the original section label, but a sequence number is appended to the label. For example, if you duplicate a section named SalesChart three times, the Section pane would show: SalesChart, SalesChart2, SalesChart3, and SalesChart4.

Renaming Sections

The first section that you create is given the default section name, for example, Query or Results. When you insert new sections of the same type as those that already exist, they are numbered sequentially, for example, Query2, Results2, and so on. To assign sections different or unique names based on your application, use the Rename command.

To rename a section:

1. In the Section pane, select the section label.
2. Select Edit, then Rename.
   
   You can also click Rename Section on the shortcut menu, or double-click the desired section.
   
   The Section Label dialog box is displayed.
3. Type a new name in the Label field and click OK.
Adding Headers and Footers to Sections

You can add custom headers and footers and page numbers to your printed section.

➤ To add a header or footer:

1. Select File, then Print Preview.
   A preview of the current section is displayed in the Content pane.

2. Select Insert, then Page Header (or Page Footer).
   The Edit Header (or Edit Footer) dialog box is displayed.

3. Enter the desired text or use the buttons in the dialog box to add current date, time, file name, page, page total, or limit values and click OK.
   The new header or footer is added to your report.
   You can change the font properties and alignment of headers and footers, but you cannot add color.

Editing a Header or Footer

➤ To edit a header or footer, double-click the header or footer to edit, make any desired changes, and click OK.

Deleting Sections

You can delete a section, but do so with care. Some sections are dependent on other sections. Deleting one section could also delete one or more sections that you did not want to delete. Note that you cannot restore a deleted section.

➤ To delete a section:

1. In the Section pane, select the section label.

2. Select Edit, then Delete Section.
   You can also select the section and click Delete Section on the shortcut menu.
   The Remove Section dialog box is displayed.

3. Click Remove.

Setting Interactive Reporting Options

The Tools menu enables you to manage various options, such as job processing options, connections, default formats, and program options.

Review the following topics for information on:

• Specifying Default Formats
Specifying Default Formats

This section explains how to set up and change the way fonts, styles, numbers, currency values, and dates are displayed. Default formats that you can set include:

- Default Fonts and Styles
- Default Number Formats

Default Fonts and Styles

You can control the way fonts are displayed in every section (except Results and Dashboard) by applying default font and formatting styles to each section element.

To change default fonts and text settings:

1. Select Tools, then Options, and then Default Formats.
   - The Default Fonts and Styles dialog box is displayed, with tabs that contain specific font settings for each section.
2. Click the tab for the section to change.
3. Select the font, size, style and alignment settings and click OK.
   - To restore the default settings, click Defaults.

Default Number Formats

You can change the way numbers, currency values, and dates are displayed, or you can create new custom formats. Use the Numbers tab of the Default Fonts and Styles dialog box to specify default settings for number formats.

Table 5  Default Number Formats

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select a Formatting Locale</td>
<td>Sets the locale or country associated with the default format to use. The locale determines the available number, date, and currency formats.</td>
</tr>
<tr>
<td>Date</td>
<td>Sets the default date format.</td>
</tr>
<tr>
<td>Timestamp</td>
<td>Sets the default time and date format for the timestamp. The timestamp is a set of characters in sequential order which show the date and time on which an event occurred. This information is used for tracking events.</td>
</tr>
<tr>
<td>Time</td>
<td>Sets the default time format.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Month (For “Add Date Groups”)</strong></td>
<td>Sets the default month format for the month used in the Add Date Groups.</td>
</tr>
<tr>
<td><strong>Null</strong></td>
<td>Sets the default format for null values. Null values are empty values for which no data exists. Null values are not equal to zero.</td>
</tr>
<tr>
<td><strong>Real</strong></td>
<td>Sets the default format for real values.</td>
</tr>
<tr>
<td><strong>Integer</strong></td>
<td>Sets the default format for integer values.</td>
</tr>
</tbody>
</table>

Table 6 lists the numeric field options and definitions along with examples.

**Table 6  Number Field Descriptions**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Integer placeholder or zero value. If a number has an integer value in this position relative to the decimal point, the integer is displayed. Otherwise, a zero is displayed.</td>
<td>Apply 0 to show 123&lt;br&gt;Apply 0.00 to show 123.45</td>
</tr>
<tr>
<td>#</td>
<td>Integer placeholder. If a number has an integer value in this position relative to the decimal point, the integer is displayed. Otherwise, nothing is displayed.</td>
<td>Apply #,##0 to show 1,234</td>
</tr>
<tr>
<td>( )</td>
<td>Formats with parenthesis options display negative values in parentheses. Otherwise, negative values display with a minus sign.</td>
<td>Apply (#,##0) to show (1,234)</td>
</tr>
<tr>
<td>;</td>
<td>A semicolon operates as a separator between two number formats. The semicolon separates a positive integer and a negative integer.</td>
<td>Apply #,##0;(#,##0) to show 1, 234 or apply (1, 234) for a negative number</td>
</tr>
<tr>
<td>$%</td>
<td>Adds the respective character to numeric values in the same position relative to the decimal point.</td>
<td>Apply $#,##0.00 to show $1,234.56&lt;br&gt;Apply 0% to show 3%</td>
</tr>
<tr>
<td>m d yy</td>
<td>Displays month, day, and year in respective positions for date-coded information.</td>
<td>Apply mm dd yy to show 05 07 99</td>
</tr>
<tr>
<td>/</td>
<td>Adds the respective character to date-coded values in the same position relative to variables.</td>
<td>Apply mm/dd/yy to show 06/23/99</td>
</tr>
<tr>
<td>HH MM SS</td>
<td>Displays hour, minute, and second in respective positions for date-coded information.</td>
<td>Apply HH:MM:SS to show 17:45:10</td>
</tr>
<tr>
<td>AM PM</td>
<td>Adds the respective character to time-coded values in the same position relative to variables.</td>
<td>Apply HH:MM:SS AM to show 17:45:10 AM</td>
</tr>
</tbody>
</table>

➢ To change a default number format:

1 Select **Tools**, then **Options**, and then **Defaults Formats**.
2 Select the **Numbers** tab.
3 Make the desired selections and click **OK**.
Selecting Program Options

This section describes how to set up default file locations, enable and disable specific operating functions, administer document features, define drill-down paths, and set OLAP options:

- General Options
- File Locations
- OLAP Options

General Options

Use the General tab to globally enable or disable specific operating functions.

To select general options:

1. Select Tools, then Options, and then Program Options.
2. Select the General tab on the Options dialog box.
3. Define the desired options and click OK.

Table 7 General Options and their Descriptions

<table>
<thead>
<tr>
<th>General Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto Logon</td>
<td>Maintains a connection whenever you create a new document. If you are currently logged on, Interactive Reporting prompts you to use the current connection.</td>
</tr>
<tr>
<td>Reset Print Properties</td>
<td>Retains the print settings with each section of the document, instead of inheriting the current default print settings.</td>
</tr>
<tr>
<td>Convert BMP to PNG</td>
<td>Converts a picture automatically in .BMP format is converted to .PNG format (Portable Network Graphic) before it is loaded into the Resource Manager. The converted .PNG image are not backward compatible in pre 9.3 release documents. When an Interactive Reporting document is created and saved using 9.3.x releases, all images are saved in a centralized format (the Resource Manager). As a result, the images cannot be read or displayed in prior releases of Interactive Reporting. This is because the relocation and rationalization of the images is not understood by releases prior to 9.3.x. The exception is images from a document saved in a pre 9.3 Release, which have been referenced once: not copied, merged or duplicated, e.g. placed in Dashboard or Report workspace as a Picture graphic in traditional fashion. To retain compatibility with earlier releases, you can run the RevertImageResources script. This script provides a facility to undo the relocation and image merging, and thereby return the Interactive Reporting document to the pre-9.3 pre-Resource Manager format. This script is included with the 9.3. Release of Dashboard Development Services. It can also be downloaded from the Hyperion Developer Network. This script can only be used on the desktop, and not in the EPM Workspace, as it relies on the COM feature of Interactive Reporting Studio. Certain new features make documents ineligible for conversion to the non-Resource Manager format: for example if the 9.3 Interactive Reporting document has Bubble and Scatter charts, then these are lost when the Interactive Reporting document is opened in the older version of Interactive Reporting Studio. For information on running the RevertImageResources script, see the 9.3. Readme.</td>
</tr>
</tbody>
</table>
### Table 8  Document Save Options and their Descriptions

<table>
<thead>
<tr>
<th>Document Save Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Compress All Documents</strong></td>
<td>Saves all documents in compressed file format. This reverses the default setting, which saves documents without compression. If enabled, you can override this privilege and save documents without compression by choosing File, then Save As and changing the Save As type.</td>
</tr>
<tr>
<td><strong>Create New Documents Compressed</strong></td>
<td>Compresses new documents.</td>
</tr>
<tr>
<td><strong>Encrypt Dashboard scripts for new document</strong></td>
<td>Enables the default encryption state for a new Interactive Reporting document. When an Interactive Reporting document is created, the default encryption state matches the one on the Designer Options dialog. This setting does not affect the encryption state of any other Interactive Reporting document (including the active document.) Note that encrypted scripts are not functional in a release prior to 6.6.3 or 8.1. Despite the default setting you select, scripts are only encrypted if the Interactive Reporting document has a valid Design mode password.</td>
</tr>
</tbody>
</table>

### Table 9  Date Handling Options

<table>
<thead>
<tr>
<th>Date Handling Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>When A Two Digit Year Is Entered, Interpret As A Year Between</strong></td>
<td>By default, if you enter a date and type only two digits for the year, Interactive Reporting handles the dates as follows:</td>
</tr>
<tr>
<td></td>
<td>- Two-digit years entered from 00 up to and including 29 are assigned to the 21st century (2000 to 2029). For example, if you enter 3/12/18, Interactive Reporting accepts the date as March 12, 2018.</td>
</tr>
<tr>
<td></td>
<td>- Two-digit years entered from 30 up to and including 99 are assigned to the 20th century (1930 to 1999). For example, if you enter 3/12/06, Interactive Reporting accepts the date as March 12, 2006.</td>
</tr>
<tr>
<td>You can change the default century to which a two-digit year is assigned by using the date-handling boxes. These boxes require a range of dates within a 99-year time period. Changes to a date format are applied globally, but do not affect dates previously formatted For example, if you want the two-digit year 25 to be assigned to the twentieth century instead of the twenty-first century use the arrow keys to scroll to the year 1999. The date in the corresponding read-only, date-handling box is automatically changed to 1900. <strong>Tip:</strong> Whenever possible, enter the year as four digits; that is, type 2006 instead of 06.</td>
<td></td>
</tr>
</tbody>
</table>

### Table 10  Microsoft Office Export Limitation

<table>
<thead>
<tr>
<th>Limitation Options</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Export to Excel2 format</strong></td>
<td>Sets the Excel format to use by default for the Export Section to Excel feature. Check this field to select the Excel2 format, which uses the Binary File Format (Biff2). The Excel2 format requires that you select an encoding from the Encoding field on the Export Section dialog box. To select the Excel97 format (Biff8), which supports Unicode (UTF-16LE), do not check this field. When the Excel97 format is used, the Encoding field on the Export Section dialog box is locked since the encoding is already Unicode (UTF-16LE).</td>
</tr>
<tr>
<td><strong>Rows</strong></td>
<td>Specify the default maximum number of rows which can be exported to the source rows in Excel. Do not enter a negative integer in this field. Note that only 65, 536 rows are processed for Excel to process any formulas. If this number is exceeded, Excel truncates the exported section data.</td>
</tr>
</tbody>
</table>
Limitation Options | Descriptions
--- | ---
Columns | Specify the default maximum number of columns which can be exported to the source rows in Excel. Do not enter a negative integer in this field.

**File Locations**

Use the File Locations tab to specify the default locations for Interactive Reporting documents and other necessary files.

To specify default file locations:

1. **Select Tools, then Options, and then Programs Options.**
   The Interactive Reporting Designer Options dialog box is displayed.

2. **Select the File Locations tab.**

3. **Enter the desired options and click OK.**

| Table 11  File Locations Options and their Descriptions |
| --- | --- |
| **Option** | **Description** |
| Documents Directory | The default directory in which to save Interactive Reporting documents when the Save File dialog box is displayed. Interactive Reporting documents are saved in the default directory with a .bqy extension. |
| Connections Directory | A directory that contains the Interactive Reporting database connection files used to connect to databases. The default Connections folder is C:\Hyperion\BIPlus\data\Open Catalog Extensions. |
| Default Connection | The Interactive Reporting connection used when no connection is specified, such as when you click the connection icon in a new document. This field is only available in Interactive Reporting Studio. |
| Preferred Repository Connection | The repository Interactive Reporting database connection file you want the user to see in the Open Repository Connection drop-down list. This field is only available in Interactive Reporting Studio. |
| HTML Template Directory | The directory of HTML templates used with the HTML Export Wizard. This field is only available in Interactive Reporting Studio. |

**OLAP Options**

Use the OLAP Options tab to select OLAPQuery and CubeQuery options.

To select OLAP options:

1. **Select Tools, then Options, and then Program Options.**
   The Interactive Reporting Options dialog box is displayed.

2. **Click the OLAP tab.**

3. **Select the desired options and click OK.**
   See the Results .OLAP Query, and Query options below.
Table 12  Results Options

<table>
<thead>
<tr>
<th>Results</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Download Database Totals</td>
<td>The Download Database totals feature creates only one column per dimension, and includes all members, and the corresponding data, in the results set. In addition, a column with each dimension’s parent members is included. By default this option is disabled. If this option is enabled through the Tools, then Programs, and then the OLAP menu, the Query option is enabled also in a newly created CubeQuery section.</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>If you open an Interactive Reporting version 11.1.1 document that contains data in Results downloaded from a CubeQuery with version 9.3.1, the existing Results data is not visible. Furthermore, using the “download to results” feature with an Interactive Reporting document created with Interactive Reporting version 11.1.1, the result set is not be the same as it was with version 11.1.1.</td>
</tr>
<tr>
<td>Auto Generate Results When Processing OLAP Query</td>
<td>Creates a Results section for any future OLAPQuery section when that OLAPQuery section is first processed.</td>
</tr>
</tbody>
</table>

Table 13  OLAP Query Options

<table>
<thead>
<tr>
<th>OLAP Query</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use OLAPQuery section</td>
<td>Enable the display of a new OLAPQuery (Essbase) section in the pre Release 9.3.1 section format. This option also enables pre Release 9.3.1 section functionality. This setting applies only to subsequent OLAP queries, and not an existing OLAP query. By default, new Essbase queries use the CubeQuery Section type. A pre Release 9.3.1 OLAP query and an OLAP query with a CubeQuery section type can co-exist in the same Interactive Reporting document (BQY).</td>
</tr>
<tr>
<td>Show Dimension Root Level (ODBO Level 0/Essbase Generation 1) Data</td>
<td>Enable the display of level 0 and generation 1 data in the OLAP tree. If you intend to use read or set a full member level hierarchy data in the Dashboards section, this option needs to be enabled. This option does not affect the display of any information in the CubeQuery section.</td>
</tr>
</tbody>
</table>

Table 14  Query Options

<table>
<thead>
<tr>
<th>Query</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto Refresh Query</td>
<td>Queries the database automatically when an item is added to or removed from the data layout, or when the Suppress and Replace options in Query Options are changed. If Auto-Refresh is disabled, you must click Process to query the database whenever you make a change in the data layout. To improve system performance, it is recommended that when you enable the Auto Generate Results option, you disable the Auto-Refresh Query option.</td>
</tr>
</tbody>
</table>

**International Options for Interactive Reporting**

Use the International tab to specify language options for Interactive Reporting documents supporting the Unicode feature.

1. Select **Tool**, then **Options**, then **Program Options**, and then **International**.
The International tab is displayed.

2 Select the desired options and click OK.

<table>
<thead>
<tr>
<th>Table 15</th>
<th>International Tab Options</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Option</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>This is the language of menus and messages.</td>
<td>Select the language to use for Interactive Reporting menus and messages. In the Interactive Reporting Web Client, this setting is available only offline.</td>
</tr>
<tr>
<td>Default BQY sort order language</td>
<td>Select the sort order language for the Interactive Reporting document. Languages differ in how data is sorted. See your administrator before selecting a language.</td>
</tr>
<tr>
<td>Default OCE conversion language</td>
<td>Select the default language of the Interactive Reporting connection file when it is converted to Unicode format.</td>
</tr>
</tbody>
</table>

**Document Properties**

The Document Properties tabs are used to describe information about an Interactive Reporting document including in which application the document was created, the server address associated with a document, and the. Document properties consist of the General tab and the Server tab.

**General Document Properties**

Use the General Document Properties dialog to display the creation and modification attributes of a document.

The following Created display fields for the document are shown on the General Document Properties dialog:

- **Document Path**—Displays the path name to which the document has been saved.
- **Application Name**—Displays the application name in which the document was created.
  - Application names include:
    - Interactive Reporting Studio
    - Interactive Reporting Web Client
    - EPM Workspace
    - Interactive Reporting Job
    - Oracle Hyperion Smart View for Office
- **Application Version**—Displays the release version in which the document was created.
- **Date**—Displays the date on which the document was created.

The Modified display fields for the document are shown on the General Document Properties dialog:

- **Document Path**—Displays the path name to which the modified document was saved.
- **Application Name**—Displays the application name in which the document was last modified. Application names include:
  - Interactive Reporting Studio
  - Interactive Reporting Web Client
  - EPM Workspace
  - Interactive Reporting Job
  - Oracle Hyperion Smart View for Office
- **Application Version**—Displays the release version in which the document was last modified.
- **Date**—Displays the date on which the document was last modified.

**Server Document Properties**

Use the Server Document Properties dialog to specify the server address associated with an Interactive Reporting document.

- To specify a server address for a document, enter the URL address in the Server Address field and click **OK**.

  The URL address must be in the format: http://<server>/workspace/dataaccess.

**International Properties**

When an Interactive Reporting document is opened in Unicode-enabled Interactive Reporting, the Interactive Reporting document file is converted from the old format to the new, and the language is determined based on the document code page. In certain locales (for example, most of the European languages) the language cannot be determined based on the code page from the Interactive Reporting document file. In this case, the user may be prompted to select the correct language for the document. When a new Interactive Reporting document file is created, Interactive Reporting selects the language based on the user locale.language set in Windows unless the user selects another locale/language.

- To select the Document Sort Order Language manually:
  1. **Select File**, then **Properties**, and then **International**.
  2. Select the sort order language from the drop-down.
  3. **Click OK**.

  The document sort order language property is stored in the configuration file with the settings:
  ```
  [Regional Settings]
  DefDocLanguage=English
  ```

  These settings can be copied to the Interactive Reporting server configuration file.
**Customizing Menus**

You can use JavaScript to customize Interactive Reporting menus. Add scripted menu items to the menu bar to:

- Run commonly used scripts
- Launch separate applications
- Export sections to a different file format with a single click

**Note:** Since version 6.0, JavaScript is used as the script-editing tool instead of the Interactive Reporting scripting language. Script written prior to version 6.0 is still recognized, but is enclosed in a wrapper and called with a JavaScript command.

To add a custom menu:

1. **Select Tools, then Customize.**

   The Customize dialog box is displayed.

2. Type the name of your custom menu in the **Menu Name** field.

3. Click **Add** to add a new menu item to the **Menu Items** list.

   The Properties dialog box is displayed.

4. **Select the Text tab.**

5. **Type a name for the menu item.**
Select one or more check boxes to indicate the sections where the menu item should be displayed.

6 **Select the Script tab to display the Script page.**

Enter script commands to be run when the item is chosen from the menu. If you plan to deploy the menu item to a group of users, make sure that commands which reference external applications or files use universal paths.

**Note:** When entering script commands, make sure the JavaScript is kept to one line. Multi-line scripts are not intended for this feature.

7 **Click OK to return to the Customize dialog box.**

Add separator lines and move menu items as needed to complete the final menu.

8 **Click OK to close the Customize dialog box.**

9 **To provide the same functionality for distributed users, copy the preference file which supports this feature to users’ machines.**

   - For Windows, the bqtools.ini file is located in the Windows directory.
   - For UNIX, the .bqtools.ini file is located in the user home directory

**Adding Custom Formats**

You can specify a custom format in an Interactive Reporting document file by adding a new format locale using delimited characters in the bqformat.ini file, which resides in the \WinNT folder and is read by Interactive Reporting to fill in the format dialog boxes. The bqformat.ini file contains all the sections corresponding to the different countries/regions that you will find in the format dialog window. The file is installed with 38 locales set up with pre-defined formats.
To add a custom format:

1. Navigate to the WinNT directory and open the bqformat.ini file using Notepad or Wordpad.
2. Change the second line of the bqformat.ini file to read `count=39`.
3. Scroll down to the line just below `38=USA` and add a line of code that reads `39=BrioSample`.

   A new locale named BrioSample will be added to the date formatting selection. This locale has the same formats as the USA date formatting selection, but additionally allows users to select a date that reads, "Today is mm/dd/yyyy"; where mm/dd/yyyy is the format for the current date.

4. Highlight all the code for USA, and select Edit, then Copy.
5. Place the cursor at the end of the last line of code and select Edit, then Paste.
6. In the newly copied code, replace any lines that read `USA_`, with `BrioSample_`.

   Be sure to leave the words date, time, number, percentage and currency after `BrioSample_`.

7. Save the bqformat.ini file and close the editor.
8. Open an Interactive Reporting document file and open up the formatting window (on the shortcut menu, select Number).
9. Use the pull down menu for 'Select a Formatting Locale' and scroll down until you see the BrioSample section that was created in the bqformat.ini file.

To add a custom format with a delimited character:

1. Navigate to the WinNT directory and open the bqformat.ini file using Notepad or Wordpad.

   Note that eight BrioSample_date formats are in this section. This procedure show you how to add one custom format with a delimiter character to the BrioSample_date section added above.

2. Change the line that reads `count=8` to `count=9`.
3. Add the code after `8=dd/mm/yyyy HH:MM`.

   `9=To\day is mm/dd/yyyy`

   This format allows users to print the words "Today is" prior to printing the date.

   The code To\day contains a delimiter character before the 'd', the backslash. This backslash must be used in order to print the word 'Today' or the letter 'd' would be replaced with the day number— the slash mark indicates the character is not to be replaced with the date or time element. The characters that should be delimited are; d, m, y, H, M, F.

4. Save the bqformat.ini file and close the editor.
5. Open an Interactive Reporting document file and test the new formatting section you just added on a date field.
6. Right-click a date column, select NUMBER, and select the BrioSample from the Formatting Locale pull-down menu
7. Select Date as the Category.
You should see "Today is mm/dd/yyyy" in the list of selections.

Notice in the Preview screen that the delimiter character prints the word "Today" correctly without errors.

As an additional exercise, try modifying the new date format without using the delimiter character and see how the format is displayed.

### Tools Menu Command Reference

Table 16 provides a quick reference to the commands available on the Tools menu and list any related shortcuts. The commands listed are those available in the query and results sections.

#### Table 16  Tools Menu Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Keyboard Shortcut</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process Query</td>
<td>Processes the current query, all queries in the Interactive Reporting document file, or a customized selection of queries.</td>
<td></td>
</tr>
<tr>
<td>Connection</td>
<td>Allows you to select, log on, log out, modify, or create an Interactive Reporting database connection file.</td>
<td></td>
</tr>
<tr>
<td>Connections Manager</td>
<td>Opens the Connections Manager dialog box.</td>
<td>F11</td>
</tr>
<tr>
<td>Save Connection</td>
<td>Saves an Interactive Reporting database connection file with an Interactive Reporting document file.</td>
<td></td>
</tr>
<tr>
<td>Administer Repository</td>
<td>Opens the Administer Repository dialog box, where you can modify object descriptions or groups.</td>
<td></td>
</tr>
<tr>
<td>Launch Dashboard Studio</td>
<td>Launches Dashboard Studio, which is a wizard-driven dashboard development tool for Interactive Reporting and lets you rapidly build dashboards without writing any code.</td>
<td></td>
</tr>
<tr>
<td>Launch Dashboard Architect</td>
<td>Launches Dashboard Architect, which is an integrated development environment (IDE) for Interactive Reporting and lets you rapidly build, test, debug and integrate Interactive Reporting applications, dramatically increasing developer efficiency and productivity.</td>
<td></td>
</tr>
<tr>
<td>Options</td>
<td>Allows you to set default formats or program options.</td>
<td></td>
</tr>
<tr>
<td>Customize</td>
<td>Opens the Customize dialog box where you can add customized menus or menu items.</td>
<td></td>
</tr>
<tr>
<td>Change Database Password</td>
<td>Change the password used to log into the database. This feature is database dependent as only the following providers enable users to change the password: Teradata Red Brick Warehouse Oracle Essbase Sybase System Informatica For all other databases, this feature is greyed out.</td>
<td></td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td><strong>Change Web Client Version</strong></td>
<td>Interactive Reporting Web Client only for locally saved (offline) documents. Enables users to change the Web Client version in which to open an Interactive Reporting document file. Change the Interactive Reporting Web Client version in which the document file is opened. Available for offline Interactive Reporting document files only.</td>
<td></td>
</tr>
</tbody>
</table>

**Export Properties**

Use the Export Properties dialog box to specify export properties for data in the Results, Table, Pivot, and OLAPQuery sections.

Table 17 Export Properties

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Break HTML Page Every ___ Row</strong></td>
<td>Defines how many rows to include on an HTML page before starting a new page (file). The default setting is 100 rows per HTML page. Either enter the number of rows or click the up and down arrows to increase or decrease the row number by ten. If &quot;No Page Break&quot; is selected, this option is disabled.</td>
</tr>
<tr>
<td><strong>No Page Break</strong></td>
<td>Exports the data to one HTML file.</td>
</tr>
<tr>
<td><strong>Export Without Quotes</strong></td>
<td>Exports a section to a tab-delimited text file with no double quotes (or any other character that is not the raw Results data) around any column data. If you do not select this option, the exported section data contains double quotes surrounding exported column/cell values containing real data.</td>
</tr>
</tbody>
</table>

**Export Document as Web Page**

Use the Export Document As Web Page dialog box to publish the contents of the Interactive Reporting document file as a Web page. You can select which sections of the current Interactive Reporting document file to include in the export set.

Interactive Reporting documents file sections are exported in the same order as they are displayed in the Interactive Reporting document. Exportable sections include: Results, Table, Pivot, Chart, and reports created with the Report Designer. Imported sections that have been added to the Interactive Reporting document files, such as text files and Excel files, can also be exported. Data models and the Query, OLAPQuery, and Dashboard sections are not exportable. Neither are empty or blank sections.

The files created by this export option include:

- Main (or frameset) HTML (.htm) page
- Document-type icons (.gif)
- Cascading style sheets (.css) that describe the layout primitives
- Individual HTML (.htm) pages for each selected section
JPEG files (\*.jpg) with the report content

After these files are posted to a Web site, you can access the files individually or use the main HTML file to view the frameset, which lists the sections in the report. The exported Web pages include navigation buttons so you can scroll back and forth between pages, or jump to the beginning or end of an Interactive Reporting document file. In addition, the current page number and total number of pages are included in the report.

To export an Interactive Reporting document file as a Web page:

1. Select the sections of the Interactive Reporting document file to export and click OK.
   
   To select all sections, click Select All. To clear all selected sections, click Deselect All.
   
   The Save As Type dialog box is displayed.

2. Navigate to the location in which to save your Web page.
   
   **Tip:** Create a separate folder to store the files created by this export option.

3. Type a name for the exported Web page in the File Name field, or accept the default name, for example Sample1.htm.
   
   The Save As Type field is set to *.htm by default. All of the HTML files that the export process creates begin with the name specified in the File Name field, for example, Sample1Chart.htm, Sample1Pivot.htm, and so on. The HTML file with the exact same name as the name specified in the File Name field is the main HTML page, and it contains the frameset and links to all of the other pages, for example Sample1.htm.

4. Click Save.
   
   To view the exported selection, open your Web browser, select File, then Open, browse to the location of the exported files, and open the main HTML page.

---

**Export to HTML Wizard**

The Export To HTML Wizard helps you create Web pages from charts, reports, and pivot tables. When you use the Export to HTML Wizard, each page of the Chart, Detail, or Designer report is a JPEG file, which is referenced by an HTML file.

**Note:** We suggest you use the Export Document As Web Page command (see Exporting a Document as a Web Page) rather than the HTML Wizard.

To export reports using the HTML Wizard:

1. On the first page of the wizard, select an option:
   
   - Click Create a new export layout if you have never used the Export to HTML Wizard before, or if you want to create a new export layout file instead of using or modifying an existing one.
Click **Use an existing export layout** to use an existing Web layout file from your Style Sheets directory.

To search for an existing export layout file, click **Browse**.

2 On the second page of the wizard, specify the style of the home page to create for the exported HTML data, and whether to link it to other existing Web pages. To do this, complete:

Enter a name for your home page in the **Home Page Name** field with an HTM extension.

Select an output style.

Interactive Reporting supports the output styles:

- Vertical panes
- Horizontal panes
- Table of Contents
- Linked Pages

3 On the third page of the wizard, specify which reports to turn into Web pages.

All the reports available for potential export to Web page format are listed under Available Reports. Each report that you select will have its own Web page, and links to these pages will appear in the order shown under Selected Reports. You must specify at least one report for export.

4 To export a report into Web page format, click a report name under Available Reports, and click the right arrow to move it to Selected Reports.

5 To rearrange the order in which the Web pages will be linked, select a report name under Selected Reports, click the up arrow to move the report up one line, or click the down arrow to move the report down one line.

6 On the fourth page of the wizard, you can optionally specify another HTML template to be used when creating your Web pages. To select another HTML template, select a report name under Selected Reports, and click the right arrow to move it to HTMLTemplates.

**Tip:** An HTML Template controls the size and style of the fonts on a Web page.

7 On the fifth page of the wizard, you can optionally link your Web pages to your source documents. This feature allows you to view the most recent updates on the Web and perform online analysis. To enable online analysis, select **Please link my Web pages to their ‘source’ documents**.

**Tip:** To use online analysis, you must have Interactive Reporting Web Client installed on your machine.

8 On the sixth page of the wizard, type the full path to the directory where you want to store your new HTML pages, or click **Browse** to navigate to that location.

9 On the seventh page of the wizard, save your selections as a new HTML Export Layout file. To save your selections and finish building the layout, specify the name and path of the layout you wish to create and click **Finish** to exit the wizard.
The Export To HTML Wizard saves your file with the extension "htm" in the specified directory.

**Conditional Formatting**

Use Conditional Formatting to associate specific item values with format styles. Conditional formatting is an effective way to call out important elements in a report. If an item value matches a conditional expression, the format is applied. For example, you could use this feature to have item values appear blue when a value is greater than 10,000, or red when it is less than 5,000. When a value does not match a condition, the default format of the value is applied.

In the example below, the format bold and red have been applied to amounts greater than 10,000, and the format bold and green have been applied to amounts less than 1,000:

Based on the section, Conditional Formatting behaves as follows:
To apply a conditional format:

1. **Select an item to which to apply a conditional format.**
   
   Conditional formatting is applied to the entire column, label or row which meets the condition, not just the selected cell.

2. **Select Format, then Conditional Formatting.**
   
   The Conditional Formatting dialog box is displayed.

3. **If desired, you can select another item to which apply the conditional formatting from the Item drop-down.**
   
   Otherwise, the item selected in step 1 is shown.

   **Note:** In the Pivot section, ”item” refers to labels and rows. In the Table sections, item refers to a column.

4. **In the Operator drop-down box, select the comparative operator to use for the conditional expression.**
   
   See below for a description of valid operators.
   
   If you select the *Between* operator, the Conditional Formatting dialog includes two Value fields. Select the beginning and ending values in these fields.

5. **In the Value field, enter the comparative value to use in the conditional expression.**

6. **Select the format to apply to item values. Valid options are:**
   
   - bold
   - italic
   - underline
   - fill color
   - text color

7. **Click Apply to apply the formats to the conditional expression.**
   
   You can view conditional formats by selecting *Sample* to the right of the conditional expression. To view other conditional expressions formats, select on the Sample bar associated with the expression.

8. **If desired, add more conditional expressions.**
When the number of expressions totals three, the Add Expression hyperlink appears. Selecting the hyperlink adds another line conditional format controls. When the number of lines becomes more than three, a scroll bar is displayed.

9 Click OK.

➢ To copy a conditional format:
1 Select the column, label or row that has the conditional formatting you want to copy.
2 Select Edit, then Copy Conditional Format.

➢ To paste a conditional format.
1 Select the column, label or row to which to paste a conditional formatting.
2 Select Edit, then Paste Conditional Format.

➢ To delete a conditional format, select the (blank) operator from the Operator field next to the format to be deleted, and click Apply.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(blank)</td>
<td>Empty operator (used to mark an expression for deletion)</td>
</tr>
<tr>
<td>= Equal</td>
<td>Item value is exactly as in the expression.</td>
</tr>
<tr>
<td>&lt;&gt; Not Equal</td>
<td>Item value is not equal to any value in the expression.</td>
</tr>
<tr>
<td>&lt; Less Than</td>
<td>Item value is less than that one in the expression.</td>
</tr>
<tr>
<td>&gt; Greater Than</td>
<td>Item value is greater than that one in the expression.</td>
</tr>
<tr>
<td>&gt;= Greater or Equal expression</td>
<td>Item value is not less than that one in the expression.</td>
</tr>
<tr>
<td>&lt;= Less or Equal</td>
<td>Item value is not greater than that one in the expression.</td>
</tr>
<tr>
<td>Begins With</td>
<td>Value in the expression is a symbolic prefix of item value.</td>
</tr>
<tr>
<td>Contains</td>
<td>Value in the expression is a symbolic infix of item value.</td>
</tr>
<tr>
<td>Ends With</td>
<td>Value in the expression is a symbolic suffix of item value.</td>
</tr>
<tr>
<td>Like</td>
<td>Item value matches wildcard string in expression ( “%” is used instead of asterisk in wildcards).</td>
</tr>
<tr>
<td>Between</td>
<td>The first expression value is less or equal to the first item value, and the second expression value is equal or greater than the second item value. To accommodate the Between parameter, the Conditional Formatting dialog includes two Value fields. Use these fields to specify the beginning and ending values.</td>
</tr>
<tr>
<td>IsNull</td>
<td>Item value is null.</td>
</tr>
</tbody>
</table>
Query Section

The Query section is the Interactive Reporting interface with databases. In the Query section, you can examine the overall informational contents of the database, and even look underneath to verify the actual data values.

Each Query section contains a Data Model. Some Data Models are locked, and only allow you to process the query; others allow you to make modifications to the query or build a new one. You can also apply filters (limits), compute and modify items, and process your request for information.

An Interactive Reporting document file can contain multiple Query sections. These query sections can access a wide range of data sources (relational databases, OLAP servers, imported data sets, and local joins). Each Query section has its own Results section and can be associated with the same database or different databases (that is, the Interactive Reporting database connection file or data model used is independently defined in each query).
Using Data Models in the Query Section

Relational queries use data models to view the server database tables and create queries. Depending on your access privileges, you can download a prebuilt data model to the Query section, or build a data model and create your own query.

When you connect to a database, the tables in the database are displayed in the Catalog pane of the Query section. A data model is a visual representation of these tables. You use a data model to create queries that specify which data to fetch from the database and retrieve for analysis.

To view the database tables in the Table catalog, select DataModel, then Table Catalog.

The Table catalog in the Catalog pane expands to show all of the tables in the database. If you are not connected to a database, Interactive Reporting prompts you for a user name and password.

To create a data model, in the Catalog pane, select a table from the Table catalog, and select Query, then Add Request Item(s).

Building Queries

To build a relational query:

1. Click the Request button on the Section title bar to display the Request line.
2. Complete one of the following actions:
   - Drag an item from the Content pane to the Request line.
   - Select an item in the Content pane and select Query, then Add Request Item(s).

To add an entire column from a table to the Request line, select the table header. You also can select more than one of the same item (to create duplicate items).

If you add more items than the Request line can display, use the arrow buttons at the right of the Request line to scroll through the requested items, or resize the Request line to display multiple rows of request columns.

Working with Items on the Request Line

As you build your query, you can reorder, remove, or hide items on the Request line. This allows you to change the way in which the query processes and displays.

Reordering Request Items

You can move Request items to reorder them for viewing results.
To reorder items on the Request line, select the item to be moved and drag it to a new location on the Request line.

Removing Request Items

You can remove items from the Request line to exclude the data from your query or Results set.

To remove an item from the Request line, select the desired item and complete one of the following actions:

- Click the Delete button on the standard toolbar.
- Click Remove on the shortcut menu.
- Press the Delete key.

If you have not yet processed the query, Interactive Reporting removes the item from the Request line.

If you have previously processed the query, the Report Refresh dialog is displayed with this message: “The section XXXX (section) references the following removed column(s): XXXX (column name). Do you want to keep references to these columns or turn off auto-refresh or remove them with the query is next processed?” You can select to keep the references, or remove them.

Caution! Remove items with caution as a computed item or report may draw data from the item you delete.

Hiding Request Items

You can hide items that are displayed on the Request line. This allows you to incorporate data in the results set without displaying it. Hidden request items cannot be referenced for computations.

To hide a request item, complete one of the following actions:

- Select the item and click Hide on the shortcut menu.
- Select the item and choose View, then Hide Request Items.

To show a hidden request item:

1. Complete one of the following actions:
   - Click in the Request line and click Unhide on the shortcut menu.
   - Select View, then Unhide Request Items.

   The Unhide Columns dialog box is displayed.

2. Select the items to view and click OK.
When you have identified the items to include in the query, you can perform a number of other operations before processing the query. You can add filters or computed items to the Request line, or you can use a Request line item to specify a sort order.

Related Topics
“Using Filters” on page 309
“Computed Items” on page 323
“Applying Sorts” on page 401

Processing Queries
After you build your query and apply filters, computations, sorts, and any other adjustments to further refine your request, you need to process it. Processing your query may take a few moments if your query is complex or if the data in linked report sections needs to be refreshed.

When you process your query, the data is retrieved to the Results section in tabular form. You can reprocess your query at any time and in any section to refresh the data. You can also return to the Query section from any other section at any time to alter the query and reprocess it.

To process a query:
- Click Process on the standard toolbar. (Click the right-arrow to select a process option.)
- Select Tools, then Process Query and select the desired process option.

Since an Interactive Reporting document file can contain multiple queries, there are three processing options on the Process drop-down list:
- Process Current—Processes the current object. In some cases more than one query may be processed, for example, if a report references results sets from multiple queries. Process Current is the default selection when using the toolbar button.
- Process All—Processes all the queries in the document. By default, queries are processed in the order in which they are displayed in the Section catalog. For example, in a document with three queries, Query1, Query2, and Query3, the queries are executed in that order when you select Process All. (See “Query Processing Order” on page 95).
- Process Custom—Opens the Process Custom dialog box so that you can select the order in which queries are processed.

The query is sent to the database and retrieved data is displayed in the Results section. While the data is retrieved, the Status bar displays a dynamic row count indicating rate and progress of server data processing and network transfer.

Note: In the Interactive Reporting Web Client, a ZException might occur if you attempt to process a large query. To process larger queries successfully, increase the value of the DAS property "mdd Partial Result Cell Count", and then restart DAS.
Query Processing Order

If there are multiple queries in a document, you can determine the order in which they get processed, and select which queries to include or exclude from the processing stream.

This feature is particularly useful when you need to use “fresh” data during a local join operation, or to ensure that your local results are populated with current data from a source query not affected by several earlier tables without duplicating the processing of some queries.

Query Processing Order is available for the Query section, OLAPQuery section and any imported data files sections that can be processed in an Interactive Reporting document when you use the “Process All” feature.

Note: Query processing order settings are saved with the document. For a temporary processing order, do not save the document or set the processing order to what is normal for the given document.

To specify a query processing order:

1. Select Tools, then Process Query, and then Processing Order.

The Query Processing Order dialog box is displayed.

The first time the Query Processing dialog box is displayed after the new document is created, all query sections belonging to the document are displayed in the order which they are displayed in the Section Catalog (top down). Query sections added after the sections have been arranged, are displayed at the end of the list in the order in which they were added when the dialog box is reopened.

2. Select a query section and move it up or down in the processing order using the arrow keys to the right.

3. Double click a query section to remove it from the processing order or add it back and click OK.

Only queries marked with an asterisk (*) are processed during “Process All”.

Processing Queries 95
Select Tools, then Process Query, and then All.

**Saving Queries**

After you process a query, your data is available until you close the Interactive Reporting document file Saving the Interactive Reporting document file saves the current formatting and layout of all sections.

➤ To save your query, select File, then Save.

**Cancelling Queries**

To cancel a query, both Interactive Reporting and the database must communicate properly for reasons other than running a query.

➤ To cancel a query, press and hold Alt+End until the query is cancelled.

The Interactive Reporting database connection file used to connect to the database must use an Asynchronous API. This is the default configuration, but it can be turned off on some databases. If the Asynchronous API is disabled, the database cannot detect a new request until the query has finished processing.

The database must pause every so often even when processing a query. As a default configuration, many databases are not set up to pause, which could result in the database being too busy to hear the cancel request until the query has finished processing.

**Building Subqueries**

You can use subqueries to filter your data. A subquery answers a specific question or provides specific information within the context of a main query, also called a “parent” statement. The database evaluates the entire query by first analyzing the subquery. The parent statement filters its rows based on the rows retrieved by the subquery.

Review these topics for information on subqueries:

“Regular Subqueries” on page 97
“Correlated Subqueries” on page 98
Regular Subqueries

A regular subquery executes the inner and outer queries once and returns the values from the inner query to the outer query. For example, you might need to find out who sold more than the average of all sales representatives in April. You first use a subquery to define what was the average sales amount in April. This information is supplied to the parent query, which determines which representatives exceeded the average of all sales in April.

To build a regular subquery:

1. Select or build a parent query.
2. Select an item on which to set a filter by:
   - Double-clicking the topic item.
   - Dragging the item from the Content pane to the filter line.
   - Right-clicking an item and selecting filter on the shortcut menu.

   The Filter dialog box is displayed.

3. **Click Advanced.**
   The Advanced button toggles the Create Subquery button.

4. **Click Create Subquery.**
   A subquery indicator shows that you are working with a subquery and not the parent query, even though the data models of the parent query are displayed in the Content pane (which now has a gray background).

   The Subquery section is considered a child of the parent query and is subordinate to the parent query. It is dependent on the parent query and does not have its own default Query or Results sections. Subqueries can be nested within other subqueries, in which case the first subquery becomes the parent query to the subquery nested inside it.

5. **Build the query.**
   Only one item can be on the Request line, but you can add server and local filters, set a filter inside a subquery as a variable, and define data functions and computations as needed.

6. **In the Section pane, click the parent query for the subquery.**
   The parent query section is redisplayed.

7. **Click Process to process the parent query and subquery.**

Modifying Subqueries

You can modify the name of the subquery filter, select the data values associated with a data item, use mathematical logic to apply the values as constraints, or display and modify the data model associated with the subquery item.

Subquery properties include:
- **Name**—Default name for the subquery filter. If you have two or more filters for a topic item (or items with similar names from different topics), you might want to rename the filter. Renaming the subquery filter does not change the SQL associated with the filter.

- **Include Nulls**—Toggles the inclusion/exclusion of null values.

- **Not**—Reverses the effect of an operator (for example, ‘Not >=’ is equivalent to <).

- **Operators**—Comparison operators for the filter expression. Values that pass the comparison test will be included.

- **Modify Subquery**—Summary subquery filter information. To modify the data model associated with the subquery, click the Modify SubQuery. If the data model section is hidden, select View, then *Unhide* section to view the section.

### Correlated Subqueries

A correlated subquery is related to a regular subquery in that it uses an inner query to feed result values to the outer query. A correlated subquery executes the outer query multiple times, once for each row returned by the inner query; it is processed by joining a column in the subquery to a column in the parent query.

For example, suppose you had to identify which sales representatives had more sales in the current month than they did in the previous month. The correlated subquery is executed for each row of sales information in the parent query to first determine what were the sales for each representative in the previous month. This data, in turn, is compared to sales for each representative in the current month, and only those representatives whose sales in the current month were greater that their previous month’s sales are returned.

To build a correlated subquery:

1. **Select or build the parent query.**
2. **Use one of the following options to select an item on which to set a filter:**
   - Double-click the topic item
   - Drag the item from the Content pane to the Filter line
   - Right-click and item and select **Filter** on the shortcut menu.
     - The Filter dialog box is displayed.
3. **Click Advanced.**
   - The Advanced button toggles the Create Subquery button.
4. **Click Create Subquery.**
   - A subquery indicator shows that you are working with a subquery and not the parent query, even though the parent query’s data models are displayed in the Content pane (which now has a gray background).

The Subquery section is considered a “child” of the parent query and is subordinate to the parent query. That is, it is dependent on the parent query and does not include its own default query or results section.
Subqueries can be nested within other subqueries, in which case, the first subquery becomes the parent query to the subquery nested inside it.

A subquery is correlated based on a join from a column in the subquery to a column in the parent query.

5 Build the query.

Only one item can be on the Request line, but you can add server and local filters, set a filter inside a subquery as a variable, and define data functions and computations as needed.

6 If the parent query is not displayed in the Catalog pane, select Show Queries from the shortcut menu in the Section pane.

7 Drag the parent query into the Content pane.

The parent query is displayed as a blank topic in the Content pane.

8 Drag the topic to correlate by into the blank parent query topic.

The Select Correlation Column dialog box is displayed and shows the tables of the parent query.

9 Use the + and – signs to navigate through the structure of the directory tree.

10 Select the column in the parent to which to join the subquery topic item and click OK.

The topic item is added to the Filter line and a join line is drawn.

11 In the Section pane, click the parent query section.

The parent query section is redisplayed.

12 Click Process to process the entire query.

The topic item added in the subquery shows the label sub next to the topic item name.

---

**Derived Tables**

A “derived table” is essentially a statement-local temporary table created by means of a subquery in the FROM clause of a SQL SELECT statement. It exists only in memory and behaves like a standard view or table.

For example, assume an Oracle 9i database table is called “state_table” and has the following row values in the “state” column.

<table>
<thead>
<tr>
<th>Table 18</th>
<th>Derived Tables Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>State</strong></td>
<td></td>
</tr>
<tr>
<td>CA</td>
<td></td>
</tr>
<tr>
<td>CA</td>
<td></td>
</tr>
<tr>
<td>CA</td>
<td></td>
</tr>
<tr>
<td>FL</td>
<td></td>
</tr>
</tbody>
</table>
If you used the following inner SELECT statement, which includes a derived table to evaluate the “state_table”, you could return the count and percentage of each state. The SQL has been written for Oracle 9i.

```
select         state, count(state) as State_Count,
               (count(state)/derived_table.tot_state_count) as State_Percentage
from State_table,
    (select count(state) tot_state_count from state) derived_table
group by state, derived_table.tot_state_count;
```

The results of the query is displayed below:

<table>
<thead>
<tr>
<th>State</th>
<th>State_Count</th>
<th>State_Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA</td>
<td>3</td>
<td>0.75</td>
</tr>
<tr>
<td>FL</td>
<td>1</td>
<td>0.25</td>
</tr>
</tbody>
</table>

Derived tables are useful when you need to generate aggregates in a table that would otherwise contain dimension type data and join the resulting aggregate with detail level facts in another table.

Additionally, the aggregate values in the derived tables can be used in the outer query’s WHERE clause (i.e., "where salary > average_salary", average_salary has been derived by the subquery). These tables can also optimize a query’s performance in some circumstances, such as minimizing sorting when some DISTINCT values are needed from some tables but not all. Finally, it might eliminate the need in some cases, to build "local results" queries. This feature enables users to access this type of SQL construct in an easy-to-build manner.

**Derived Tables Rules and Behavior**

The rules and behavior of a derived table include:

- A Derived Table cannot be “iconized”.
- A Derived Table only has two speed menu options: “Add Selected Items” and “Remove”.
- When you modify a Query section that is used as a Derived Table:
  - Derived tables are available for connections to Teradata, DB2, Oracle, and Microsoft SQL Server. If you use the feature when you build a query, an active connection is required if the server code is ODBC or OLEDB so that the “smart ODBC” can determine the database type.
  - All client versions of Interactive Reporting use derived tables. Scripts in Interactive Reporting document file opened with EPM Workspace can also use this feature.
  - If you remove all items from the Request line, the topic are displayed empty (there are no columns) in the section(s) where they are used as a derived query.
  - If the columns referenced by the “deriving” Query section are removed from the Request line, the effect is the same as if a “sync with database” had found the column removed.
If you rename items in a Query section used as a derived table, you must ensure that the new name is a valid name for the database in use. These names are used during the SQL generation, and are quoted if "quoted identifiers" are selected in the database connection.

- Items from "Local Results" collection cannot be added to the Query.
- The icon used for a derived table in the Table Catalog tree is shown on the left.
- Both the referencing query and the derived table query must use the same data source even if one or the other is not connected. For example, assume Query1 is built from DataModel1, and Query2 is a query section not derived from a DataModel section. That is, you must be sure that both Query1 and Query2 can connect to the same data source name (such as an ODBC data source name or an Oracle TNS name), and they share the same API and server codes. The data source name alone may not be sufficient since two names can be identical over two different APIs. However, it is not required that the connections use the same database credentials. You do have to ensure that the referenced tables are accessible from whatever logon is used in all selected Query sections.

- Interactive Reporting Web Client users need at least a Query role to use derived tables in a data model, or data model role to add or remove them from the data model.

To build a query that uses a derived table:

1. Build the query which will use the "derived table" and process the query.
2. Insert a new query by choosing New Query on the Insert menu.
   A derivable query can be built for a relational database.
3. Build the "Derived Table" query by clicking anywhere in the Catalog pane and selecting Derivable Queries on the shortcut menu.

   The “Derivable Queries” option is not displayed if no query sections can be used in the current query.

   (The Table Catalog below shows the Derivable Queries tree expanded and the Tables tree contracted. Local Results only are displayed when a user has requested them from the speed menu.)

   The only queries that are displayed in this list are those that:
   - have the same connection information as the current Query section
   - have at least one item on the Request line
   - do not use local joins
   - do not also contain derived table topics

   The topic name, when a derived table is added to the work area, is the same as the Query section name, and the column names are the same as the names of the items on the Request line in the Query section being added with the exception of any (data) function component. In cases where the function is displayed on the Request line, the resulting topic item name would be the same as the name that is displayed in Results after processing the query. The
following diagram illustrates how the Request line for Query2 are displayed as a topic in another Query section.

4 Create a manual join by dragging an item from at least one topic to another, including to/from the added derivable query sections.

Once the derived table becomes a topic, items from it can be added to the Request, Filter, or Sort lines of the containing query. It can be referenced in computed item dialogs, and can be used in custom GROUP BY logic.

If you use the Show Values feature when setting a filter on an item in the “derived table”, the SQL that would be used is the same as if you set a filter on the same column in the source Query section.

5 Process the query using the derived table.

You can process the query by way of “Process” command when viewing the query section (or one of its dependent sections) or by checking it in the Process Custom dialog.

The “Process All” command processes the query containing the derived table. It also processes the query section from which the derived table is derived, unless it explicitly removed using the Query Processing Order dialog (which is recommended to avoid duplication of processing at the database.)

Additions to the Request line of the source Query are reflected in the topic item list of the referencing Query section the next time it is displayed.

Once a Query section has been added to another Query or Data Model works area, changes to the source query section will check for dependencies and warn you of any discrepancies that might occur. For example, you might receive a warning if an item was removed from the source query’s Request line which is used someplace else (Request, Filter, Sort, etc.) in a Query that is deriving a table from the source query.

Derived Tables and SQL

Review the following sections for information on:

- Custom SQL
- Processing a Query that Contains another Query Section

Custom SQL

If you open the Custom SQL window in a query that has a derived table topic, the SQL for the derived table is shown in the custom SQL window as part of the overall query. At this point, the SQL is locked; changes in the source query section for the derived table are not reflected in this SQL until such time as the user presses the Reset button. When you process the query with the Custom SQL window open, it is executed “as is”.

102 Querying
Processing a Query that Contains another Query Section

When you process a Query section that contains another Query section, it forms SQL that uses what are called “derived tables”, which are essentially subquery statements in a FROM clause. The SQL generation phase behaves as follows:

- Generates the SQL for the Query section being used as a topic in the current Query section being processed, including resolving any variable filters. Any Sort line items in the source Query section are ignored in generating the SQL. This generation only occurs if the contents of the Query as a topic is referenced by the currently processing Query section, following the rules for join path generation specified for the currently processing section.

- The resulting SQL is added to the currently processing Query section if required, in the FROM clause, surrounded by parentheses, as with any subquery.

- The entire subquery is given a table alias. The alias will follow the pattern for any table involved in a query and will thus be of the form “ALn”. As with subquery filters, any aliases used in the subquery itself will be distinct from the alias names used in the currently processing Query section’s table references. The alias will be used thereafter to refer to columns in the “derived table”, as is normal for SQL generation based on physical tables.

- Following the table alias name, “ALn”, a parenthesized list of column aliases is added. This list is identical to the list of names of the topic items in the Query topic. (For Oracle, the column aliases are displayed in the select list of the subquery instead of in a separate parenthesized list, for example, SELECT AL1.STORE AS store….)

- Before submitting the resulting SQL to the database, a check is performed to ensure that the query is properly joined. If it is not, the standard behavior, as specified by the Interactive Reporting database connection file preference “Allow Non-Joined Queries”, is taken.

- Following submission of the SQL, any references to the column alias names on the Request line of the currently processing Query section are available to build the dependent Results section.

- The minimum requirement is that if the “derived table” is referenced by the currently processing query, its SQL is generated as if that section were processed alone, with the exceptions noted above. At initial implementation or in a future release, the SQL would include only those columns actually needed by reference in the currently processing Query. This reduces the number of items in the SELECT list of the subquery, with a corresponding reduction in the number of column alias names.

For example, assume the following Query section is to be used as the source for a “derived table” in another Query section:
In this example, the computed item’s definition is “UCASE (Pcw_Customers.Buyer)”. The filter on ‘Store Type’ is set to ‘Discount’.

The Query section using “Query” as a derived table might appear as follows:

In the previous diagram, the filter on City is set to ‘Los Angeles’. The SQL generated when the above query is processed, per the minimum requirements, would be:

```
SELECT AL2.STATE, SUM(AL1.UNITS) FROM PCW_SALES AL1,
(SELECT AL3.STORE_ID, AL3.CITY, AL3.STATE, UCASE(AL3.BUYER)
FROM PCW_CUSTOMERS AL3
WHERE AL3.STORE_TYPE = 'Discount') AL2(STORE_ID, CITY, STATE, COMPUTED)
WHERE AL2.STORE_ID = AL1.STORE_ID AND AL2.CITY = 'Los Angeles'
GROUP BY AL2.STATE
```

**Working with Query Section Data**

You can further enhance the data requested in the Query section by working with the data, as described in these topics:

- “Processing Results to a Database Table” on page 105
- “Estimating Query Size” on page 106
- “Displaying Database Remarks” on page 106
- “Preaggregating Data Using Functions” on page 107
- “Appending Queries” on page 109
- “Using Local Results” on page 111
- “Using Stored Procedures” on page 112
- “Setting Query Options” on page 113

You can accomplish these tasks by using the commands on the Query menu.
Note: See “Query Menu Command Reference” on page 124 for a complete list of the commands available on the Query menu.

Processing Results to a Database Table

Instead of retrieving data to the Results section, you can instruct Interactive Reporting to create a table in the database to store your results set. Items on the Request line become the column headings of the new table, and you can append new columns to the table and query it as needed.

Note: The Interactive Reporting database connection file and database to which you are connected determine whether you can use this feature. You must have Create and Insert privileges on the database to process results to a database table.

To process results to a database table:

1 Select Query, then Process Results To Table.
   The Process Results to Database Table dialog box is displayed.
2 Specify the information requested.
   - **Table Name**—Name of the new table to be created or the name of an existing table to which you want to append columns. You can create tables under your own owner name or under different databases or owners. If you do not have the correct privileges or do not specify an alternate location, the table is created under your own owner name. Use the format `DATABASE.OWNER.TABLENAME` to specify alternate names.
   - **Create Table**—Creates a new table in which Request items form columns.
   - **Append To Table**—Appends Request items as new columns in an table.
   - **Grant Access To**—Enables either everyone or specific users to access the new tables. Type PUBLIC or specific user IDs (separated by commas) for each user who should have access to the table. If not selected, access to the table is filtered to your own user ID.
3 Click OK.
   The table is created or modified under the specified database and owner name.

To verify that the query was processed and the results saved as a database table:

1 Select DataModel, then Table Catalog.
   The Table Catalog expands in the Catalog pane.
2 Select Refresh on the shortcut menu.
   The table is displayed in the list of database tables.

Interactive Reporting tracks tables under your database user name and stores a list of these tables in the `bqtbls5.ini` file.
To delete tables you created using the Process To Database Table feature:

1 Select Query, then Process Results To Table.

The Process Results to Database Table dialog box is displayed. Tables created under your user ID are displayed in the Tables Created By list.

2 Select a table from the list and click Delete.

Estimating Query Size

Queries that sift through and retrieve enormous amounts of data can take a long time to process, and may consume unnecessary system and server resources. If you suspect these factors exist, you may want to size your query before you process it.

The Estimate Size feature queries the database to see how many records your query will retrieve. You can use this feature to test a questionable query or to decide whether to prevent or postpone processing a large results set.

To estimate the size of a query, select Query, then Estimate Query Size.

Interactive Reporting queries the database and counts the number of records to retrieve if the query is processed. This process may take a while for server-intensive queries.

Displaying Database Remarks

Database remarks provide detailed contextual information about a table or column. Remarks may describe the origin, derivation, or details about data model topics and items, which can help you identify and select the information you need. Database remarks often exist as metadata when you map data in a data warehouse project or if you use a CASE tool to manage your database.

To display database remarks, use one of the following options:

- Select Query, then Show Remarks.
- Click Show Remarks on the shortcut menu.

The following is a sample remark:
Preaggregating Data Using Functions

Depending on how you plan to view your data, you can select to preaggregate data at the database server. Preaggregation (also called server aggregation) is a querying strategy that uses functions to summarize data as it is retrieved from the database. Instead of returning a line-item list of every row that meets the criteria on your Request line, you can order the database to group related information. This results in one row representing the combined (aggregate) value of each distinct group.

You use data functions (provided by your RDBMS) to preaggregate data in a query. When a data function is applied to a Request item, the data related to that item is aggregated when the query is processed. If you need both summary data and increasing levels of detail breakdown in your reporting or analysis, do not preaggregate the data. Report sections will automatically provide an aggregated summary view, and component levels of detail data can be reached using drill-down tools.

If your data set is potentially very large, or incorporates very discrete levels of transactional data that do not apply to your analysis, it may be best to preaggregate the data at the server to return a more manageable data set. Preaggregating data in your query assumes that you have a clear idea of the data to look at and a good conceptual understanding of relational databases. If you are unsure about preaggregation, process the query without applying data functions. If you find that it would be better to preaggregate, return to the Query section and apply data functions to the query.

Use data functions to preaggregate data as it is retrieved from the database. The following table lists the prebuilt data functions that you can apply to items in the Request line.

Tip: Aggregation is manifested in different ways in the Query and Reporting sections.

When using data functions, remember that with the exception of counts, data functions are applied almost entirely to numeric data items and the results are computed with respect to dimensional, nonnumeric items on the Request line, such as name and date items as in the following examples.
Example

Query 1 includes only items for Region and Amount_Sales. The data function Sum is applied and the data returned consists of one row for each region with an aggregate sum for that region in the Amount Sales column.

<table>
<thead>
<tr>
<th>Region</th>
<th>Amount Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>Americas</td>
<td>$89,755,836</td>
</tr>
<tr>
<td>Asia Pacific</td>
<td>$29,556,011</td>
</tr>
<tr>
<td>Europe</td>
<td>$34,569,833</td>
</tr>
</tbody>
</table>

Example

In Query 2, the item Fiscal_Year is added to Query 1, breaking out rows for each state/fiscal year combination with Units totaled on a per state, per year basis.

<table>
<thead>
<tr>
<th>Fiscal Year</th>
<th>Region</th>
<th>Amount Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>1999</td>
<td>Americas</td>
<td>$11,411,979</td>
</tr>
<tr>
<td>1999</td>
<td>Asia Pacific</td>
<td>$4,250,710</td>
</tr>
<tr>
<td>1999</td>
<td>Europe</td>
<td>$5,066,636</td>
</tr>
<tr>
<td>2000</td>
<td>Americas</td>
<td>$57,343,857</td>
</tr>
<tr>
<td>2000</td>
<td>Asia Pacific</td>
<td>$25,305,302</td>
</tr>
<tr>
<td>2000</td>
<td>Europe</td>
<td>$29,503,197</td>
</tr>
</tbody>
</table>

Example

In Query 3, the Product_Line Name has been included and the data function is changed to Average. The number of rows increased, with data summarized as an average per state, per year, per product line.

<table>
<thead>
<tr>
<th>Fiscal Year</th>
<th>Region</th>
<th>Product Line Name</th>
<th>AVG(Amount Sales)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
To apply a data function in the Query section:

1. Select an item on the Request line.
2. Select Query, then Data Function and select the desired function.

The item is renamed to reflect the data function you selected. For example, \( \text{SUM(Units)} \) or \( \text{AVG(Amount_Sold)} \).

When the Query is processed, the data is returned from the server in aggregate form.

To remove a data function in the Query section:

1. Select the item on the Request line.
2. Select Query, then Data Function, and then None.

Appending Queries

When you need to view and merge multiple queries in a combined results set, there are four query operators that allow you to merge two or more separate queries. For example, you may need to merge the results of amount from sales by item, plus units from state sales by state. The operators and their functions are:

<table>
<thead>
<tr>
<th>Query Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>⬤ Union</td>
<td>All distinct rows selected by either query are retrieved. No duplicate rows are retrieved.</td>
</tr>
<tr>
<td>⬤ Union All</td>
<td>All rows selected by either query, including duplicate rows, are retrieved.</td>
</tr>
<tr>
<td>Query Operator</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------</td>
</tr>
<tr>
<td>[Intersect]</td>
<td>All distinct rows selected by both queries are retrieved.</td>
</tr>
<tr>
<td>[Except]</td>
<td>All distinct rows selected by the first query but not the second query, are retrieved. (Oracle database servers refer to the Except operator as “Minus.”)</td>
</tr>
</tbody>
</table>

To specify an evaluation order, you can click \( (\) to add parentheses to an expression which includes two queries or more. The default evaluation order is left to right.

**Note:** If your database supports the Intersection and Except operators, but they are not available in the Operator drop-down list, check the Allow SQL-92 Advanced Set Operations connection preference.

The rules governing the use of these operators are:

- The number of columns in the Select clause in both queries must be equal.
- The data type returned in the columns in both queries must match.

For example, if Column 1 in the first query is a date, Column 1 in the second query must also be a date.

**Note:** Items on the Union line can be repositioned to see the results of different intersections.

To append a query:

1. **Verify data types and associated column(s).**
   This ensures that you know how to merge data in the second query.

2. **Build the Request line.**
   Add server and local filters, data functions, and computations to the query as needed.

3. **Select Query, then Append Query.**
   An Operator drop-down list and a second query tab is added below the Request, Filter, and Sort lines. The drop-down list shows whether the queries are linked by way of a union, a union all, an intersection, or an except.

4. **Build the second query.**

5. **To merge multiple queries, select the operator to use from the Operator drop-down list.**

6. **Click Process.**
   You can have Interactive Reporting generate automatically the join path required by the context of the query by using the automatic join path feature. This feature eliminates the need for you to predefine any join paths, because Interactive Reporting determines the paths. When multiple paths are available, you are prompted for which one to use.
Using Local Results

Local results are a snapshot of a Results section shown in topic format. They are used to add the results of one query to another in an Interactive Reporting document file.

➤ To use local results, click anywhere in the Catalog pane and select Local Results on the shortcut menu.

A Table catalog named Local Results is added to the Catalog pane.

➤ To append results from two different Result sets into one single section:

1 Verify that each of the queries are in two different query sections.

The above step is applicable whether the queries are based on the same or different databases.

2 In each of the results section, create a computed column with the values 'A' and 'B' respectively (one per section). These values provides you with a column to join on that will never match.

3

4 Build a new query with the two local results and use the join type of 'Outer' on these computed columns.

Use all columns from both results sections.

5 Process the query

6 Once you have all the results back, create the “original” columns in your final results section by using a decode logic.

For example, you might use: if (resultsa.column1!=null) {resultsa.column1} else {resultsb.column1}

Limitations of Local Results

Since local results are maintained on the desktop and not by the database server, there are limitations when using Local Results. The following functions are not available when using Local Results:

- Filters, computed items, data functions, or query properties to further analyze the dataset
- The following Query Options menu governors (disabled if only local results topics make up the Query):
  - Returning unique rows
  - Row filter
  - Time filter
  - Auto-process
  - Custom Group by
- Query filters on Local Results Topic Items
- More than one Filter Local Join
- Filter Local Joins used with Local Joins
Meta topics

Access or change properties for Local Results Topic Items

Append Query features of Unions or Intersections with Local Results Topic Items

Process to table a query

Note: A query based on local result topics will not perform as well as the equivalent database query.

Processing Order

When using process all, the query producing the results may be processed twice if the query using its results are listed first in the section catalog. It is also possible for the query using the local results to use stale data if it was saved with results, and the query that produced them is reprocessed. To prevent this from happening, you can defined the order in which queries are processed. For more information, see Query Processing Order.

Using Stored Procedures

Stored procedures are precompiled, complex queries that are executed on a database server and maintained by a database administrator. Stored procedures execute very quickly and are usually created to accomplish tasks that SQL cannot do alone. Interactive Reporting treats stored procedures as locked standard queries and does not allow you to modify the procedures.

You can use Interactive Reporting to process stored procedures through Open Client or ODBC, collect the results, and generate reports as you would with a standard query. Stored procedures can be loaded from your desktop and appear as a query object in the Content pane.

Note: ODBC only. Interactive Reporting supports stored procedures that return results. This support is contingent on the driver and database. The driver and database must support the required ODBC calls, including SQL Procedures to retrieve a list of available procedures and SQLProcedureColumns which identify the parameters required to execute the procedure. For Oracle, results are recognized in ODBC by specifying reference cursor parameters when the procedure is created. The Interactive Reporting database connection file must specify the database as “ODBC” rather than “Oracle” to work properly.

Note: The ODBC driver must recognize the ODBC syntax for calling procedures: {call <procedure name> (parameter list)}. If the procedure has no parameters, the parentheses surrounding the parameter list are optional. Interactive Reporting does not insert empty parentheses in the call to execute the procedure. In addition, the driver must accept literal values for any specified parameter. Drivers that require parameter markers, for which values are provided when the procedure is executed, are not currently supported.
To open a stored procedure:

1. **Select Query, then Stored Procedures.**

   The Stored Procedures dialog box appears.

2. **Select the database owner name that contains the stored procedure.**

   Any stored procedure to which you have been granted access is displayed in the Stored Procedures list.

3. **Select a stored procedure from the list and click Load.**

   The stored procedure appears as an icon in the Content pane. No items appear on the Request line until the stored procedure is processed.

To process a stored procedure:

1. **Click Process.**

   If the stored procedure calls for user input, a dialog box appears and prompts you with up to 10 entry fields. If more than 10 arguments are required, successive dialog boxes appear.

2. **If an argument dialog box is displayed, enter appropriate values as arguments to the stored procedure.**

   The arguments supplied are similar to variable filters. If necessary, see your database administrator for clarification on the arguments needed to process a particular stored procedure.

3. **If the stored procedure queries the database, the database server returns data to the Results section and the adds items to the Request line.**

### Setting Query Options

When working with very large or unfamiliar databases, you may occasionally process a query that takes a long time to run or returns more data than is manageable. To prevent problems under these conditions, set query options before processing.

To set query options:

1. **Select Query, then Query Options.**

   The Query Properties dialog box is displayed.

2. **Select the desired restrictions for the current query and click OK.**

   - **Return Unique Rows**—Eliminates duplicate rows from the data set retrieved by the query. It is not a regression. If the data source for a query is a local results table, even though this feature is enabled, the Unique Rows filter is not enforced.

   - **Return First ___ Rows**—Filters the number of database rows retrieved to the number entered.

   - **Time Limit ___ Minutes**—Filters the amount of time the query is allowed to run to the number entered. Seconds are entered as a decimal number. Time filters work for
asynchronous database connections and cancel at the earliest opportunity for nonasynchronous connections.

- **Auto Process**—Specifies the current query as a Standard Query to be processed automatically on download from the Repository (Designer only).
- **Custom Group By**—Customizes the Group By criteria used to compute aggregate Request items, with selected items not factored into the grouping. This feature is available only when a data function is placed on a Request item.

**Editing with SQL**

You may wish to use SQL to create or edit your query. Working with SQL involves:

- **Editing with SQL**
- **Importing SQL Files**
- **Reviewing the SQL Log**

**Editing with SQL**

If you are familiar with Structured Query Language (SQL) and prefer to edit your own query, activate the Custom SQL window and edit the statement directly. Interactive Reporting sends the contents of the Custom SQL window to the server when you click Process.

You can use the Custom SQL window to edit query syntax, or to troubleshoot a query which is not processing correctly. With this function, you can only edit the FROM and TO clauses of the SQL statement.

**Tip:** You must leave the Custom SQL window open for Interactive Reporting to use your edited SQL statement when processing the query.

➢ To open the Custom SQL window, select **View**, then **Custom SQL**.

You can also type edit directly in the Custom SQL window.

Reset restores the SQL statement contained when the Custom SQL dialog box was opened. Use the Window menu to display or hide the Custom SQL window once it is opened.

➢ To save your edits and close the window, you can click **Close** or click the checked Custom SQL feature on the View menu.

**Determining the Number of Rows from and SQL Statement**

To determine the number of rows retrieved or if data was selected in an SQL statement, use one of the following techniques:

**Technique 1**

begin-report
do count_rows
do return_true
end-report

**Technique 2**

begin-procedure count_rows
move 0 to #count ! This is optional unless main1 is called more than once
begin-select
  column1
column2
column3
add 1 to #count
from table1
end-select
if #count > 0
  show 'Number of rows selected was ' #count edit 999
else
  show 'No rows selected'
end-if
end-procedure main1

begin-procedure return_true
begin-select
  'true' &true
column1
column2
column3
from table1
end-select
if &true = 'true'
  show 'Selected at least one row'
else
  show 'No rows selected'
end-if
end-procedure

**Importing SQL Files**

The Import SQL file feature allows you to take a complete SQL statement from a text file, import it into a query, and retrieve the data set from the database server. When the file is imported, it is scanned to determine the number of columns that will be returned by the SQL, with the request line becoming populated with a column indicator for each of the columns. Using this feature, you can take advantage of SQL statements you have already written.

Before importing a SQL file, verify the following:

- You are connected to a database server and are working in the Query section.
- There must be no tables in the Contents pane.
- The SQL file to be imported must begin with a SELECT statement, and you should know the number of columns to be displayed in the Results section.

Once the SQL file is imported into the query, you cannot edit it, drag items from table onto the Request line (however, you can specify a user-friendly name for the Request line item and identify its data type), use the custom SQL feature, or display its properties.
To import an SQL file:

1. **Select File, then Import Data File, and then SQL.**
   The Import SQL File dialog box is displayed.

2. **Select the desired SQL file and click Open.**

3. **In the Number of Columns dialog box, enter the number of columns that you want the query to run and click OK.**
   For extremely complex SQL statements, Interactive Reporting needs to be advised on how many columns to expect from the database. The default number in the dialog box is the "best guess" option.
   Interactive Reporting inserts the SQL statement directly into the content, nested between the header and footer "Imported SQL Statement." If the statement is larger than the entire Contents pane use the scroll keys to view it.

4. **Double-click a Request line item.**
   You can also click a column on the Request line and click Properties on the shortcut menu.
   The Item Properties dialog box is displayed.

5. **Specify a name for the item in the Name field.**

6. **Associate the item with a data type by selecting a data type from the pull-down list and click OK.**

7. **Click OK to retrieve the results.**
   The column data types default to string unless they are manually changed.

To refresh or delete a column name:

1. **Edit the SQL in a text editor, such as Windows Notepad or vi in UNIX and save it.**

2. **In the Query section, right-click on the area labeled '--Imported Sql Statement--'.**

3. **Select either Refresh or Delete as needed.**

To edit SQL that has no associated file:

1. **Select View, then Query Log.**

2. **Process the query containing the imported SQL text.**

3. **Copy the SQL statement from the Query Log window and paste it into a new text file using your editor.**
   When you refresh an imported SQL statement, Interactive Reporting attempts to open the same file used when you first imported the statement. If that file no longer exists, you are prompted to select a new file. Because editing the SQL statement might change the SELECT list (which appears on the Request line) you must redo any edits made to the name and/or data type of those items after refreshing the SQL statement.
Reviewing the SQL Log

When you process a query, Interactive Reporting translates your graphical request into an SQL or other query statement (MDX – OLEDB and OLAP servers, and Essbase also have their own languages), which in turn is forwarded to the database server. You can view the generated query statement in any section by displaying the Query Log window.

➤ To display the SQL Log, select View, then Query Log.

When this feature is active, the Query log is treated as an open window. You can bring it to the front by clicking Windows, then Query Log. The Query log closes and drops from the Window menu when it is deactivated on the View menu.

➤ To save the contents of the SQL Log to a file, select File, then Export, and then Query Log.

The Export File dialog box prompts you to enter document and directory names for the new file.

Importing Data Files

An alternate method of "querying" is to import data from a file. Interactive Reporting can import files in Microsoft Excel (.XLS), and in both comma (.CSV) and tab (.TXT) delimited text formats.

When you import data, the content of a file is delivered to the desktop data cache and displayed in the Results section. You can use imported data as you would the results of a query to build reports and perform data analysis.

➤ To import a data file:

1 Select File, then Import Data File, and then Data File.

The Import File dialog box is displayed.

2 Navigate to the location of the file.

On a Windows operating system, select the correct type of the import file type menu to make the file easier to find.

3 Select the file and click OK.

The data from the imported file is displayed in columnar Results Format.

Setting Data Type Properties

You may want to confirm or change the data type of an item to preserve the precision of a mixed data type computation, or to change the way a data item is handled for example interpreting numbers as strings).

Attention to data types is most important when computing items in the Query section. In this case the database server performs the computation, and the Interactive Reporting may receive
the computed item with an unanticipated data type. To ensure that server correctly handles data computations, you should set the data type when performing mixed-data type computations.

Local calculations (Results, Pivot) are handled internally, and adjustment between 16- and 32-bit integers, for example, can be handled safely using the automatic or number data type specification.

To set data type properties for a request item:

1. Select a request item and choose Properties from the shortcut menu. The Item Properties dialog box is displayed.
2. Click Options and select a datatype from the Datatype drop-down list.
3. Click OK.

The following table describes the supported data types:

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatic</td>
<td>A data type is determined automatically given the data type of the reference items and the computations performed.</td>
</tr>
<tr>
<td>Blob</td>
<td>Binary large object which is truncated at 64,000 bytes. Blob data types can include image formats such as: JPEG, .BMP, .GIF, and .PNG. See also “Picture (BLOB Image) Support” on page 119.</td>
</tr>
<tr>
<td>Byte</td>
<td>Variable data type of length determined by a single byte of computer storage. Bytes can store numeric values from 0 to 255, or a single text character.</td>
</tr>
<tr>
<td>Date</td>
<td>Calendar date in server default format (typically mm/dd/yy).</td>
</tr>
<tr>
<td>Integer (16-bit)</td>
<td>Retains a 16-bit value (2 bytes). A 16-bit integer stores integer values from 0 to 16,777,216, and signed integers between +8,388,608 and -8,388,608.</td>
</tr>
<tr>
<td>Integer (32-bit)</td>
<td>Retains a 32-bit value (4 bytes). A 32-bit integer stores integer values from 0 to 4,294,967,296, and signed integers between +2,147,483,648 and -2,147,483,647.</td>
</tr>
<tr>
<td>Long Text</td>
<td>Character data (long text) exceeding 25 bytes (use the string data type for text strings up to 255 characters). The maximum long text retrieved is 4000; characters anything greater than that is silently truncated.</td>
</tr>
<tr>
<td>Packed Real</td>
<td>Real numbers packed for use with EDA middleware. The results in the Interactive Reporting are the same as real numbers.</td>
</tr>
<tr>
<td>Real</td>
<td>Decimal numbers up to 5 positions right of the decimal.</td>
</tr>
<tr>
<td>String</td>
<td>Text strings to a maximum length of 256 characters.</td>
</tr>
<tr>
<td>Time</td>
<td>Time in format set by user preference.</td>
</tr>
</tbody>
</table>

Interactive Reporting supports very large character data type column fields, such as Microsoft SQL Server text data type files, and IBM's DB2 Character Large Object (CLOB) data types.

The DB2 CLOB data type is SQL99-compliant and is used to store very large variable-length character strings (up to 10MB). Its length property is displayed with a default value of 4000 characters; however, this may be updated to any value between 1 and 4000 by an Interactive Reporting user. The length property determines the maximum length of the CLOB data string that can be displayed after a database query. CLOB column fields with data that are greater than 4000 bytes in length are truncated after the 4000th character.
<table>
<thead>
<tr>
<th>Data Type</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>TimeStamp</td>
<td>Date/time combination in format set by user preference.</td>
</tr>
</tbody>
</table>

**Picture (BLOB Image) Support**

Pictures can be queried from a relational database if they are image data collected as a BLOB data type and stored as a binary unit in the database management system with ODBC connections. BLOB image files available to include: .JPEG, .BMP, .GIF, and .PNG image formats. Pictures can be dragged and dropped from a query table to the request line, displayed as strings in columns in the Results and Table sections, or shown graphically in the bands of the Report section. Interactive Reporting dashboard developers can select images from Results sets, not just static images from the file system. Wherever applicable, computed items and some functionality, such as the group by clause in SQL, unions and outer joins, use the primary key column.

The following restrictions apply to BLOB images in Interactive Reporting:

- The ODBC connection must use a server-defined join (see “Metadata Definition: Joins” in Interactive Reporting Help).
- BLOB images cannot be retrieved using a stored procedure.
- It is not possible to determine the data type of an image in advance.
- BLOB images cannot be used with the Process to Table feature.
- BLOB images cannot be used with the Import SQL feature.
- In the EPM Workspace if a user attempts to add a BLOB column to the filter, there is no popup message to indicate that the functionality is not supported.
- Text Wrap and Suppress Duplicates features cannot be applied to BLOB columns.
- If a BLOB column is placed in the Query section and processed in a pre 9.3 release Interactive Reporting document file, images are not backward compatible.

See also:

- Adding Pictures in Query
- Adding Tooltips to Pictures
- Working with Pictures in Results and Tables
- Working with Pictures and Computed Items
- Working with Pictures in Reports

**Adding Pictures in Query**

In the Query section, once BLOB images are retrieved as pictures, they are treated like any other table topic item. They can be dragged and dropped from the table directly to the request line. Only pictures in tables that have a primary key defined are included in a query. Pictures cannot
be sorted in the Query section. If you add a non-BLOB image to the request line, a broken link to the image is displayed in the report section.

**Adding Tooltips to Pictures**

A tooltip can be associated with a picture, which is a tiny box that is displayed next to the picture and identifies it in the Results and Table section. For Dynamic pictures, the tooltip field accepts a column name which represents a non-BLOB column item on the request line used to describe the picture. If no tooltip is selected for a BLOB image, the picture column text shows <<Picture: unique id>>. In this case, the unique id is the tooltip text. Otherwise, the corresponding primary key value is shown. For static images, the tooltip is text that you specify on the properties dialog of a picture.

To associate a tooltip with a BLOB image:

1. **Select the BLOB image on the request line.**
2. **Right-click and select Properties.**
   - The Items Properties dialog box is displayed.
3. **Click Options.**
   - The Datatype drop-down and Tooltip drop-down are displayed. The Tooltip drop-down contains a list of all non-blob items on request line, and a "None" (default) option.
4. **From the Tooltip drop-down, select the non-BLOB image to use as the tool tip.**
5. **Click OK.**

**Working with Pictures in Results and Tables**

A column is displayed in the Results and Table sections for every picture with a BLOB data type on the request line. Descriptive text is displayed instead of the image in each cell in the format <<Picture:unique id>>. The unique id is tooltip text if any has been specified, otherwise, the unique id is the primary key value.

**Working with Pictures and Computed Items**

A picture function is available in Computed Items so that you can associate an image with a computed column. In the Results and Table section, the picture is shown as descriptive text instead of the image in each cell in the format <<Picture:unique id>>. The unique id is tooltip text if any has been specified, otherwise the unique id is the Resource name. In the Reports section, the computed item containing the picture can be shown graphically. For example you could use an “if else” statement to show an image if a certain set of conditions are met, and another criteria if the condition is not met:

```sql
if (Amount_Sales >= 10000) {Picture("C:\graphics\smile.gif")}
else{Picture("C:\graphics\crying.gif")}
```
**Working with Pictures in Reports**

The Report section can include pictures obtained from the database (BLOB datatypes) by using an embedded object (Results and Tables) or it can use static images from the Resource Manager. Pictures can be dragged from the Catalog and dropped into the report layout components:

- report table
- report body
- report group header
- report page header of footer

**Note:** You cannot drag a picture into a Table Facts column.

Once a picture has been added to the report, all images are sized to fit the bounding rectangle defined in the report, and pictures can be resized as needed.

Pictures are sorted by the their underlying textual unique id (i.e. the text displayed in a table/result section), which is particularly useful when they are added to the report group headers (by way of the data layout).

**Resource Manager**

The Resource Manager is a user interface utility used to load, manage and share pictures in the Interactive Reporting document file. By storing one copy of the image and referencing the copy as a resource when used elsewhere, the Resource Manager reduces the size of an Interactive Reporting document file, which in turn reduces the memory footprint required to open some documents.

For example, a company logo might appear several times in an Interactive Reporting document file, in Reporting headings and in Dashboards. With the Resource Manager, only one copy is stored and reused by reference to its resource name.

A picture can be used as a resource only if it is added to the Resource Manager. Users can still import images directly in the Interactive Reporting document file, but in this case, they are considered “dynamic” pictures.

Each Interactive Reporting document file has its own Resource Manager. A new picture is added to the Resource Manager from disk by using an Import feature. Once imported, pictures can be viewed in several formats:

- listed
- in detail
- preview
- thumbnail

Resource Manager pictures can also be merged, renamed and deleted.
In pre 9.3 release Interactive Reporting document file, Resource Manager images are not backward compatible when images are:

- Converted from .BMP format to .PNG format
- Merged using the Resource Manager Merge or Merged All feature.
- Images are dragged from the Resource Node in the Report section or Dashboard section.
- Duplicated from the Report or Dashboard sections.
- Copied from the Report or Dashboard sections.

When an Interactive Reporting document (BQY) is created and saved using 9.3.x releases, all images are saved in a centralized format (the Resource Manager). As a result, the images cannot be read or displayed in prior releases of Interactive Reporting. This is because the relocation and rationalization of the images are not understood by releases prior to 9.3.x.

The exception is images from an Interactive Reporting document file saved in a pre 9.3 Release, which have been referenced once: not copied, merged or duplicated, that is, placed in Dashboard or Report workspace as a Picture graphic in traditional fashion.

To retain compatibility with earlier releases, you can run the RevertImageResources script. This script provides a facility to undo the relocation and image merging, and thereby return the Interactive Reporting document file to the pre-9.3 pre-Resource Manager format. This script is included with the 9.3. Release of Dashboard Development Services. It can also be downloaded from the Hyperion Developer Network. This script can only be used on the desktop, and not in EPM Workspace, as it relies on the COM feature of Interactive Reporting Studio.

Certain new features make documents ineligible for conversion to the non-Resource Manager format: for example if the 9.3 Interactive Reporting document file has Bubble and Scatter charts, then these are lost when the Interactive Reporting document file is opened in the older version of Interactive Reporting Studio. For information on running the RevertImageResources script, see the Hyperion System 9 BI+ 9.3. Readme.

To import a picture into the Resource Manager:

1. **Select Tools, then Resource Manager.**
   The Resource Manager dialog is displayed.

2. **Click  ![Image Icon].**
   The Select Image dialog is displayed.

3. **Navigate to the graphic and click Open.**
   The image is listed in the Resource Manager.
   If you import a picture that is already in the Resource Manager, a warning message is displayed.
   A picture in .BMP format is converted to .PNG format (Portable Network Graphic) before it is loaded into the Resource Manager.
To change the view of pictures in the Resource Manager:

1. **Click**.

   A list of available views is displayed.

2. **Select the view of pictures on the Resource Manager dialog.**

   Available view include:
   - **List**—Displays picture names only.
   - **Details**—Displays picture name, size, format type and date imported.
   - **Preview**—Displays actual picture.
   - ** Thumbnails**—Displays reduced size version of all pictures.

To replace a picture:

1. **Select the picture.**

2. **Right-click and select Replace.**

   The Select Image dialog is displayed.

3. **Navigate to the new picture and click OK.**

   An alert shows all sections in which the old picture is referenced. Picture properties set up for the picture in the Resource Manager are only used for new instances of the picture.

4. **Click Replace.**

To merge a duplicate picture in the Resource Manager:

1. **Select the picture.**

2. **Right-click and select Merge Duplicates.**

   An alert shows all sections in which the duplicate pictures are referenced.

3. **Click Merge.**

   The first selected duplicate is saved in the Resource Manager and all other duplicates are deleted. In this case, any references in the Dashboard and Report sections must be updated.

To merge all duplicate pictures in the Resource Manager:

1. **Select the picture.**

2. **Right-click and select Merge All Duplicates.**

   Duplicates must have an identical name, size and type to use this feature.

   An alert shows all sections in which the duplicate pictures are referenced.

   All but the first selected picture is deleted.

3. **Click Merge All.**
To delete a picture from the Resource Manager:
1 Select the picture.
2 Right-click and select Delete.
3 Click Delete.

Once a picture is deleted, all sections that refer to the picture show a broken picture image. That is, the picture is represented by a red-cross icon.

Deleting a picture from the Dashboard or Report sections only deletes the reference and not the actual picture.

To rename a picture in the Resource Manager:
1 Select the picture.
2 Right-click and select Rename.

A picture name must be unique and cannot be empty.

Avoid using the following characters when renaming a picture: _ " / : * ? “ < > | +

**Picture General Properties**

Use the Picture General Properties to display attributes of a picture resource. Picture attributes are determined when the picture is loaded to the Resource Manager. There are no configurable properties on this dialog.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Displays the file format of the picture.</td>
</tr>
<tr>
<td>Size</td>
<td>Displays the size of the picture.</td>
</tr>
<tr>
<td>Imported</td>
<td>Displays the date and time on which the picture was imported to the Resource Manager.</td>
</tr>
<tr>
<td>Data Source</td>
<td>Displays the file path of the disk where the picture is located.</td>
</tr>
<tr>
<td>Configurable options</td>
<td>Displays the picture percent scale (height and width), and picture effect (none, stretch, clip and title).</td>
</tr>
</tbody>
</table>

**Note:** The attributes shown on this dialog are informational only, and cannot be changed on this dialog.

**Query Menu Command Reference**

The table below provides a quick reference to the commands available on the Query menu and lists any related shortcuts.
Table 22  Query Menu Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Keyboard Shortcut</th>
<th>Shortcut Menu</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process Results To Table</td>
<td>Allows you to create a table in the database to store your results set.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Estimate Query Size</td>
<td>Queries the database to see how many records your query will retrieve.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Show Remarks</td>
<td>Displays any remarks recorded about a topic or topic item.</td>
<td>Ctrl+I</td>
<td>*</td>
</tr>
<tr>
<td>Add Request Item(s)</td>
<td>Adds the selected topic item to the Request line.</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Add Filter(s)</td>
<td>Allows you to create a filter for the selected topic item.</td>
<td>Ctrl+L</td>
<td>*</td>
</tr>
<tr>
<td>Add Sort(s)</td>
<td>Adds the selected topic item to the Sort line.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add Computed Item</td>
<td>Allows you to add a new data item derived from server-side calculations</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>performed on an topic item.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Functions</td>
<td>Applies a prebuilt data function to the selected Request item.</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Variable Filter</td>
<td>Designates the selected filter item as variable, which causes Interactive</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>Reporting to prompt the user for the filter values when the query is</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>processed.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Customize Filter</td>
<td>Allows you to control access to the features on the Filter dialog box.</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Append Query</td>
<td>Allows you to combine two or more queries in one Results set.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stored Procedures</td>
<td>Loads a stored procedure and displays it as a query object in the Content</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>pane.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Query Options</td>
<td>Opens the Query Properties dialog box where you can specify options for</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>rows returned, time filters, and so on.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Insert Query**

Use the Insert New Query dialog box to insert a new query. You can insert the query associated with a master data model, or log on using the current Interactive Reporting database connection file. Additionally, Interactive Reporting Studio users can select another Interactive Reporting database connection file.

➢ To insert a new query:

1. **Select Insert**, then **New Query**.

The Insert Query dialog box is displayed.

2. **Select Master Datamodel** to use the master data model associated with the query.

3. **Select Logged on connection** to connect to the database using the current Interactive Reporting database connection file.

4. **Select No connection** not to connect to any database.
Click Yes.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Master Datamodel</td>
<td>Each time a new query is inserted into an Interactive Reporting document file that contains a master data model, you are prompted to link the new query to the master data model. The benefit is that any changes to the master data model get propagated to all dependent queries that are based on the master data model.</td>
</tr>
<tr>
<td>Logged on Connection</td>
<td>Uses the current Interactive Reporting database connection file.</td>
</tr>
<tr>
<td>No connection</td>
<td>The new Interactive Reporting document file is not associated with an Interactive Reporting database connection file, allowing you to work offline.</td>
</tr>
<tr>
<td>Options</td>
<td>Displays the second Insert Query dialog box. Use this dialog to specify a new or recent Interactive Reporting database connection file. This command is not available in Interactive Reporting Web Client.</td>
</tr>
</tbody>
</table>

To create a new Interactive Reporting database connection file:

1. **Select Insert, then New Query.**
   
The Insert Query dialog box is displayed.

2. **Do not select any options on the first Insert Query dialog and click No.**
   
The second Insert Query dialog is displayed.

3. **Select A New Database Connection File and click OK.**
   
The Database Connection Wizard is displayed.

4. **Follow the instructions given in the wizard.**
   
   For more information, see “Creating Interactive Reporting Database Connections (.oces)” in the Interactive Reporting Help.

To create a new Interactive Reporting document file using a recent new Interactive Reporting database connection file:

1. **Select Insert, then New Query.**
   
The Insert Query dialog box is displayed.

2. **Do not select any options on the first Insert Query dialog and click Options.**
   
The second Insert Query dialog is displayed.

3. **Select Recent Database Connection Files.**

4. **Select an Interactive Reporting database connection file from the Recent Database Connection Files list.**

5. **If you don’t see the desired Interactive Reporting database connection file, click Browse to display the Select Connection dialog box.**

6. **Navigate to the desired Interactive Reporting database connection file and click Open.**
   
The Connection Password dialog box is displayed.

7. **Type your user name in the Host User field and password in the Host Password field, and click OK.**
Tip: If you do not have the right Interactive Reporting database connection file to connect to a particular database, ask your administrator to provide the Interactive Reporting database connection file or to help you create an Interactive Reporting database connection file.

To create a blank Interactive Reporting document file with no Interactive Reporting database connection file, select No Connection and click OK.

Process Results to Database Table

Use the Process Results to Database Table dialog box to create a table in the database to store your results set. Items on the Request line become the column headings of the new table, and you can append new columns to the table and query it as needed.

Note: The Interactive Reporting database connection file and database to which you are connected determine whether you can use this feature. You must have Create and Insert privileges on the database to process results to a database table.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table Name</td>
<td>The name of the new table you want to create or the name of an existing table to which you want to append columns. You can create tables under your own owner name or under different databases or owners. If you do not have the correct privileges or do not specify an alternate location, the table is created under your own owner name. Use the format DATABASE.OWNER.TABLENAME to specify alternate names.</td>
</tr>
<tr>
<td>Create Table</td>
<td>Creates a new table in which Request items form columns.</td>
</tr>
<tr>
<td>Append to Table</td>
<td>Append Requests items as new columns in a table.</td>
</tr>
<tr>
<td>Grant Access to Table</td>
<td>Enables either everyone or specific users to access the new tables. Type PUBLIC or specific user IDs (separated by commas) for each user who should have access to the table. If not selected, access to the table is filtered to your own user ID.</td>
</tr>
<tr>
<td>Tables Created by</td>
<td>Lists tables created by Interactive Reporting under your user account.</td>
</tr>
<tr>
<td>Delete Table</td>
<td>Deletes a selected table.</td>
</tr>
</tbody>
</table>
Results Section

When you process a query or import data, Interactive Reporting retrieves data to your desktop and displays it in the Results section. Although the query may have accessed several different database tables, the results set is displayed as a single table. Each requested item is displayed as a column in the table and each database record is a row. The Status bar shows the date and time the results set for a particular Results section was last processed (or imported).

Use the Results section to:

- Verify that your query returned the correct information.
- Refine and extend the data set by applying filter conditions or create new computed or grouped items.
- Sort or use text and column formatting features to enhance the appearance of data results.
- Add summary totals or subtotals and compute them with data functions.
- Print or export the retrieved data to other applications.

All reports, including tables, pivots, charts, and those created using the Report Designer, are based on the data that is retrieved to the Results section.

Understanding Data Types

To effectively work with the data in the Results section, you need to understand how Interactive Reporting handles data. Certain functions can only be used on certain types of data.
The Results section formats data in table format. A table is either a fact table or a dimension table. A table is a fact table if it contains at least one fact column. A dimension table contains only dimension columns.

A fact is a quantifiable entity, such as a value or unit of measure. Facts are the numeric values in a relational database that are available for analysis.

A dimension is a descriptive item, such as a name or label.

Results Data Layout

Data queried into the Results section automatically loads into the Results data layout to provide an immediate view of items included in the query. This allows you to plot, view, and manipulate the data set returned with the query.

To toggle the Results data layout, click Data Layout on the command line.

Zooming Results

You can resize the Results section to fit the Contents pane or the printed page.

To resize the Results section:

1. Select View, then Zoom.
2. Select a size from the sub-menu.

100% is the default size, which prints to fit 8-1/2 * 11 paper.

Formatting Results Columns

Interactive Reporting provides the following formatting options for columns in the Results section:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remove Column</td>
<td>To remove a selected column from the table (and data layout), click Remove Column on the shortcut menu. If you remove an item from the Contents pane, it is removed from the data layout and the data set entirely. You should approach item removal with caution since computed items may draw data values from the deleted item and the Chart, Pivot and Table sections reference the Results set. If a computed item references the column to be deleted, a Report Refresh dialog box is displayed and prompts you to either keep the reference or to remove it.  Note: The Remove Column option is available for detail and fact columns.</td>
</tr>
<tr>
<td>Select Column</td>
<td>To select a column, click anywhere inside the column.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Size Column Width</td>
<td>By default Interactive Reporting truncates Results columns evenly and without regard to the length of data values. Numeric data that does not fit is replaced with pound signs (#). To auto-size a column so that all values display, choose Format, then Column, and then Auto Size Width. For narrow or uniform columns to display values without truncation, wrap the text with each column item. To apply the standard width to a column, click the column and choose Format, then Column, and then Standard Width. To resize a column manually, click the column, place the cursor over the right edge of the column, and drag the right border to a new position.</td>
</tr>
<tr>
<td>Modify Column</td>
<td>To modify a computed column or a grouped column, select the column and choose Results, then Modify Column.</td>
</tr>
<tr>
<td>Hide/Show a Column</td>
<td>To hide a column, select the column and click Hide Column on either the shortcut menu or the Results menu. To unhide a column, click anywhere within the Contents pane and click Unhide Column on either the shortcut menu or the Results menu. When the Unhide Column dialog box is displayed, select the desired column and click OK.</td>
</tr>
<tr>
<td>Hide/Show Column Titles</td>
<td>To hide or show column titles, choose Format, then Column Titles. When column titles are hidden, the Column Titles option is unchecked. When column titles are displayed, the Column Titles option is checked.</td>
</tr>
<tr>
<td>Size Titles</td>
<td>To apply a custom height to a title, drag the bottom edge of a title row to the new position. To apply the standard height to a title row, choose Format, then Row, and then Standard Height.</td>
</tr>
<tr>
<td>Suppress Duplicates</td>
<td>To suppress the duplicate values for a column, select the column and click Suppress Duplicates on either the shortcut menu or the Format menu. Use this option to display the first instance in the column of a duplicate value when individual database records include redundant information. This feature is especially useful if records are associated with the same date, location or customer. Note: This option is available for detail and fact columns.</td>
</tr>
<tr>
<td>Wrap Text</td>
<td>To wrap text within a column, select the column and click Text Wrap on either the shortcut menu or the Format menu. Note: This option is available for detail and fact columns.</td>
</tr>
<tr>
<td>Group Columns</td>
<td>To merge dimension labels into new groupings and aggregate the associated data, click Add Grouping Columns on either the shortcut menu or the Results menu.</td>
</tr>
<tr>
<td>Sort Ascending</td>
<td>To sort column values in ascending order, click Sort Ascending on the Results menu, shortcut menu, or standard toolbar.</td>
</tr>
<tr>
<td>Sort Descending</td>
<td>To sort column values in descending order, click Sort Descending on the Results menu, shortcut menu, or standard toolbar.</td>
</tr>
<tr>
<td>Move Column</td>
<td>To move a column, click the column in the Contents pane and drag it to a new position. The corresponding data layout item shifts to reflect the change.</td>
</tr>
</tbody>
</table>

### Formatting Results Rows

Interactive Reporting offers the following formatting options for Rows in the Results section:
Table 25  Results Rows Formatting

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hide/Show Row Numbers</td>
<td>To hide row numbers, choose Format, then Row Numbers. (The Row Numbers option is unchecked in hidden mode). To show column titles, choose Format, then Column Titles. (The Row Numbers option is checked in show mode.) You can print row numbers on reports, but you cannot copy them to the clipboard or export them to a file.</td>
</tr>
<tr>
<td>Size Rows</td>
<td>To apply a custom height to a row, drag the bottom edge of the row to the new position. To apply the standard height to a row, choose Format, then Row, and then Standard Height. When you resize a row, all rows are repositioned in the report.</td>
</tr>
<tr>
<td>Eliminate Duplicate Rows</td>
<td>To eliminate duplicate rows, choose Query, then Query Options, and enable the Return Unique Rows option in the dialog box.</td>
</tr>
</tbody>
</table>

**Note:** If the data source for a query is a local results table even though the Return Unique Rows feature is enabled, the Unique Rows limit is not enforced.

### Formatting Results Items

The following table lists common formatting techniques for Results items.

Table 26  Results Item Formatting

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>To modify the way numbers, dates, and currency are displayed in the Results section, select the item and click Number on the shortcut menu or the Format menu; apply any desired changes on the Number Properties dialog box.</td>
</tr>
<tr>
<td>Alignment/Justify</td>
<td>To modify the way an item is justified within a row or column, use one of the following options:</td>
</tr>
<tr>
<td></td>
<td>● Select an item, choose Format, then Justify, and select the desired justification.</td>
</tr>
<tr>
<td></td>
<td>● Select an item, click Alignment on the shortcut menu, and enter the desired changes in the Alignment Properties dialog box.</td>
</tr>
<tr>
<td>Font</td>
<td>To modify the font of a report item, select the item, click Font on the shortcut menu or the Format menu, and enter the desired changes in the Font Properties dialog box.</td>
</tr>
<tr>
<td>Borders and Background</td>
<td>To modify the border of a column, select the column and choose Format, then Borders and Background.</td>
</tr>
<tr>
<td></td>
<td>To modify the background of a Results section, click anywhere within the Contents pane, choose Format, then Borders and Background and enter the desired information in the Border and Background Properties dialog box.</td>
</tr>
<tr>
<td>Grid lines</td>
<td>To modify gridline properties, choose Format, then Grid Lines, and select the grid line feature and enter the desired information in the Gridlines Properties dialog box.</td>
</tr>
</tbody>
</table>

### Enhancing Your Results Set

Before generating reports or exporting the data set, verify that the Results section contains all the information you need. You might want to redesign your query and process it again so that the data set is more manageable.
As you begin to refine your querying technique, you will increasingly use more sophisticated filters and computed items and will understand when to use aggregation in the Query section. Complex queries can be somewhat difficult for new users. When using data aggregation features, check the data set before creating reports.

You can return to the Query section to modify your query and refresh the data set if necessary, but you can also apply many of the same refinements locally in the Results section, including filters, sorts, and computations.

Enhancing your results set involves the tasks in these topics:

- “Filtering Results” on page 133
- “Sorting Results Data” on page 134
- “Adding Computed Items to Results” on page 134
- “Applying Data Functions to Results” on page 135
- “Adding Columns Automatically” on page 140

**Filtering Results**

Local filters applied in the Results section enable you to temporarily screen out portions of data for reporting purposes, without eliminating them from the data set.

You can apply filters to columns to locally filter the data set retrieved from the query. Since the other reporting sections reference the results set, the local filters are also disseminated to these sections.

Local filters are useful for managing your data set. If you decide you do not need all the information retrieved by your query, you can use a local filter to exclude data from the display. These filters are a good way to filter the data set to reflect temporary and hypothetical situations. You can always suspend or delete the filter to return data to the display and make it available for reporting.

▸ To apply a local filter to a column:

1. **Double-click a column.**
   
   The Filter dialog box is displayed.

2. **If desired, enter a descriptive name for the filter.**

3. **Select an arithmetic or logical operator from the drop-down list.** See also “Using Operators” on page 314.

4. **Define filter values by clicking one of the following options:**
   
   - **Show Values**—Shows database values associated with the item.
   
   - **Custom Values**—Supplies an empty field for inputting custom values. Click the check mark to add a value to the list of values. You can also display values from a custom list previously saved with the Interactive Reporting document file or loaded from a file.

5. **Select the values to include in the filter definition.**
Select values individually or click Select all and deselect the values you do not want to include. Only selected items are applied to the filter definition. To create a snapshot of the values, click Select All and click Transfer to move the variables to the Custom Values.

6 When the correct values for the filter expression are highlighted in the values list, click OK.

Tip: You can apply only one filter to a column.

➢ To remove a local filter, select the item and choose Remove on the shortcut menu.

Related Topics
“Server versus Local Filter Processing” on page 310

Sorting Results Data
Use the sort buttons to quickly sort a Results column or report item locally on your desktop. You can apply sequenced, nested sort conditions to Request items in the Results section. For information about sorts, see Chapter 13, “Applying Sorts.”

➢ To sort column data in the Results section, select the column or item to be sorted and click the ascending or descending button on the standard toolbar.

➢ To apply sort conditions using the Sort line, click Sort on the Results section titlebar and drag the items to be sorted to the Sort line.

To reverse the sort order for an item, select the item and click the ascending or descending sort icon on the Standard toolbar. Data is sorted in ascending (alphabetical or numerical) order by default.

Related Topics
“Complex Sorting in the Query, Results, and Table Sections” on page 402,
“Sorting Data” on page 401

Adding Computed Items to Results
You can rank and provide statistics for the values represented as totals or subtotals in your Results section. The Add Computed Item command enables you to build equations to compute totals, or to apply functions to values. Computations are performed on the desktop by Interactive Reporting and involve only the data in your Results set. Therefore, you can only create new computed items – you cannot modify original data items that were retrieved from the database.

Computed items are like normal data items, and can be included in reports or reused to compute other data.

For example, you can modify the Amount Sold item by building an equation around it, multiplying by the Unit Price item and renaming the resulting item 'Revenue'. You can also
apply a scalar function such as Cume to Amount Sold and return each individual value as a cumulative running total, or simply multiply Amount Sold by the local tax rate to find the tax owed on each sale.

In the Results section, reference items are filtered to the items that is displayed on the Request line of the original query. Also, the scalar functions used to compute items are provided at the desktop level rather than the RDBMS.

For more information about computed items, see Chapter 12, “Computed Items.”

Applying Data Functions to Results

In the Results section, you can only use a data function for totals and subtotals. The other values cannot be recalculated without redoing the query. Data functions return to the underlying values and recalculate the value according to the type of function specified.

You can apply a break (subtotal), grand, or custom total to any column. The following table lists the data functions that you can use with break totals and grand totals.

<table>
<thead>
<tr>
<th>Table 27</th>
<th>Break Total and Grand Total Data Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>Sum</td>
<td>Returns sum of underlying values.</td>
</tr>
<tr>
<td>Average</td>
<td>Returns average of underlying values.</td>
</tr>
<tr>
<td>Minimum</td>
<td>Returns lowest of underlying values.</td>
</tr>
<tr>
<td>Maximum</td>
<td>Returns highest of underlying values.</td>
</tr>
<tr>
<td>Count</td>
<td>Returns number of underlying values.</td>
</tr>
<tr>
<td>Other</td>
<td>Allows you to create a custom function using JavaScript.</td>
</tr>
</tbody>
</table>

Inserting Break Totals

Use the Insert Break Total dialog box to apply a break total to a column. Break totals are typically used in reference to subtotals. Subtotals are generated by a break column. The break column contains the data by which you group (break) your total. A break column contains characters (as opposed to numbers). When you apply a break total, the total automatically sorts data by the break column and concatenates the unique values.

To apply a break total (subtotal):

1. Select a column and choose Results, then Break Total.
   The Insert Break Total dialog box is displayed.
2. Select a break column from the At Every Break drop-down list.
3. Select a data function from the Break Total Function drop-down list.
4 Select one or more columns on which to display the break total and click OK.

To remove a break total:

1 Highlight a break total row by clicking in the left margin.
   The break total row is highlighted

2 On the shortcut menu, click Remove Row.
   All break totals associated with the rows are removed from the column.

### Table 28 Insert Break Total

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>At every break in</td>
<td>The column on which to break.</td>
</tr>
<tr>
<td>Break total function</td>
<td>The data function to apply. Available data functions include:</td>
</tr>
<tr>
<td></td>
<td>● Sum—Returns the sum of underlying values.</td>
</tr>
<tr>
<td></td>
<td>● Average—Returns the average of underlying values.</td>
</tr>
<tr>
<td></td>
<td>● Minimum—Returns the lowest of underlying values.</td>
</tr>
<tr>
<td></td>
<td>● Maximum—Returns the highest of underlying values.</td>
</tr>
<tr>
<td></td>
<td>● Count—Returns the number of underlying values.</td>
</tr>
<tr>
<td></td>
<td>● Other—Allows you to create a custom function using JavaScript. See Working with Custom Functions for more information.</td>
</tr>
<tr>
<td>Add break total to</td>
<td>The column or columns on which to apply the break total.</td>
</tr>
</tbody>
</table>

### Inserting a Grand Totals

Use the Insert Grand Total dialog box to apply a grand total to a column. A grand total on a numeric column applies a default "sum" function. However, each column can have a number of grand totals, each with a different aggregate function (such as minimum, maximum, average, and so on) applied to it.

To apply a grand total to a column using a data function:

1 Select a column and choose Results, then Grand Total.
   The Insert Grand Total dialog box is displayed.

2 Select a data function from the Grand Total Function drop-down list.

3 Select one or more columns to be totalled from the Add Grand Total To list and click OK.
   The total and any subtotals in the column are computed to reflect the new data function.
Table 29  Insert Grand Total

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grand total function</td>
<td>The data function to apply. Available data functions include:</td>
</tr>
<tr>
<td></td>
<td>- Sum—Returns the sum of underlying values.</td>
</tr>
<tr>
<td></td>
<td>- Average—Returns the average of underlying values.</td>
</tr>
<tr>
<td></td>
<td>- Minimum—Returns the lowest of underlying values.</td>
</tr>
<tr>
<td></td>
<td>- Maximum—Returns the highest of underlying values.</td>
</tr>
<tr>
<td></td>
<td>- Count—Returns the number of underlying values.</td>
</tr>
<tr>
<td></td>
<td>- Other—Allows you to create a custom function using JavaScript. See Working with Custom Functions for more information.</td>
</tr>
<tr>
<td>Add grand total to</td>
<td>The column or columns on which to apply the grand total.</td>
</tr>
</tbody>
</table>

Inserting Column Totals

Interactive Reporting adds a row labeled Total to the bottom of the table, and display the total as the last entry in the selected column.

➢ To calculate a column total, select the column to be totaled and click the [on the Standard toolbar.

Modifying Total Function

Use the Modify Total Function dialog box to change the standard mathematical functions for totals in the Results Section.

For example, you could total yearly amounts three to four times and apply different functions to each of the totals. In this way, your Results section could contrast values such as sums, averages, or maximum values with other standard or customized mathematical functions that you specify.

➢ To modify a total function, select a function from the Grand Total Function drop-down list box and click OK.

You can select any of the following functions:

Table 30  Grand Total Functions (Continued)

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sum</td>
<td>Returns the total of a column of numbers.</td>
</tr>
<tr>
<td>Average</td>
<td>Returns the average (arithmetic mean) of values in a number column. (The Avg includes NULL values when calculating the arithmetic mean.)</td>
</tr>
<tr>
<td>Maximum</td>
<td>Returns the largest of items in a column.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>Minimum</td>
<td>Returns the smallest of items in a column.</td>
</tr>
<tr>
<td>Count</td>
<td>Counts and returns the number of rows in a column.</td>
</tr>
<tr>
<td>Other</td>
<td>Applies a custom function.</td>
</tr>
</tbody>
</table>

## Working with Custom Functions

Use the Custom Function dialog box to insert standard and custom mathematical (numeric) functions when you add a total in the Results Section.

Custom functions in the Results section involve only the data in your Results set. Apply the custom functions that you use to aggregated values. Aggregation means that data is grouped and summarized by subjecting the data to some mathematical operation, such as:

- Sum
- Average
- Maximum
- Minimum
- Count

The result is a summary of the data at a higher level.

When you insert a mathematical expression at the total level, the total must be aggregated. For example, write: `Sum(Units) * 5` instead of: `Units * 5`.

You can also use the Custom Function dialog box to calculate multiple column totals on the same rows since by default Interactive Reporting staggers columns totals.

1. **To apply a custom function:**
   1. **Access the Custom dialog box by doing one of the following:**
      - Select a numeric column and choose Results, then Break Total. In the Insert Break Total dialog box, select Other from the Break Total Function drop-down list.
      - Select a numeric column and choose Results, then Grand Total. In the Insert Grand Total dialog box, select Other from the Grand Total Function drop-down list.
      - Select a numeric column and click the Grand Total button on the Standard Toolbar. Interactive Reporting displays the total in a new row at the bottom of the selected column. Double-click either the row cell for the total, or double-click another column of numeric values.
      - If you double-clicked a total, the Modify Total Function dialog box is displayed. In the Modify Total Function dialog box, select Other from the Grand Total Function drop-down list.

2. In the Custom Function dialog box, enter the desired expression and click OK.
The expression must consist of a mathematical operation (aggregate value), the name of the numeric values column, and if applicable, any JavaScript operators.

**Note:** The Custom Function dialog box accepts any expressions built with JavaScript.

## Grouped Columns

Use the Grouped Column dialog box to add a grouped column to your results. Grouped columns, like computed items, create new data in your results set by grouping data from a column. You can use grouped columns to consolidate nonnumeric data values into more general group values and map the group values to a new column in the data set.

Grouped columns are new items added to the Results section and are available for use in report sections.

For example, your company sales database may contain the items: State, Sales Region, and Country, which allow you to aggregate data on different levels in reports. However, suppose you are looking to track sales by subregion, or you want to see data for one state versus an average for all other states combined. You can do this by grouping states together to create a Subregion item or other custom dimension.

To add a grouping column:

1. Select a column as a base for your grouped column.
2. Select **Results**, then **Add Grouping Column**, or choose **Add Grouping Column** on the shortcut menu.

   The Grouped Column dialog box is displayed. Use the column values to build the grouping categories for the new item.

3. Type a name for the new column in the **Column Name** field.
4. Create custom group values and link them to values in the base column.
   - Click **New Groups** to create groups and add them to the Groups list.
   - Select a group in the Grps list; select items from the Available Values list and use the arrows to add them to the Items In Group list for the selected group.
   - Remove selected values from a group by using the arrow to move them back to the Available Items list.
   - Double-click a group name to modify it.
   - Click the **Options** button and specify options for Ungrouped values.

   Specify options for ungrouped values:
   - **Null**—Leaves the values ungrouped and unaggregated.
   - **Default**—Allows you to specify a default name to assign to all ungrouped values.
   - **Individual Group**—Assigns each ungrouped values the name originally assigned to it.

5. When the grouping definitions are complete, click **OK**.
The new grouping column is added to the Request line and to the Content pane.

To modify a grouped column, select the grouped column and choose **Results**, then **Modify Column**.

<table>
<thead>
<tr>
<th>Table 31</th>
<th>Grouped Column Options</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Option</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>Column Name</td>
<td>Names the new grouping column in the Results window.</td>
</tr>
<tr>
<td>New Groups</td>
<td>Creates a custom group to be displayed as a value in the new grouping column.</td>
</tr>
<tr>
<td>Groups</td>
<td>Select a custom group to define by adding or removing items.</td>
</tr>
<tr>
<td>Items in Group</td>
<td>Removes an item from a selected custom group.</td>
</tr>
<tr>
<td>Available Values</td>
<td>Adds items to a selected custom group.</td>
</tr>
<tr>
<td>Options</td>
<td>Indicates how to represent unassigned values within the grouping column. Options include:</td>
</tr>
<tr>
<td></td>
<td>• Null—Leaves the values ungrouped and unaggregated.</td>
</tr>
<tr>
<td></td>
<td>• Default—Allows you to specify a default name to assign to all ungrouped values.</td>
</tr>
<tr>
<td></td>
<td>• Individual Group—Assigns each ungrouped value the name originally assigned to it.</td>
</tr>
</tbody>
</table>

**Adding Columns Automatically**

By default, Interactive Reporting retrieves data to your desktop and displays it in the Results section as columns. You can manually add request items by having Interactive Reporting return an empty Results set. This allows you to add columns as you need them.

This feature allows you to display selected rows. You can still sort or create filters using columns not displayed in the Results section.

To toggle AutoAdd columns, select **Results**, then **AutoAdd Columns**.

If the AutoAdd Columns feature is selected, all requested items are displayed in columns.

If the AutoAdd Columns feature is not selected, no columns are returned to the Results section and you have to manually add requested items.

**Breaking Out Dates**

Use date breakout columns to separate date-typed columns into Year, Quarter, and Month items. The new items are automatically derived using date functions available to computed items.

For example, when you add date groups for an item Order Date, the item is broken into constituent date items. A new Year item is created as an integer, Qtr as a string, and Month as a new date.
To break out date items:

1. In the Content pane, select a date-type column.
2. Select Results, then Add Date Group.

**Note:** This feature automatically sets the display format of the new Month item to mmm so that the data sorts correctly. Quarters are based on the calendar year beginning 1/1.

---

**Working with the Results Table**

Interactive Reporting offers a number of options for working with table components (that is, columns and rows) in the Results section. These commands are found on the Format and Results menus. Many of these commands also have corresponding toolbar icons and are available on the shortcut menu.

- “Selecting Columns and Rows” on page 141
- “Deleting Columns” on page 141

---

**Selecting Columns and Rows**

- To select a column, click anywhere inside the column.
- To select a row, click the row header (row number).

**Deleting Columns**

- To delete a selected column from the Results table (and data layout), select Results, then Remove.

If an item is removed from the Content pane, it is completely removed from the data layout and the data set.

**Caution!** Remove items with caution as computed items and other report sections may draw data values from the deleted item.

---

**Formatting Commands**

You can use the commands available on the Format menu to change the appearance of fonts, backgrounds, borders, color, row heights, and column widths.
Saving Query Results with the Document

The Save Query Results with Document dialog box displays all of the query sections and computed columns contained in your document. Use this dialog box to specify whether to save the results set of your query with your document, and whether to save computed column expressions by section as a snapshot.

Saving results with your query allows you to analyze and generate reports without being connected to the database. Results are saved for an individual query or for multiple queries for which results have been generated. You also can specify whether to save any computed columns in the results set as a snapshot with the document.

Saving your results set makes sense if you cannot connect to a database, for example, when traveling or working remotely, or if you are scheduling or forwarding documents for someone else’s use.

You cannot choose to save computed columns as snapshots unless you first choose to save the corresponding query results.

Note: If you intend to work with a document that includes a Report Designer section, you must save results with the document. If you do not save results with the document, the Report Designer section is not available.

To save results with your document:

1. Select File, then Save Query Results With Document.
   The Save Query Results With Document dialog box appears and displays all of the query sections contained in your document.

2. In the Query Results pane, select the check box next to the query results to save OK.

3. In the Computed Columns pane, select the section with the computed column to save as a snapshot.
   The query results and snapshots for computed columns are automatically saved the next time you save the document.
   Computed values saved as snapshots are not recalculated when the document is opened. Not even dynamic expressions (for example, values that reference the systime function such as date or time) are recalculated. They are recalculated only when the query is reprocessed. Documents that are saved with computed columns as snapshots tend to be larger in size than documents that do not contain snapshots, but they take less time to open.
   To save the computed columns with the document, do not check the Computed Columns field.

4. Click OK.

To automatically recalculate the values of computed columns when a document is opened, do not select the corresponding results section in the Computed Columns list. The document may take longer to open, especially if the results set contains a large number of computed columns or uses complex formulas in the definitions, since all computed values are recalculated in the Results section and in any other section that references the Results section. On the other hand,
documents that do not contain snapshots tend to be smaller in size than documents that contain snapshots of computed columns.

The following table lists the selection options and effects for saving query results and snapshots of computed columns with documents.

Table 32

<table>
<thead>
<tr>
<th>Save Query</th>
<th>Save Computed Columns (as Snapshot)</th>
<th>What Happens</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selected</td>
<td>Selected</td>
<td>Results are saved with the document and computed columns are saved as a snapshot. Computed columns are not recalculated when the document is opened. Values are recalculated only when the query is reprocessed.</td>
</tr>
<tr>
<td>Selected</td>
<td>Not Selected</td>
<td>Results are saved with the document but computed columns are not saved as a snapshot. Computed columns are recalculated when the document is opened.</td>
</tr>
<tr>
<td>Not Selected</td>
<td>Selected</td>
<td>either Results nor computed columns are saved with the document.</td>
</tr>
<tr>
<td>Not Selected</td>
<td>Not Selected</td>
<td>Neither Results nor computed columns are saved with the document.</td>
</tr>
</tbody>
</table>

**Note:** You cannot save computed columns as snapshots unless you first save the corresponding query results.

**Exporting Result Sets**

After processing a query, you can export the data contents of the Results section for use in other applications. There are several ways to export, the most common being into file formats such as Excel or Lotus.

Interactive Reporting also exports to HTML format, making it easy to distribute data to many corporate intranets or Web sites. Exported Results section data is raw and unaggregated. If you export from a report section, the data is drawn from the desktop data cube and is preaggregated. Scripts created by the Interactive Reporting JavaScript engine can be saved to a text file.

Export options are discussed in detail in “Exporting Data” on page 45. This section also covers export properties such as whether to use page breaks in HTML files, or whether to include double quotation marks in tab-delimited text files.

**Extended Access for Interactive Reporting**

Extended Access for Interactive Reporting enables users to jointly analyze multidimensional and relational sources in one document. It retrieves the flattened OLAP results from a Oracle Hyperion SQR Production Reporting, and Oracle Hyperion Financial Reporting query in the Reporting and Analysis repository and imports the data into the Interactive Reporting document as a results section. Extended Access for Interactive Reporting also enables:
An end user can perform quick ad-hoc analysis by drilling down, swapping and pivoting on
the data to see patterns or exceptions.

The results section can be used as a new data source and can be joined with other data sources
either from the database or other tables.

The end user can perform offline analysis for multidimensional data.

The Interactive Reporting document can be refreshed with multidimensional data.

The Query/Report (Oracle Hyperion Web Analysis) can be refreshed.

The latest job output/snapshot (Production Reporting and Financial Reporting) can be
retrieved.

The following steps assume that a Web Analysis query (Web Analysis document), Production
Reporting query (Production Reporting job output) or a Financial Reporting query (Batch report)
is resident in EPM Workspace. These queries should be formatted so that their data is
compatible in a relational query (flattened data), see the EPM WorkspaceUser online help.

Note: Extended Access for Interactive Reporting does not support null values. If a null value is
needed for the document, the user must convert the null value to a special value in Oracle
Hyperion Financial Reporting, Oracle Hyperion Web Analysis and Oracle Hyperion SQR
Production Reporting. Then the user must use an Interactive Reporting function to
convert it back to a null value.

Extended Access for Interactive Reporting can only be done in the Interactive Reporting Web
Client application. If the Interactive Reporting Web Client application has not been installed,
select Tools, then Install and then Interactive Reporting Web Client. When the Interactive
Reporting Web Client Setup dialog is displayed, complete the installation.

To import a multidimensional data source into an Interactive Reporting document:

1. In EPM Workspace, navigate to the Interactive Reporting document to which you want to extend access
   for a multidimensional report.
2. Select the Interactive Reporting document and select Open as, then Interactive Reporting Web Client
   from the shortcut menu.
   The selected document is displayed in the Interactive Reporting Web Client.
3. Navigate to the section to which you want to add the multidimensional report as a results set.
4. Select File, then Import Data, and then From Repository.
   The Oracle Hyperion Reporting and Analysis Select dialog is displayed.
5. Navigate to the folder which contains the repository document.
6. Double-click the object to select it.
   Make sure that the object name displays in the Name field.
7. Click Select.
   The object is displayed in Sections list as a results set.
# Results Menu Command Reference

The following table provides a quick reference to the commands available on the Results menu and lists any related shortcuts.

## Table 33 Results Menu Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Keyboard Shortcut</th>
<th>Shortcut Menu</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter</td>
<td>Opens the Filter dialog box.</td>
<td>Ctrl+L</td>
<td>*</td>
</tr>
<tr>
<td>Sort Ascending</td>
<td>Sorts the selected column values in ascending order (alphabetical or numeric).</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sort Descending</td>
<td>Sorts the selected column values in descending order (alphabetical or numeric).</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add Computed Item</td>
<td>Opens the Insert Computed Item dialog box.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add Grouping Column</td>
<td>Opens the Grouped Column dialog box. Use to merge dimension labels into new groupings and aggregate the associated data.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add Date Groups</td>
<td>Separates date-type items into year, quarter, and month items.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modify Column</td>
<td>Use to modify a computed column or a group column.</td>
<td>Ctrl+M</td>
<td></td>
</tr>
<tr>
<td>Remove</td>
<td>Removes the selected column (or data layout item).</td>
<td>Del</td>
<td>*</td>
</tr>
<tr>
<td>Break Total</td>
<td>Opens the Insert Break Total dialog box.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Grand Total</td>
<td>Opens the Insert Grand Total dialog box.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hide Column</td>
<td>Hides the selected column from view.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unhide Column</td>
<td>Opens the Unhide Column dialog box.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AutoAdd Columns</td>
<td>Automatically adds columns in the Content pane for all requested items.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>If not selected, no columns are displayed in the Content pane. Turn this option off to manually add columns for requested items.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Aggregate Shared Members</td>
<td>If the “Aggregate Shared Members” option is enabled, the total is included for the initial shared member and for any parent members above it.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table Section

A table is a columnar arrangement of data. Tables are used as building blocks in other reporting sections. A Table section functions is used in much the same way as the Results section. All of the commands that are available on the Results menu are also available on the Table menu. However, you cannot apply complex filters or aggregate data in the Table section.

The data in a Table section is derived from the section that is active when you insert a new table. When you insert a Table section from the Results section, the new Table section is attached to the Results section. This means any changes to the results set are propagated immediately to the Table section.

You can also create additional tables, Pivot tables, Charts, and other reports form a data set of a Table section just as you would from the Results section. Filters, computed columns, grouping columns, and other actions that modify the data set of an active section set carry over to all tables and reporting sections built from that section.

Creating Tables

Use the Table data layout to construct, plot, and manage data.

1. To create a table based on the Results section data:
   - From the Results section, select Insert, then New Table.
     
     If the Table data layout is not already displayed, click data layout on the Section title bar.

2. Drag Results items from the Catalog pane to the Table data layout.
Interactive Reporting populates automatically the table columns.

**Tables as a Data Staging Area**

A *computed item* in the Results section becomes a mere data element when added to a Table section. Thus, the Table section can become an intermediate calculation staging area. This ability comes in handy in applications where you wish to place filters on computed items.

For example, suppose you want a report of the top ten producers in your organization. It is easy to add a computed item to your Results section that is based on the Rank function. (This function lets you find out each producer’s rank relative to the others.) However, when you try to filter the results based on that ranking, Interactive Reporting informs you that filters cannot be placed on aggregate items.

To get the results, insert a new Table section based on your Results section. Next, add all the relevant Results items, including the computed Rank field. Once Rank is a column in the table, it is no longer a computed item. It is a regular number on which you can now place a filter. Since the Table section is based on the Results section, your Top 10 report is updated automatically each time you run the query.

**Manipulating Table Data**

The Table menu provides a number of commands that enable you to manipulate the data in the Table section. Review the following sections for information on:

- Filtering Table Data
- Sorting a Table
- Adding Computed Items
- Adding Grouping Columns
- Adding Date Groups
- Applying Data Functions to Tables

**Filtering Table Data**

Filtering data in a table filters the data displayed in the table columns. You can apply *filters* in the Table section in addition to any filters set in the originating section. Filters set in the Table section are automatically propagated to any other reports that inherit their data set from the table. You can apply only one filter per column.

To return data to the display and make it available for reporting, delete or suspend the filter.

➢ To filter data in a table:

1. **Double-click a column, or click the column heading and select Table, then Filter.**

   The Filter dialog box is displayed.
2  Select an arithmetic or logical operator from the drop-down list box.

3  Define the potential filter values by selecting one of the following options:
   - **Show Values**—Shows column values associated with the item.
   - **Custom Values**—Supplies an empty field for inputting custom values. Select the check
     mark to add a value to the list.

4  In the Values list, select the values to include in the filter definition.
   Individually select values or click **Select All** and deselect the values that you do not want to
   include.

5  **When the values are highlighted in the values pane, click **OK**.**
   The filter is applied to the column and the column name is added to the Filter line.

   ➤  To remove a filter in a table, select the filter item and choose **Table Remove**, then **Remove**.

   ➤  To remove a filter in a table, select the filter and do one of the following:
      - Click the **Delete** key.
      - Select **Table**, then **Remove**.
      - Click **Remove** on the shortcut menu.

   ➤  To remove all filters in a table, click **Filter** on the Filter line and select **Table**, then **Remove**.

   ➤  To remove all filters in a table, click **Filter** on the Filter line and do one of the following:
      - Click the **Delete** key.
      - Select **Table**, then **Remove**.
      - Click **Remove** on the shortcut menu.

### Sorting a Table

You can sort the rows in a table by one or more columns in ascending or descending order. You
   can also apply sequenced, nested sorts to columns in the Table section.

   ➤  To sort a column, select the column and choose **Table**, then **Sort Ascending** or **Sort
       Descending**.

   ➤  To apply sort conditions:

   1  **Click **Sort** on the Section Title bar to display the Sort line.**

   2  **Drag Results items from the Catalog pane to the Sort line.**
      You can add items to the Sort line that are not in the data layout.

   3  **Establish a final sort sequence by reordering sort items.**
Items are sorted left to right on the sort item. To reorder the sequence, drag each item to its new position.

4 Double-click specific sort items to toggle ascending and descending sort orders.

   Ascending is the default sort order.

5 Click Sort Now on the Sort line.

### Adding Computed Items

You can rank and provide statistics for the values represented in the totals or subtotals. The Add Computed Item feature enables you to build equations to compute totals, or to apply functions to values. Computed items are like normal data items and can be included in reports or reused to compute other data. For example, you can modify an Amount Sold item by building an equation around it, multiplying by a Unit Price item, and renaming the resulting item Revenue. You can apply a scalar function such as \( \text{Cume} \) to Amount Sold and return each individual value as a cumulative running total, or simply multiply Amount Sold by the local tax rate to find the tax owed on each sale.

The Computed Item dialog box is used to build a computed item expression. The computed item expression is a value, variable, logic statement, or equation that instructs Interactive Reporting how to perform a computation.

To create a computed item:

1. **Select Table, then Add, and then Computed Item**
   
   The Computed Item dialog box is displayed.

2. **In the Name field, type a name that describes the computation.**
   
   The default name is Computed, which is numbered sequentially if there is more than one. If you assign a name to a computed item that is identical to a scalar function name, Interactive Reporting numbers the name starting with the number 2.

3. **Define the new data item by building an expression in the Definition text box.**
   
   Use the operator buttons to insert arithmetic and logical operators at the insertion point.
   
   - Click **Reference** to display the Reference dialog box, and select Request items to place in the equation.
   
   - Click **Functions** to apply scalar functions using the Functions dialog box.

   You can also type any portion of the equation or the entire equation directly into the Definition text box using JavaScript. The names are case sensitive, and you must replace spaces in item names with underscores (‘_’).

4. **If necessary, click Options to set a new data type for the item.**

5. **When the equation is complete, click OK.**

   The computed item is added to the data layout and it is displayed as a table column.
Tip: To retrieve the top five records in a table, first add a computed item on the results section to rank the records. The computed item would be Rank (Sorted_Column), where the Sorted_Column would be sorted in descending order. Next, create a new table section and add the columns of data. Add a filter on the computed item column in the Table section and set it to < 6.

Note: Add a computed item on the results section to rank the records. Such a computed item should be Rank (Sorted_Column), where the Sorted_Column would be sorted in descending order. Next, create a new table section and add the columns of data. Add a limit on the computed item column in the Table section and set it to < 6.

Adding Grouping Columns

Grouping columns is a way of creating new data in your results set by grouping data from a column. You can use grouping columns to consolidate non-numeric data values into more general group values and map the group values to a new column in the data set.

Grouping columns are new items added to the Table section and are available for use in report sections.

For example, your company sales database may contain the items: State, Sales Region, and Country, which enable you to aggregate data on different levels in reports. However, suppose you are looking to track sales by subregion, or want to see data for one state versus an average for all other states combined. You can do this by grouping states together to create a subregion item or other custom dimension.

To add a grouping column:

1. Select a column from which to base grouping column.
2. Select Table, then Add Grouping Column.
   - The Grouped Column dialog box is displayed.
   - Use the column values to build the grouping categories for the new item.
3. Type a name for the new column in the Column Name field.
4. Create custom group values and link them to values in the base column.
   - Click New Groups to create groups and add them to the Groups list.
   - Select a group items from the Available Values list. Use the arrows to add them to the Items In Group list for the selected group.
   - Remove selected values from a group by using the arrow to move them back to the Available Items list.
   - Double-click a group name to modify it.
   - Specify options for ungrouped values as follows:
     - Column Name—Names the new grouping column in the table.
- **New Groups**—Creates a custom group to be displayed as a value in the new grouping column.

- **Options**—Indicates how to represent unassigned values within the grouping column, that is, as null values, as members of a default group (named in the adjacent edit field), or as their own individual groups.

- **Groups**—Selects a custom group to define by adding or removing items.

- **Items In Group**—Removes an item from a selected custom group.

- **Available Values**—Adds items to a selected custom group.

- Select one of the following options to define the preferences for ungrouped columns:
  - **Null**—Leaves the values ungrouped and disaggregated.
  - **Default**—Enables you to specify a default name to assign to all ungrouped values.
  - **Individual Group**—Assigns each ungrouped values the name originally assigned to it.

5 When the grouping definitions are complete, click **OK**.

The new grouping column is added to the data layout and to the table.

### Modifying Grouping Columns

You can modify a grouping column to change the group structure.

- To modify a grouping column, select the grouping column and choose **Table**, then **Modify Column**.

### Adding Date Groups

Use date breakout columns to separate date-typed columns into Year, Quarter, and Month items. The new items are automatically derived using date functions available to computed items.

For example, when you add a *date group* for an item *Order Date*, the item is broken into constituent date items. A new *Year* item is created as an integer, *Qtr* as a string, and *Month* as a new date.

- To break out date items:
  1. **Select a date-type column** in the Content pane.
  2. **Select Table**, then **Add Date Group**.

**Note:** This feature automatically sets the display format of the new Month item to *mmm* so that the data sorts correctly. Quarters are based on the calendar year beginning 1/1.
Applying Data Functions to Tables

In the Tables section, a Dashboard can be used only for totals and subtotals. Data functions return to the underlying values and recalculate the value according to the type of function specified.

You can apply a break (subtotal), grand, or custom total to any column. A grand total on a numeric column applies a default sum function. However, each column can have a number of grand totals, each with a different aggregate function applied to it. The table below lists the data functions that you can use with break totals and grand totals.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sum</td>
<td>Returns sum of underlying values.</td>
</tr>
<tr>
<td>Average</td>
<td>Returns average of underlying values.</td>
</tr>
<tr>
<td>Minimum</td>
<td>Returns lowest of underlying values.</td>
</tr>
<tr>
<td>Maximum</td>
<td>Returns highest of underlying values.</td>
</tr>
<tr>
<td>Count</td>
<td>Returns number of underlying values.</td>
</tr>
<tr>
<td>Other</td>
<td>Enables you to create a custom function using JavaScript.</td>
</tr>
</tbody>
</table>

Column Totals

Interactive Reporting adds a row labelled Total to the bottom of the table and display the total as the last entry in the selected column.

➢ To calculate a column total, select the column to be totaled and click on the Standard toolbar.

Grand Totals

➢ To apply a grand total to a column using a data function:

1 Select a column and choose Table, then Grand Total.
   The Insert Grand Total dialog box is displayed.

2 Select a data function from the Grand Total Function drop-down list box.

3 Select one or more columns to be totaled from the Add Grand Total To list and click OK.
   The total and any subtotals in the column are computed to reflect the new data function.
Break Totals

To apply a break total (subtotal):

1. Select a column and select Table, then Break Total.
   The Insert Break Total dialog box is displayed.
2. Select a break column from the At Every Break in drop-down list box.
3. Select a data function.
4. Select one or more columns on which to display the break total and click OK.

Working with Table Components

Interactive Reporting offers a number of options for working with table components (that is columns and rows) in the Table section. These commands are found on the Format and Results menus. Many of these commands also have corresponding toolbar icons and shortcut menu items.

Selecting Columns and Rows

- To select a column, click anywhere inside the column.
- To select a row, click the row header (row number).

Deleting Columns

- To delete a selected column from the Results table (and data layout), select Results, then Remove.

If an item is removed from the Content pane, it is completely removed from the data layout and the data set.

Caution! Remove items with caution as computed items and other report sections may draw data values from the deleted item.

Table Menu Command Reference

The table below provides a quick reference to the commands available on the Table menu and lists any related shortcuts.
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Keyboard Shortcut</th>
<th>Shortcut Menu</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter</td>
<td>Opens the Filter dialog box.</td>
<td>Ctrl+L</td>
<td>*</td>
</tr>
<tr>
<td>Sort Ascending</td>
<td>Sorts the selected column values in ascending order (alphabetical or numeric).</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Sort Descending</td>
<td>Sorts the selected column values in descending order (alphabetical or numeric).</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Add Computed Item</td>
<td>Opens the Insert Computed Item dialog box.</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Add Grouping Column</td>
<td>Opens the Grouped Column dialog box. Use to merge dimension labels into new groupings and aggregate the associated data.</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Add Date Groups</td>
<td>Separates date-type items into year, quarter, and month items.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modify Column</td>
<td>Use to modify a computed column or a group column.</td>
<td>Ctrl+M</td>
<td></td>
</tr>
<tr>
<td>Remove</td>
<td>Removes the selected column (or data layout item).</td>
<td>Del</td>
<td>*</td>
</tr>
<tr>
<td>Break Total</td>
<td>Opens the Insert Break Total dialog box.</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Grand Total</td>
<td>Opens the Insert Grand Total dialog box.</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Hide Column</td>
<td>Hides the selected column from view.</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Unhide Column</td>
<td>Opens the Unhide Column dialog box.</td>
<td></td>
<td>*</td>
</tr>
</tbody>
</table>
7

Querying Multidimensional Databases

In This Chapter

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>OLAPQuery Section</td>
<td>157</td>
</tr>
<tr>
<td>OLAPQuery Options</td>
<td>158</td>
</tr>
<tr>
<td>Building OLAP Queries</td>
<td>161</td>
</tr>
<tr>
<td>Refining OLAPQuery Data</td>
<td>164</td>
</tr>
<tr>
<td>Processing OLAP Queries</td>
<td>169</td>
</tr>
<tr>
<td>Applying Filters</td>
<td>170</td>
</tr>
<tr>
<td>Changing Data Views</td>
<td>175</td>
</tr>
<tr>
<td>Formatting OLAPQuery Items</td>
<td>178</td>
</tr>
<tr>
<td>Drilling Through from a Multi-Dimensional Database to a Relational Database</td>
<td>178</td>
</tr>
<tr>
<td>OLAP Menu Command Reference</td>
<td>180</td>
</tr>
</tbody>
</table>

OLAPQuery Section

Use the OLAPQuery section to query multidimensional or OLAP databases. When you connect to a multidimensional database, the document opens an OLAPQuery section.

The user interface is similar to the Pivot section, except the OLAPQuery section displays the multidimensional database components as a hierarchical tree in the Catalog pane. These components include:

- **Dimensions**—categories of information, such as Location, Products, Stores, and Time. Dimensions are for informational use only and cannot be used as query items. (If a dimension contains multiple hierarchies, they will be represented under the dimension.)
  - **Levels**—Groups of similar types of members. For example, using the members listed in a Location dimension, Japan, USA, and France belong to the Country level. San Francisco, Paris, Tokyo, and Rome belong to the City level. 35 Main Street belongs to the Address level.
  - **Hybrid (Analysis)**—Further dimensional data is present by way of Hybrid Analysis (the lowest levels of an Essbase or DB2 cube that can reside in a relational database, but are not reflected in the cube structure (metadata) that is resident on the Essbase Server itself). If a dimension has Hybrid Analysis data, the Essbase Server, in conjunction with the Essbase Integration Server automatically retrieves the data from the appropriate relational table source and passes it to Interactive Reporting. If a dimension has available hybrid analysis, it is indicated to the right of the level as shown below.
Members—Content values for a dimension. In the Location dimension, they could be:
San Francisco, Japan, Paris, 35 Main Street, Tokyo, USA, France, Rome, and so on. These
all are values for location.

Members are shown only under the Values (I) icon and can be dragged only into the
Top Labels or Side Labels panes in data layout.

Member Property—A descriptive piece of information about a member that can be
retrieved and displayed in the OLAPQuery. This information is metadata and does not
in itself constitute a distinct member in the dimensional hierarchy. For example, let’s
assume the following hierarchy:

Product
  Category
  Product Name

The Product Name level might have the following properties defined:

Product Description
Product SKU
Color
Size
Weight

You can drag a property into the data layout with its corresponding level, but you cannot
drag it into the Slicer or the Measures pane, or apply a filter to it.

Measures—Numeric values in a database cube that are available for analysis. The measures
could be margin, cost of goods sold, unit sales, budget amount, and so on.

Individual measures are shown under the Measures icon and can be dragged only to the
Measures pane.

Note: Essbase only. Interactive Reporting supports Essbase Attribute Dimensions. An
Attribute Dimension displays in the OLAPQuery section with the word “attribute”
to right of the Attribute Dimension’s name. In addition, for each Attribute
Dimension, Interactive Reporting creates an Attribute Calculation Dimension. An
Attribute Calculation Dimension is displayed at the bottom of the Catalog pane. You
can position the members contained within the Attribute Calculation Dimension in
the Top or Side Labels of the data layout.

OLAPQuery Options

Interactive Reporting enables you to set options that control various properties in the
OLAPQuery section.

- “General OLAPQuery Options” on page 159
- “Database-specific OLAPQuery Options” on page 160
To define OLAPQuery options, select OLAP, then OLAP Query Options.

**General OLAPQuery Options**

General OLAPQuery options can be adjusted regardless of the database. The General tab in the OLAP Query options dialog box is displayed as follows:

### Table 36  Results Options

<table>
<thead>
<tr>
<th>Results</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Download Database Totals</td>
<td>The Download Database totals feature creates only one column per dimension, and includes all members, and the corresponding data, in the results set. In addition, a column with each dimension's parent members is included. By default this option is disabled. If this option is enabled through the Tools, then Programs, and then OLAP menu, the Query option is enabled also in a newly created CubeQuery section. <strong>Note:</strong> If you open an Interactive Reporting version 11.1.1 document that contains data in Results downloaded from a CubeQuery with version 9.3.1, the existing Results data is not visible. Furthermore, using the “download to results” feature with an Interactive Reporting document created with Interactive Reporting version 11.1.1, the result set is not be the same as it was with version 11.1.1.</td>
</tr>
<tr>
<td>Auto Generate Results When Processing OLAP Query</td>
<td>Creates a Results section for any future OLAPQuery section when that OLAPQuery section is first processed.</td>
</tr>
</tbody>
</table>

### Table 37  OLAP Query Options

<table>
<thead>
<tr>
<th>OLAP Query</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use OLAPQuery section</td>
<td>Enable the display of a new OLAPQuery (Essbase) section in the pre Release 9.3.1 section format. This option also enables pre Release 9.3.1 section functionality. This setting applies only to subsequent OLAP queries, and not an existing OLAP query. By default, new Essbase queries use the CubeQuery Section type. A pre Release 9.3.1 OLAP query and an OLAP query with a CubeQuery section type can coexist in the same Interactive Reporting document (BQY).</td>
</tr>
<tr>
<td>Show Dimension Root Level (ODBO Level 0/Essbase Generation 1) Data</td>
<td>Enable the display of level 0 and generation 1 data in the OLAP tree. If you intend to use read or set a full member level hierarchy data in the Dashboards section, this option needs to be enabled. This option does not affect the display of any information in the CubeQuery section.</td>
</tr>
</tbody>
</table>

### Table 38  Query Options

<table>
<thead>
<tr>
<th>Query</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto Refresh Query</td>
<td>Queries the database automatically when an item is added to or removed from the data layout, or when the Suppress and Replace options in Query Options are changed. If Auto-Refresh is disabled, you must click Process to query the database whenever you make a change in the data layout. To improve system performance, it is recommended that when you enable the Auto Generate Results option, you disable the Auto-Refresh Query option.</td>
</tr>
</tbody>
</table>
Database-specific OLAP Query Options

Database-specific OLAP Query options depend on the database to which you are connected. As a result, the content on the tab in the dialog box varies according to the type of database connection.

- “Display Options for Essbase Databases” on page 160
- “Display Options for Microsoft OLE/DB” on page 161

Display Options for Essbase Databases

The tab for Essbase databases is displayed as follows:

<table>
<thead>
<tr>
<th>Table 39</th>
<th>Display Option for Essbase Database</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Option</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td><strong>Query Options</strong></td>
<td>Defines query options in the following areas:</td>
</tr>
<tr>
<td></td>
<td>- <strong>Suppress Missing Rows</strong>—Suppresses the retrieval of any missing rows where all cells are null.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Suppress Zero Rows</strong>—Suppresses the retrieval of any zero rows where all cells are null.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Specify the number of decimal places to retrieve</strong>—Sets the number of decimal places that the server will return.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Enable Hybrid Analysis Value Retrieval</strong>—This feature allows you to retrieve the lowest levels of an Essbase cube that reside in an external relational database. These levels are not reflected in the cube structure (metadata) that is resident on the Essbase Server itself. Instead, their existence and retrieval is performed by the separate &lt;Essbase Integration Server product.</td>
</tr>
<tr>
<td></td>
<td>In order for this functionality to be performed, however, the Essbase server must first receive instructions to enable Hybrid Analysis retrieval. This instruction takes the form of an Essbase Report Script keyword:</td>
</tr>
<tr>
<td></td>
<td>&lt;HYBRIDANALYSISISON&gt;</td>
</tr>
<tr>
<td></td>
<td>Conversely, to disable Hybrid Analysis retrieval, the instruction keyword</td>
</tr>
<tr>
<td></td>
<td>&lt;HYBRIDANALYSISOFF&gt;</td>
</tr>
<tr>
<td></td>
<td>The applicable keyword used sent to the Essbase Server can be viewed on the Query Log.</td>
</tr>
</tbody>
</table>

This feature is only supported in Hyperion Essbase version 6.5 and IBM DB2 OLAP version 8.1.
### Option Description

**Alias Table**
- Defines the alias table to use in an OLAPQuery.
- **Select an Alias Table**—When you use aliases to assign user-friendly names to database physical member and/or generation/level names, Essbase stores the aliases in an Alias Table in the cube. Since a cube can have multiple alias tables, you can select the alias table to use and modify the query based on the value you enter.
- For example, you could define Store Category members as codes, but define an alias for each Store Category to use as a descriptive alias in the Alias Table.
- In the following example, for the member name "0199", you could see either "January 11999, "Jan99" or Fiscal Month 1" depending on the selected alias table. By default, uses the default alias table and if another alias table were not selected, you would see "January 1999."

<table>
<thead>
<tr>
<th>Physical Member Name</th>
<th>Default Alias Table Value</th>
<th>Alias Table 1 Value</th>
<th>Alias Table 2 Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0199</td>
<td>January11999</td>
<td>Jan99</td>
<td>Fiscal Month 1</td>
</tr>
</tbody>
</table>

### Display Options for Microsoft OLE/DB

The tab for Microsoft/OLE DB for OLAP databases is displayed as follows:

Table 40  Display Options for Microsoft OLE/DB

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Suppress Empty Rows</td>
<td>Suppresses the retrieval of any empty rows for which there is no measure data.</td>
</tr>
</tbody>
</table>

### Building OLAP Queries

Interactive Reporting uses the data layout to manage multidimensional information. The data layout enables you to plot, view, and manipulate dimensions. The three data layout panes are:

- **Side Labels**—Contains non-quantifiable dimensions or members.
- **Top Labels**—Contains non-quantifiable dimensions or members.
- **Measures**—Contains quantifiable dimensions or measures. The Measures pane in data layout holds the values of the cube.

**Note:** Interactive Reporting treats time and date dimensions as non-quantifiable values.

- To build an OLAP query:
  1. Select **Tools**, then **Options**, then **Program Options**, then **OLAP**.
     - The OLAP tab is displayed.
  2. Select **Use Old OLAP Query section** and click **OK**.
By default, new Essbase queries use the new Section type (CubeQuery). The Old OLAP query can be used by selecting Tools, then Options, then Program Options, then OLAP and then Use Old OLAP Query section.

3 Select Insert, then New Query.

The Insert Query dialog box is displayed.

4 Select an Interactive Reporting connection file in which to connection.

See Insert Query for more information.

The Query section is displayed.

5 If the data layout is not visible, click Data Layout on the Section title bar.

6 In the Catalog pane, select one or more measures (such as units or amounts) and choose OLAP, then Add Fact/Measure to add the item(s) to the Measures pane in data layout.

7 In the Catalog pane, select one or more levels or members and choose OLAP, then Add Side Labels to add the item to the Side Labels pane in data layout.

8 In the Catalog pane, select one or more levels or members and choose OLAP, then Add Top Label to add the item to the Top Labels pane in data layout.

➢ To refresh the dimension values in the Catalog pane, select OLAP, then Retrieve Dimensions.

When you drag items from the Catalog pane to the data layout, only the level names is displayed. For example, if you drag CA into the data layout, the data layout displays State. Level names are displayed in the data layout preceded by an icon, 📊.

You can reorient, or pivot, your OLAPQuery by interchanging the items in the top and side dimensions. This feature is useful for juxtaposing data in one dimension with data from other dimensions. By pivoting dimensions from the top to the side, alternate relationships become evident.

**OLAPQuery Section Data Layout Rules**

The following rules apply to the OLAPQuery section data layout:

- A dimension can be represented only on one axis. For example, if the level Year is in the Side Labels pane, you cannot drag the level Quarter to the Top Labels pane.

- If you move the level of one dimension, Interactive Reporting automatically move all levels of the same dimension.

- To pivot data in the OLAPQuery section, move the items in the data layout panes. You cannot use the label handles to pivot data.

- Levels from the same dimension must be grouped together in both the Side Labels and Top Labels panes.
For example, you cannot use the following order for side-label levels, since levels from different dimensions are mixed (Year, Quarter, and Month come from the Time dimension, Store Type comes from the Store dimension, and Product Category comes from the Product dimension).

Instead, the data layout requires that you use this order (Year, Quarter, and Month are all from they same dimension so they are grouped together).

- The hierarchy of a dimension cannot be broken. For example, Year must come before Quarter, which must come before Month.

- “Hybrid Analysis” (Essbase only)—If data is successfully retrieved from the Essbase servers, a “dummy” Level is added to the data layout called "Hybrid1". The new Level of data is shown in the Contents Pane. If data is not retrieved successfully from the Essbase servers (i.e. an error message is returned) the message "No more levels to drill into" message is displayed. If you subsequently drills-down on a returned Hybrid Analysis Level of data, the retrieved Hybrid Analysis Level of data is called "Hybrid2" and so on.

- OLE DB only —If you retrieve dimensional level properties from the database, you can drag each property into the data layout after you add its corresponding level.

For example, if the data layout contains Country, Year and you drag Manager, (a property of Country) into the data layout, the Manager property is kept with Country and not added after the Year level. A property cannot be dragged into the Slicer pane, the Measures pane in data layout, or have a filter applied to it.

Note that in order to retrieve dimensional level properties, you must enable the Show Member Properties checkbox when creating the Interactive Reporting database connection file.

### OLAPQuery Member and Level Rules

If you include both members and levels together in a query, a union of the two data sets occurs, and not an intersection. For example, if you select a State level and select San Francisco (which is a city), your query retrieves all states and San Francisco.

The following able shows the results of different queries.

<table>
<thead>
<tr>
<th>Data Layout Item(s)</th>
<th>Component Type</th>
<th>Query returns:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Country (All)</td>
<td>Level</td>
<td>All countries</td>
</tr>
<tr>
<td>State (All)</td>
<td>Level</td>
<td>All states regardless of country</td>
</tr>
<tr>
<td>City (All)</td>
<td>Level</td>
<td>All cities regardless of state or country</td>
</tr>
<tr>
<td>Canada</td>
<td>Member</td>
<td>Canada only</td>
</tr>
<tr>
<td>CA</td>
<td>Member</td>
<td>California only</td>
</tr>
<tr>
<td>San Francisco</td>
<td>Member</td>
<td>San Francisco only</td>
</tr>
<tr>
<td>State (All), San Francisco</td>
<td>Level, Member</td>
<td>All States regardless of country and San Francisco</td>
</tr>
</tbody>
</table>
Refining OLAPQuery Data

Once you have identified the items to include in your OLAP query, you can perform numerous operations to refine the data such as setting slicer filters, specifying drill data, adding computed items, data functions, and so on. Review the following sections for information on each of these functions.

- Specifying a Slicer
- Drilling Down
- Drilling Up
- Hybrid Analysis and Drilling (Essbase and DB2 only)
- Adding Computed Items
- Using OLAPQuery Functions

Specifying a Slicer

A slicer is a sort of third axis in a query that filters data. The other axes are the row axis and the column axis. A slicer defines a logical slice of the server cube by instructing the server to ignore all values not part of your slice. For example if you were running a query for general category stores, you could apply a slicer that slices the category stores into store subsets, such as computer stores, discount stores, and electronic stores.

When working with a slicer, use only an individual member from a dimension. The dimension cannot be used in a Top Label or Side Label (no dimension can be represented on more than one axis at any time).

**Tip:** A query can have multiple slicers, each from a different dimension.

➢ To specify a slicer:

1. **Click Slicer** on the Section title bar to open the Slicer pane.
2. **Select a member from a dimension in the Catalog pane and drag it to the Slicer pane.**
   Every dimension folder contains a members subfolder named “Values for …” that domain. The subfolder contains the members eligible for selection in the Slicer pane.
3. **Click Process.**
If you are running in Hardwire mode (see “Processing OLAP Queries Automatically” on page 169), the slice is applied instantly.

**Drilling Down**

The Drill Down feature retrieves data from the multidimensional database cube, following the hierarchy down to the granular level. When you want to learn more about an item, such as a product line, drill down into the item label. You can drill down on one or items.

For a member drill down, any Top Label or Side Label can be drilled down so that you can view the structure of the hierarchies for any particular dimension. Every time you select a specific label in a dimension row or column, show only the data for that label value. When you select the dimension tab for a level, show all the members of that dimension level.

Use one of the methods to drill down on a label:

- Double-click the label.
- Select the label and choose **Drill Down** on the shortcut menu.
- Select the label and choose **OLAP**, then **Drill Down**.

**Tip:** You can specify what level of data is the next level displayed when you drill down in an OLAPQuery.

**Note:** You cannot set filters while in a drilled-down state on a dimension.

**Note:** **Essbase only.** For a measure drill down, you can show how different measures consolidate. A drill down on a measure is done on a progressive basis, one level at a time on a 1 to n path (sequential rather than nested). For example, if Profit is the parent of Tax and Pre-Tax Profit, and Revenue and Expenses are children of Pre-Tax Profit, the Tax and Pre-Tax columns are drilled down first and you must double-click the Pre-Tax label to display the Revenue and Expense columns.

**Drilling Up**

If you used the Drill Down feature, you return to your original view of the data by drilling up one level at a time. To drill up, select the level to drill up to and use one of the following methods:

- Double-click the level.
- Select **Drill Up** on the shortcut menu.
- Select **OLAP**, then **Drill Up**.
Hybrid Analysis and Drilling (Essbase and DB2 only)

When you drill down to retrieve Hybrid Analysis data, the drill down retrieves only the next level of information, even if there is a selection made for “Drill Through”, or the "Drill to all Levels" or "Drill to Lowest Level" options have been enabled on the OLAP Query Options dialog.

If you can drill into data returned by way of hybrid analysis, Interactive Reporting adds a “dummy” level to the data layout called "Hybrid1" and displays the new Level of data accordingly in the Contents Pane. If data is not retrieved successfully from the Essbase servers, you get the message: "No more levels to drill into" message. If you continue to drill down on the Hybrid Analysis level of data, each successful drill down returns the data as "Hybrid2" in the data layout (and "Hybrid3", "Hybrid4”… etc. for any subsequent Hybrid Analysis Level retrievals).

You can use drill up on Hybrid Analysis data to redisplay that level as the lowest level in the Contents pane. All lower level Hybrid Analysis data is removed from the Contents pane and data layout.

Adding Computed Items

Computed items allow you to create a new column by building an expression to compute measures, or by applying functions to measures. Computed items are like normal data measures and can be included in reports or reused to compute other measures.

Computed items are displayed in virtual columns (as opposed to columns that are physically stored in the cube). They are automatically calculated during the query and supplement the information already stored in the database.

For example, you can modify the Amount Sold item by building an expression around it, multiplying by the Unit Price item, and renaming the resulting item Revenue.

Note: This feature is only available for an MS OLAP database.

To compute or modify a measure:

1 Select OLAP, then Add Computed Item.

The Modify Item dialog box is displayed.

2 Specify the information requested in the following fields:

   - **Name**—Specify a new column name that reflects the computation result.
   - **Definition**—Build an expression by adding items from the pad or the Functions dialog box.
   - Use the keypad to select and insert arithmetic and logical operators.
   - If you are familiar with MDX (Multi Dimensional eXpressions), type your instruction directly in the Definition field.
   - **Measure**—Select the MDX equivalent from the list of available measures for the expression.
Functions—Apply a numeric function to a selected measure in the Definition field. Depending on the function you select, the Functions dialog box changes to accommodate the selected function. For more information about functions, see “Using OLAPQuery Functions” on page 167.

3 When the expression is complete, click OK.

The new measure name is added to the data layout.

For more information on computed items, see Chapter 12, “Computed Items.”

Note: You can only add computed items if your database supports them. Examples of databases that support computed items are OLE DB for OLAP-compliant databases such as MS OLAP and SAP BW.

Using OLAPQuery Functions

Use OLAPQuery functions to insert standard numeric functions in computed measure expressions. Numeric functions compute a new measure for each value associated with it. You can use two types of OLAPQuery functions in the OLAPQuery section:

- Interactive Reporting Functions—Non-MDX functions that allow you to perform common mathematical computations in MDX. The % of Column, % of Row, and % of Total functions allow you to use only a measure name from the query (not all measures in the cube) as a parameter.

- MDX functions—Standard mathematical functions that you apply to computed item expressions. Interactive Reporting supports a number of MDX functions. (For more information on MDX functions, consult your MDX documentation.)

The table below provides a quick reference to the commands available on the Query menu and lists any related shortcuts.

The following table lists the functions available in the OLAPQuery section.

<table>
<thead>
<tr>
<th>Function</th>
<th>Type of Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>% of Column</td>
<td>Interactive Reporting</td>
<td>Calculates the value of the specified measure as a percentage of the total for the column.</td>
</tr>
<tr>
<td>% of Row</td>
<td>Interactive Reporting</td>
<td>Calculates the value of the specified measure as a percentage of the total for the row.</td>
</tr>
<tr>
<td>% of Total</td>
<td>Interactive Reporting</td>
<td>Calculates the value of the specified measure as a percentage of the total for all rows and columns.</td>
</tr>
<tr>
<td>% Change</td>
<td>Interactive Reporting</td>
<td>Calculates the percentage change of the specified measure for a particular dimension from the previous member in that dimension. For example, this function could be used to calculate the percentage change from sales from the previous year.</td>
</tr>
<tr>
<td>Function</td>
<td>Type of Function</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Absolute Change</td>
<td>Interactive</td>
<td>Calculates the absolute change of the specified measure for a particular dimension from the previous member in that dimension. For example, this function could be used to calculate the difference in sales from the previous year.</td>
</tr>
<tr>
<td>Avg</td>
<td>MDX</td>
<td>Calculates the average of the selected measure evaluated over the specified dataset.</td>
</tr>
<tr>
<td>Correlation</td>
<td>MDX</td>
<td>Returns the correlation of a dataset against two measures.</td>
</tr>
<tr>
<td>Count</td>
<td>MDX</td>
<td>Calculates the number of members for the specified dimension in the report.</td>
</tr>
<tr>
<td>Covariance</td>
<td>MDX</td>
<td>Measures the tendency of two values to vary together. Variance is the average of the squared deviation of a value from its mean. The covariance is the average of the values of the deviations of feature values from their means.</td>
</tr>
<tr>
<td>Linregpoint</td>
<td>MDX</td>
<td>Calculates the linear regression of a dataset and returns the value of “b” in the regression line $y = ax + b$.</td>
</tr>
<tr>
<td>Linreg2</td>
<td>MDX</td>
<td>Calculates the linear regression of a dataset and returns $r^2$ (the coefficient of determination).</td>
</tr>
<tr>
<td>Linregslope</td>
<td>MDX</td>
<td>Calculates the linear regression of a dataset and returns the value of “a” in the regression line $y = ax + b$.</td>
</tr>
<tr>
<td>Linregvariance</td>
<td>MDX</td>
<td>Calculates the linear regression of a dataset and returns the variance that fits the regression line $y = ax + b$.</td>
</tr>
<tr>
<td>Max</td>
<td>MDX</td>
<td>Returns the maximum value of the selected measure evaluated over the specified dataset.</td>
</tr>
<tr>
<td>Median</td>
<td>MDX</td>
<td>Calculates the median value of the selected measure evaluated over the specified dataset.</td>
</tr>
<tr>
<td>Min</td>
<td>MDX</td>
<td>Returns the minimum value of the selected measure evaluated over the specified dataset.</td>
</tr>
<tr>
<td>Stdev</td>
<td>MDX</td>
<td>Calculates the standard deviation of the selected measure evaluated over the specified dataset.</td>
</tr>
<tr>
<td>Sum</td>
<td>MDX</td>
<td>Calculates the sum of the selected measure evaluated over the specified dataset.</td>
</tr>
<tr>
<td>Variance</td>
<td>MDX</td>
<td>Calculates the variance of the selected measure evaluated over the specified dataset.</td>
</tr>
</tbody>
</table>

To apply a data function:

1. Select OLAP, then Add Computed Item, and then click Functions in the Modify Item dialog box.

The Functions dialog box is displayed.

2. Select the Oracle Hyperion Interactive Reporting Web Client or MDX function from the Functions list.

A description of the selected function is displayed below the Functions list and explains the type of calculation the function performs.

3. Select the measure.

You can select any measure in the cube, not just a measure in the query. The Cube Hierarchy list shows the organization of the cube including both members and levels.

Some functions require that you specify a second measure to perform the calculation. See the specific function if you are required to specify a second measure.
The Count function requires no measure.

4 Define the dataset to evaluate the function and click OK.

To add a member or level to the dataset from the Cube Hierarchy list, select a member or level, and click Add.

To remove a member or level from the dataset, select a member or level and click Remove.

Processing OLAP Queries

After you build your OLAP query and apply filters, computations, sorts, and any other adjustments to further refine your request, you need to process it. Processing your query may take a few moments if your query is complex, or if the data in linked report sections needs to be refreshed.

To process an OLAP query, select Tools, then Process Query, and then Option.

Since a document can contain multiple queries, the Process drop-down list has three processing options:

- **Process Current**—Processes the current object. In some cases more than one query may be processed, for example, if a report references results sets from multiple queries. Process Current is the default selection when using the toolbar button.
- **Process All**—Processes all the queries in the document.
- **Process Custom**—Opens the Process Custom dialog box so that you can indicate which queries to process by selecting a query’s check box.

Interactive Reporting sends the query to the database and retrieve the data to the OLAPQuery section. While the data is being retrieved, the Status bar displays a dynamic count indicating rate and progress of server data processing and network transfer.

Processing OLAP Queries Automatically

If you select to run in Hardwire mode, Interactive Reporting queries the database automatically when you add an item to or remove an item from the data layout and instantaneously retrieves the data. You do not have to click Process.

**Note:** You should consider the size of the cube you are querying to determine whether to use Hardwire mode.
To select Hardwire mode, select **OLAP**, then **OLAP Query Options** and then **Hardwire Mode** in the Design sectionally the General tab.

## Working with an OLAPQuery Offline

To view, plot, and work with an OLAPQuery offline, download the data set to an OLAPResults section within the document. Once downloaded, the data can be integrated with the Chart, Table, and other reporting sections. If you need to modify the query, reconnect to the database and apply any necessary changes.

To download the OLAPQuery data set, select **OLAP**, then **Download to Results**.

An OLAPResults section is created for the query. You can use the OLAPResults data set to insert a new chart, pivot, or other report.

## Creating a OLAPResults Section Automatically

You can have a Results section created automatically when you click **Process**. This eliminates the need to select **OLAP**, then **Download to Results** (see “Working with an OLAPQuery Offline” on page 170).

To automatically create a Results section when you click **Process**:

1. Select **Tools**, then **Options**, then **Program Options** and choose the **OLAP** tab.
2. Select **Auto-Generate Results Section When Processing an OLAP Query**.

When you select this option, Interactive Reporting creates an OLAPResults section automatically for any OLAPQuery section that you create in this session when that OLAPQuery section is first processed.

## Applying Filters

Filters enable you to define and apply filters to a query once Top Labels or Side Labels have been added to the query. You set filters by applying comparison operators on the values for a specific member. Review the following sections for information on:

- **Applying Member Selection Filters**
- **Applying Measure Filters (Essbase)**
- **Applying Variable Filters**

## Applying Member Selection Filters

Use a member selection filter to filter data retrieved from the server cube. A member selection filter is similar to a slicer except that the member selection filter introduces the member value in your report, and multiple members may be selected from a single hierarchy.
To apply a member selection filter, drag individual member values from the Catalog pane to the data layout:

Another method of member selection uses an expression to dynamically retrieve the list of members that satisfy selected parameters, for example, the Top N or Bottom N. You specify these parameters in the Filter dialog box.

**Note:** If you used the Drill down feature to navigate down to a members level, you have to use the drill up feature to return to the original level before you apply Member Selection filters.

To apply a member selection filter using an expression:

1. **Drag a level into the data layout and double-click the level name.**
   The Filter dialog box is displayed.
2. **Specify the information requested in the Filter dialog box and click OK.**

### Applying Measure Filters (Essbase)

You can filter data retrieved from the server cube with a measure filter, which is similar to a member selection filter. A measure filter uses an expression to dynamically retrieve the list of measures that satisfy selected parameters, for example, the Top N or Bottom N. You specify these parameters in the Filter dialog box. Additional parameters are available based on the selected multidimensional database.

To apply a measure filter:

1. **Double-click a measure in Data Layout.**
   The Filter dialog box is displayed.
2. **Select the data operator from the Data Operator drop-down list.**
   The selections shown on this list depend on the database to which you are connected. See “Operator Types and Data Operators” on page 174.
3. **Specify any database specific parameter requests, such as a column index (that is, the column on which to apply the measure filter) or value.**
4. **Move the member(s) to the Applied Filters list and click OK.**

### Applying Variable Filters

A variable filter is a filter you specify when you process a query. You can use variable filters for standardized documents that you distribute to many users, or to automatically reset filters when you need new conditions every time you run a particular query. A (v) next to an item indicates it has a variable filter.
You place a variable filter on an item using the Filter dialog box, which is accessed in either the Slicer pane or the data layout, depending on your database. The table below specifies how various OLAP databases access the Filter dialog box.

<table>
<thead>
<tr>
<th>Database</th>
<th>Access Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Essbase</td>
<td>Side Labels pane, Top Labels pane, Measures pane, Slicer pane</td>
</tr>
<tr>
<td>OLE DB</td>
<td>Side Labels pane, Top Labels pane, Slicer pane</td>
</tr>
</tbody>
</table>

**Member Variable Filters**

- To set a *member* variable filter:
  1. In the data layout, double-click the item you want to define as a member filter.
     - The Filter dialog box is displayed.
  2. Select the method for retrieving items from the database from the Operator Type list.
     - The selections shown on this list depend on the database to which you are connected. See “Operator Types and Data Operators” on page 174.
  3. Specify any database-specific parameter requests.
  4. Click **Show Values** to display values in the database.
  5. In the Members field, select a *member(s)*.
  6. Move the member(s) to the **Applied Filters** list.
  7. Click **Set As Variable** and click **OK**.
     - The OLAPQuery section is redisplayed.
  8. Click **Process**.
     - The Filter dialog box is redisplayed.
  9. Select the *member(s)* that you want to define as a variable filter in the Members field.
  10. Move the member(s) as a variable filter to the **Applied Filters** list and click **OK**.

**Measure Variable Filters**

- To set a *measure* variable filter:
  1. Double-click the item to define as a measure filter in the data layout.
     - The Filter dialog box is displayed.
  2. Select the data operator from the Data Operator drop-down list.
     - The selections shown on this list depend on the database to which you are connected.
  3. Specify any database-specific parameter requests, such as a column index or value.
4 Move the measure to the Applied Filters list.
5 Click Set As Variable and click OK.
The OLAPQuery Section is redisplayed.
6 Click Process.
The Filter dialog box is redisplayed.
7 Select a measure variable filter by selecting the data operator from the Data Operator drop-down list and specifying any database-specific parameter requests, such as a column index or value.
8 Move the measure variable filter to the Applied Filters list and click OK.

**Slicer Variable Filters**

To set a slicer variable filter:
1 Click Slicer on the Section title bar to open the Slicer pane.
2 Select a member from a dimension in the Catalog pane and drag it to the Slicer pane.
   Every dimension folder contains a members subfolder named Values For Domain, which contains the members that are eligible for selection in the Slicer pane.
3 Double-click the member in the Slicer pane.
The Slicer dialog box is displayed.
4 Select a member.
5 Click the Set As Variable field and click OK.
The OLAPQuery section is redisplayed.
6 Click Process.
The Slicer dialog box is redisplayed.
7 Select the member(s) you want to use as a variable filter and click OK.

**Essbase Substitution Variables**

In Essbase, a substitution variable acts as a dynamic filter. Substitution variables are defined on the server using Essbase Application Manager. Your administrator names the substitution variable and set its value equal to a user-defined parameter. For example, a variable might be named Latest Period and have its value set to equal November. When the latest period ends, the variable’s value could be reset to December, and so on.

An advantage of this type of variable is that saved queries capture the variable’s name instead of hard coding the actual value. Each time the query is run, different data could be returned if the variable’s value has been changed on the server.

To apply a substitution variable:
1 Double-click an item in the data layout.
The Filter dialog box is displayed.

2. Click **Substitution Variable** in the Operator Type drop-down list.
   The Filter dialog box retrieves all available variables in the Substitution Variables list.

3. Move the member(s) to the **Applied Filters** list and click **OK**.
   You can also double-click the member to add it to the Applied Filters list.
   The OLAPQuery section is redisplayed.

4. Click **Process**.
   The Filter dialog box is redisplayed.

5. In the Members field, select the member(s) to use as a **variable filter**.

6. Move the member(s) to the Applied Filters list and click **OK**.

### Operator Types and Data Operators

The OLAPQuery section supports the following operator types and data operators:

- OLE DB Operator Types and Data Operators (see Table 44)
- Essbase Operator Types and Data Operators (see Table 45)

**Note:** Not all providers support all operators.

**Table 44  OLAP Query Operator Types and Data Operators**

<table>
<thead>
<tr>
<th>Operator Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Select Members from DB</strong></td>
<td>Creates a filter based on a member(s) retrieved from the database. (Supported for the Essbase, MSOLAP and SAP/BW databases).</td>
</tr>
<tr>
<td><strong>Select by Measure</strong></td>
<td>Creates a filter based on a measure that you specify.</td>
</tr>
<tr>
<td><strong>Top N</strong></td>
<td>Retrieves only the top N values where each top N value is at least the specified Index value.</td>
</tr>
<tr>
<td><strong>Top N %</strong></td>
<td>Retrieves only the top N % values where each top N % value is at least the specified Index value.</td>
</tr>
<tr>
<td><strong>Top Sum</strong></td>
<td>Retrieves the top N (the smallest number possible) values, such that their sum is at least the specified Index value.</td>
</tr>
<tr>
<td><strong>Bottom N</strong></td>
<td>Retrieves only the bottom N values where each bottom N value is at least the specified Index value.</td>
</tr>
<tr>
<td><strong>Bottom N %</strong></td>
<td>Retrieves the bottom N% where each bottom N % value is at least the specified Index value.</td>
</tr>
<tr>
<td><strong>Bottom Sum</strong></td>
<td>Retrieves the bottom N (the smallest number possible) values such that their sum is at least the specified Index value.</td>
</tr>
<tr>
<td><strong>Data Operator</strong></td>
<td>Description</td>
</tr>
<tr>
<td><strong>= Equal</strong></td>
<td>Retrieves only records where the filtered item equals the specified value(s).</td>
</tr>
<tr>
<td><strong>&lt;&gt; Not Equal</strong></td>
<td>Retrieves only records where the filtered item does not equal the specified value(s).</td>
</tr>
</tbody>
</table>
## Changing Data Views

In the OLAPQuery section, there are numerous ways to change the way you view the data. Changing data view involves:

- Suppressing Rows
- Adding Totals
- Adding Data Functions
- Showing OLAP Results as a Chart

### Suppressing Rows

You can suppress the following types of rows:

<table>
<thead>
<tr>
<th>Operator Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; Less than</td>
<td>Retrieves only records where the filtered item is less than the specified value(s).</td>
</tr>
<tr>
<td>&lt;= Less than or Equal to</td>
<td>Retrieves only records where the filtered item is less than the specified value(s).</td>
</tr>
<tr>
<td>&gt; Greater than</td>
<td>Retrieves only records where the filtered item is greater than the specified value(s).</td>
</tr>
<tr>
<td>&gt;= Greater than or Equal to</td>
<td>Retrieves only records where the filtered item equals, or is greater than the specified value(s).</td>
</tr>
</tbody>
</table>

### Match Member

The Match Member performs a trailing-wildcard member search for member names and alias names that match the pattern you specify. It returns the member and alias names it finds. The search uses the Essbase formula: `mbrName | genName | levName, "pattern")`. The character pattern to search for, including a wildcard character (`*` or `?`). `?` substitutes one occurrence of any character. You can use `?` anywhere in the pattern. `*` substitutes any number of characters. You can use `*` only at the end of the pattern. To include spaces in the character pattern, enclose the pattern in double quotation marks (`"`).

---

**Table 45  Essbase Operator Types and Data Operators**

<table>
<thead>
<tr>
<th>Operator Type/ Data Operator</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equal</td>
<td>Retrieves only records where the filtered item equals the specified value(s).</td>
</tr>
<tr>
<td>Match Member</td>
<td>The Match Member performs a trailing-wildcard member search for member names and alias names that match the pattern you specify. It returns the member and alias names it finds. The search uses the Essbase formula: `mbrName</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>= Equal</td>
<td>Retrieves only records where the filtered item equals the specified value(s).</td>
</tr>
<tr>
<td>&lt;&gt; Not Equal</td>
<td>Retrieves only records where the filtered item does not equal the specified value(s).</td>
</tr>
<tr>
<td>&lt; Less than</td>
<td>Retrieves only records where the filtered item is less than the specified value(s).</td>
</tr>
<tr>
<td>&lt;= Less than or Equal to</td>
<td>Retrieves only records where the filtered item is less than the specified value(s).</td>
</tr>
<tr>
<td>&gt; Greater than</td>
<td>Retrieves only records where the filtered item is greater than the specified value(s).</td>
</tr>
<tr>
<td>&gt;= Greater than or Equal to</td>
<td>Retrieves only records where the filtered item equals, or is greater than the specified value(s).</td>
</tr>
</tbody>
</table>
- **Missing Rows** (*Essbase only*)—Suppresses the retrieval of any missing rows where all cells are null.

- **Zero Rows** (*Oracle Essbase only*)—Suppresses the retrieval of any zero rows where all cells are null.

- **Empty Rows** (*OLE DB only*)—Suppresses the retrieval of any empty rows for which there is no measure data.

To suppress rows in your OLAPQuery data, select OLAP, then OLAP Query Options and click the DB Specific tab.

The options that are displayed on the DB Specific tab depend on the database to which you are connected. See “Database-specific OLAPQuery Options” on page 160 for more information.

## Adding Totals

Interactive Reporting enables you to add either database totals or local totals. Database totals are calculated by querying the actual database. Local totals are calculated and applied to surface values.

### Adding Database Totals

You can include or exclude database totals to tailor the look of the OLAPResults section and any charts you create using the Show As Chart feature. When you activate the Database Totals feature, totals are retrieved into the OLAPResults section as additional rows or columns. In the Chart section, database totals are plotted.

The OLAPQuery section includes database totals by default. You may wish to turn off this feature if you intend to use the Drill Down feature to navigate the multidimensional cube, or if you plan to export the OLAPQuery section.

To add database totals:

1. Select OLAP, then OLAP Query Options and then select Database Totals in the Design section of the General tab.

2. Click Process to add the totals to the data.

Interactive Reporting displays the result as the first item at each level of the dimension.

**Note:** If you enable database totals in the OLAPQuery section, totals are copied as static values into any Results section. As a result, they will not be treated as dynamically updated totals.
To remove database totals, remove the check mark next to Database Totals on the General tab and click Process to reprocess the query.

Adding Local Totals

To add local totals, select the desired dimension handle, right-click, and choose Add Totals.

Interactive Reporting totals the data and displays the result as the last item at each level of the dimension.

To remove local totals, click on a total label and press the Delete key.

Adding Data Functions

Column or row totals added to the OLAPQuery section are aggregates and can be recalculated using data functions. When applied to totals, data functions apply the calculation to surface values.

When applied to surface values, data functions recalculate the values in the visible cells or surface of the OLAPQuery section. For example, you can show the total sale, average sale, and maximum sale of each product by quarter. Each of these dimensions is based on the same value. They only differ in the data function that is applied.

To apply a total function:

1. Select the dimension handle for a particular measure.
2. Click on the Standard toolbar to calculate the total.
3. Select a column of a particular measure.
4. Choose OLAP, then Data Function and then select a function.

The following table lists the data functions available in the OLAPQuery section.

<table>
<thead>
<tr>
<th>Data Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sum</td>
<td>Returns sum of all values. This is the default function in all report sections.</td>
</tr>
<tr>
<td>Average</td>
<td>Returns average of all values.</td>
</tr>
<tr>
<td>Count</td>
<td>Returns number of values.</td>
</tr>
<tr>
<td>Maximum</td>
<td>Returns highest value.</td>
</tr>
<tr>
<td>Minimum</td>
<td>Returns lowest value.</td>
</tr>
<tr>
<td>% of Column</td>
<td>Returns surface values as a percentage of their respective column item.</td>
</tr>
</tbody>
</table>
### Showing OLAP Results as a Chart

You can perform interactive analysis on the OLAPQuery data by viewing the data as a chart. When you select this option, an OLAPQuery Results section is automatically created, as an OLAP Chart section.

**Note:** Because the Chart created by the Show As Chart command is stationary, you cannot perform drill-down analysis on it. In addition, it is recommended that you deactivate the Database Totals feature since the chart plots the totals when totals are retrieved from the database.

➢ To show the query as a chart, select OLAP, then Show as Chart.

### Formatting OLAPQuery Items

You can use the commands available on the Format menu to add corner and data labels, and to change the appearance of fonts, backgrounds, borders, color, row heights, and column widths.

### Drilling Through from a Multi-Dimensional Database to a Relational Database

In general, OLAP data is aggregated and obtained from a relational database source. As a result, there may be occasions where you want to see the relational data associated with the multi-dimensional data.

For example, assume that you create an OLAP analysis to show your company’s sales aggregated to Country, State, and City levels. Assume further that the sales data for each store within a city is stored in a transactional, relational database. In this case, you could drill down in the OLAPQuery section to sales data for USA, then California, and then San Francisco. To see the data for the stores in San Francisco, however, you would need to drill through to a relational database.

Drilling through from a multi-dimensional database to a relational database involves:

- Setting Drill-through Options
- Drilling Through
**Setting Drill-through Options**

Drill-through options define the mapping between a multi-dimensional database and a relational database.

► To set drill-through options:

1. **Select OLAP, then Set Drill Through.**

   The Set Drill-Through dialog box is displayed.

2. **Select options in the Set Drill-Through dialog box and click OK.**

<table>
<thead>
<tr>
<th>Table 47</th>
<th>OLAPQuery Drill Through Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>Dimensions</td>
<td>The dimensions in the current OLAP query. Click a dimension to select it.</td>
</tr>
<tr>
<td>Selected Dimension</td>
<td>The dimension selected by the user. This is the dimension to use for mapping.</td>
</tr>
<tr>
<td>Specify Relational Query</td>
<td>The relational query sections that are currently available. To specify a relational query, click the arrow to the right of the drop-down list and select a query from the list that is displayed.</td>
</tr>
<tr>
<td>Relational Topics</td>
<td>The topics contained in the selected relational query. Click a topic to select it.</td>
</tr>
<tr>
<td>Selected Topic</td>
<td>The topic selected by the user. This is the topic to use for mapping. The selected topic should have topic items with names corresponding to the selected dimension levels.</td>
</tr>
<tr>
<td>Map/UnMap Dimension</td>
<td>Maps the selected dimension to a topic (or unmaps the selected dimension from a topic). When you map a dimension to a topic, Interactive Reporting stores an internal link between the dimension and the selected topic.</td>
</tr>
<tr>
<td>Specify Fact Topic</td>
<td>The topic used to map to the OLAP measures. The selected topic should have topic items with names corresponding to the OLAP measures. To specify a topic, click the arrow to the right of the drop-down list and select a topic from the list that is displayed. The topics that appear are the topics available in the selected query.</td>
</tr>
</tbody>
</table>

**Tip:** To drill-down to any level in the relational data, enable the “Set as Dimension” property for the relational topics that represent the OLAP dimension data. To do this, right-click the topic in the original relational query section Contents pane, select Properties, and click the checkbox next to “Set as Dimension” in the Topic Item Properties dialog box.

**Drilling Through**

► To drill through from a multi-dimensional database to a relational database:

1. **Select a dimension and drill-down to the lowest level.**

2. **Do one of the following:**
   - Double-click the dimension.
Select OLAP, then **Drill Through**.

The Drill-Through dialog box is displayed enabling you know that there are no additional OLAP levels to drill into.

3 **Click Yes to drill through to the relational data source.**

Interactive Reporting creates a new Pivot section (along with associated Query and Results sections) for the relational data.

**Note:** Interactive Reporting ignores slicers when drilling through to a relational database. If a slicer is present, a message is displayed letting you know that the slicer will be ignored.

**Tip:** After you drill through on a dimension and create a new Pivot section, you can return to the OLAPQuery section and drill down on additional dimensions if desired. When you drill down on an additional dimension, you can choose whether to create a new Pivot section or update a pivot section with the new dimension data.

**Tip:** If you update a pivot section with new data, ensure that the new data to be added to the Pivot section maps to a Fact Topic that is the same as the Fact Topic in the pivot section.

## OLAP Menu Command Reference

The following table provides a quick reference to the commands available on the OLAP menu and lists any related shortcuts.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Keyboard Shortcut</th>
<th>Shortcut Menu</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Retrieve Dimensions</strong></td>
<td>Refreshes the dimension values in the Catalog pane.</td>
<td>F9</td>
<td></td>
</tr>
<tr>
<td><strong>Add Side Label</strong></td>
<td>Adds the selected item to the Side Labels pane.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Add Top Label</strong></td>
<td>Adds the selected item to the Top Labels pane.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Add Fact/Measure</strong></td>
<td>Adds the selected item to the Measures pane.</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td><strong>Add Computed Item</strong></td>
<td>Opens the Computed Items dialog box. (This feature is for MS OLAP only)</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td><strong>Drill Down</strong></td>
<td>Allows you to progressively narrow your focus on a selected item.</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td><strong>Drill Up</strong></td>
<td>Returns the original view of data.</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td><strong>Drill Through</strong></td>
<td>Drills through from a multi-dimensional database to a relational database.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Remove Total</strong></td>
<td>Removes local totals from selected dimensions.</td>
<td>Del</td>
<td>*</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
<td>Keyboard Shortcut</td>
<td>Shortcut Menu</td>
</tr>
<tr>
<td>----------------------</td>
<td>------------------------------------------------------------------</td>
<td>-------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Hide Items</td>
<td>Removes selected items from the OLAPQuery report.</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>Show Hidden Items</td>
<td>Retrieves hidden items from the selected row or column to the OLAPQuery report.</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>Show All Items</td>
<td>Retrieves all hidden items to the OLAPQuery report.</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>Download To Results</td>
<td>Downloads the OLAPQuery data set to an OLAPResults section for offline work.</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>Show As Chart</td>
<td>Charts the OLAPQuery data set; automatically creates OLAPChart and OLAPResults sections.</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>OLAP Query Options</td>
<td>Accesses the OLAPQuery Options dialog box, where you can set options for your OLAPQuery.</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>Data Functions</td>
<td>Recalculates the surface values in the OLAPQuery data set.</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>Set Drill-Through</td>
<td>Accesses the Set Drill-Through dialog box, where you can define the options for drilling through from a multi-dimensional database to a relational database.</td>
<td>*</td>
<td></td>
</tr>
</tbody>
</table>
Pivot Section

The Pivot section enables you to extract meaningful information from your query results. Pivot tables are interactive tools used to slice and dice data for ad-hoc, interactive, and multidimensional analysis. Pivot tables enable you to add, move, rename, focus on, and group dimensions to gain customized views of the data. You can use the Pivot section to rotate or pivot rows and columns to see different summaries of data or display the details for areas of interest. You also can automatically include subtotals and grand totals, or use your own formulas by adding computed items.

Creating a Pivot Table

Pivot tables are made up of:

- **Facts**—Core numeric data that you slice and dice dimensionally in your analysis.
- **Dimensions**—Descriptive items that break aggregate data (facts) into logical categories. In the Pivot section, dimensions are either Column Labels or Row Labels.

For example, if you analyze Unit Sales by region, the numbers are your data values or facts. Region is a dimension. Presented in aggregate, facts are subdivided by your chosen dimension labels.

To create a pivot table:

1. Select Insert, then New Pivot.
If the data layout is not visible, click **Data Layout** on the Section title bar.

Perform one of the following actions:

- Drag each Request item to be included in the table from **Elements** to a pane in the data layout (Column Labels, Row Labels, or Facts).
- From **Elements**, select one of more Request items and select **Pivot**, then **Add Selected Items**, and then **Column Labels, Row Labels or Facts**.

**Tip:** Add Request items such as *Unit Sales* or *Amount Sales* to the Facts pane in data layout to create the data grid. Add dimensions such as *Product Line* or *Region* to the *column and row labels* panes in the data layout to create subdivisions.

**Tip:** You can select and drag multiple items to the same data layout pane to add multiple values or dimensional levels for analysis. Items are hierarchically ordered in the sequence in which they are displayed in the data layout pane.

### Pivoting Data

Use the Pivot feature to reorient the axes of a *pivot table* and view your data in new ways. *Pivoting* a table enables you to more easily compare the new data to data in the originating table, which makes pivot tables more powerful than common spreadsheets.

By default, pivot handles are transparent. They only appear when you select a label from a top or side. When the pivot handle appears, it is shown in white with a light grey handle (—). Once the selection is off, the pivot handle is transparent again.

To pivot data in a table, Select a dimension handle and drag it to any position on the same or opposite dimensional axis.

### Charting a Pivot Table

You can automatically generate charts from your current pivot table to view a graphic representation of your data.

To automatically chart your pivot table, select **Insert**, then **Chart This Pivot**.

Interactive Reporting creates a new Chart section that displays a bar chart based on the data from the pivot table.

### Working with Pivot Tables

You can select pivot table elements and perform a wide range of tasks with data elements. Since report sections organize data hierarchically, if you alter a data value item, all instances of the item within the report are affected. Review these topics for information:
Selecting Pivot Table Elements

To select a facts column for formatting, layout or modification, click anywhere on the column of data values. Do not click on the label.

To select a column of Row Labels or a row of Column Labels for formatting, layout, or analysis, click the dimension handle at the end of the column or row of labels.

By default, pivot handles are transparent. They only appear when you select a label from a top or side. When the pivot handle appears, it is shown in white with a light grey handle ( ). Once the selection is off, the pivot handle is transparent again.

To select one complete row or column for formatting or analysis, press the modifier key Alt for Windows or Ctrl+Alt for Motif. Then, select the row or column label.

To select an individual Column Label or Row Label for formatting, select the label itself.

Moving Pivot Table Elements

To move a column to a new location in the Content pane, select the column in the Content pane and drag it to a new position.

To remove Request items from the data layout or columns from the Content pane, select the element and select Pivot, then Remove Selected Item.

Note: When you delete a Request item from the data layout or a column from the Content pane, you cannot use the Undo feature to reinsert the column or Request item.

To move a pivot element item using the data layout, select the item name in the data layout pane, drag the item to a new position or to another data layout pane.

The display updates to reflect the repositioning or reassignment of the item.
Note: To move items between Column Labels or Row Labels panes and the Fact pane (or vice versa), you must first remove them from the data layout, and then add them again to the chosen pane.

**Changing Label Nesting Levels**

In pivot tables, labels from one dimension frequently are nested within another dimension. Nesting means that one set of labels is displayed as a subdivision of labels at a higher layer of data. You can change the way labels nest to emphasize different relationships.

For example, you can show Year and Quarter as data items in the Column Labels pane in the data layout. The Quarter labels (Q1, Q2, Q3, and Q4) are nested within each year label (1998, 1999). If you move Year after Quarter, each year is displayed as a subset of each quarter. Q1 values are broken down by labels 1998, and 1999.

To change the nested level of labels:
1. With more than one data item in an data layout pane, select a data item.
2. Drag that item to the other side of the second data item in the same pane in the data layout.

The labels in those dimensions switch positions and the data is nested in a different manner.

**Sorting Pivot Tables**

Sorting facts or dimensions enables you to display objects in ascending and descending order according to value.

To sort plotted values and labels:
1. If the Sort line is not visible, click Sort on the Section title bar.
2. In the Sort list, select an item to use as the basis of your sort.
3. In the By drop-down list box, click the sort type (either a label or value).
4. In the Using drop-down list box, select the method of calculation for a data value.

By default Interactive Reporting plots data in ascending order. To sort in descending order, click the descending icon.

**Hiding Pivot Facts**

You can restrict the data displayed in Pivot fact columns using the Hidden Items feature. When you select this feature, the fact column is removed from the report, but not the data layout. You can restore the hidden data item at any time.

To hide a fact column, select one or more fact column names in the Facts data layout pane and choose Hide Items from the shortcut menu.

The fact column name is dimmed in the data layout.
To restore a hidden fact column, select the dimmed column in the Facts data layout pane and choose Show Hidden Items from the shortcut menu.

## Formatting Pivot Items

The following table lists common formatting techniques that you can use when working with pivot tables.

<table>
<thead>
<tr>
<th>Table 49 Formatting Pivot Items</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Format</strong></td>
</tr>
<tr>
<td>Font</td>
</tr>
<tr>
<td>Style</td>
</tr>
<tr>
<td>Number</td>
</tr>
<tr>
<td>Justify</td>
</tr>
</tbody>
</table>

## Using Data Calculations in Pivot Tables

The Pivot section provides a number of ways to perform data calculations that can help you analyze business trends. These calculations range from simple totals and subtotals that are useful in most types of pivot tables to more complex data function for specialized contexts. Review the following sections for information on:

- Adding Totals and Subtotals
- Adding Cumulative Totals
- Using Data Functions
- “Using True Computed Item Totals” on page 190
- Using Surface Values in Data Functions
- Using Weighted Averages
- Adding Computed Items

### Adding Totals and Subtotals

You can calculate totals for both columns and rows in a pivot table. If you layered dimension items along the column or side of your pivot table, you can calculate totals for any *layer* in the *hierarchy*. When you select inner dimensions for totaling, subtotals are created for each of the categories in the outer dimensions.
For example, assume your pivot table has facts of Units and Amount Sold. These facts are further broken down by Region and Territory on the side, and by Year and Quarter on column. Calculating totals by Region produces a total row at the bottom of the pivot table, summing the data from all regions for each column. Calculating totals by Quarter produces one total column under each year label, summing the data for each set of four Quarter labels.

Tip: An intelligent aggregate is applied to the specified data when totaling unless you specify otherwise. For example, the total of a column of averages calculates an average rather than a sum total.

To add totals to a pivot table:

1. Click a row or column dimension.

2. Select Pivot, then Add Totals or click on the standard toolbar.

   The totals and breaks them according to the next higher dimension item.

To add subtotals to pivot tables:

1. Select an inner dimension.

2. Select Pivot, then Add Totals or click on the standard toolbar.

   Interactive Reporting adds subtotals to each one of the categories of the next higher dimension.

### Adding Cumulative Totals

Add cumulative totals to break totals by dimension and restart them at each dimensional grouping in a pivot table. Cumes work best when all dimensions are located at the column or side of the pivot table, and data label column heads are placed orthogonally.

To add a cumulative total:

1. Select a fact in the data grid of the pivot table.

2. Select Pivot, then Add Cume.

   The Pivot Cume dialog box is displayed.

3. If desired, type a new name for the pivot cume.

4. Select the scope of the pivot cume from the drop-down list box.
The Scope drop-down list box includes all of the dimensions in the pivot table. The default scope is the lowest level dimension that is displayed in the pivot table.

5. Click OK.

A new fact column is added that maintains a cumulative running total of the original fact by the dimension (scope) specified.

### Using Data Functions

A data function enables you to change the nature of the values displayed in a pivot table and enables you to decide the kind of value represented in a pivot table. When you use a data function, Interactive Reporting recalculates the selected values according to the function applied to the underlying data values (which are originally from the Results section).

Data functions are particularly useful if you want to display different types of values side by side. If you add the same fact (such as Amount Sales) to the data layout several times, you can apply a different data function to the very same dimension.

For example, you can show the total sale, average sale, and maximum sale of each product by quarter. Each of these computed items uses Amount Sales as its underlying value. They only differ in the data function used to calculate them.

**Note:** When you add multiple instances of a Request item to the Facts pane in the data layout, Interactive Reporting appends number to the name (for example, Amount_2, Amount_3).

To apply a data function:

1. **Select a fact in the data grid of the pivot table.**
2. **Select Pivot, then Data Function, and then Function.**

The data values are recalculated and populate the row or column of the pivot table.

The table below lists the data functions available in the Pivot section.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sum</td>
<td>Returns sum of all values. This is the default setting.</td>
</tr>
<tr>
<td>Average</td>
<td>Returns average of all values.</td>
</tr>
<tr>
<td>Count</td>
<td>Returns number of values.</td>
</tr>
<tr>
<td>Maximum</td>
<td>Returns highest value.</td>
</tr>
<tr>
<td>Minimum</td>
<td>Returns lowest value.</td>
</tr>
<tr>
<td>% of Column</td>
<td>Returns surface values as a percentage of their respective column item.</td>
</tr>
</tbody>
</table>
### Function and Description

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>% of Row</td>
<td>Returns surface values as a percentage of their respective row item.</td>
</tr>
<tr>
<td>% of Grand</td>
<td>Returns surface values as a percentage of all like values in the pivot table.</td>
</tr>
<tr>
<td>Increase</td>
<td>Returns the incremental difference between the final two instances of a total column or row. Apply only at the innermost dimensional level of a pivot table.</td>
</tr>
<tr>
<td>% Increase</td>
<td>Returns the percent difference between the final two instances of a total column or total row. Apply only at the innermost dimensional level of a pivot table.</td>
</tr>
<tr>
<td>Non-Null Average</td>
<td>Returns average of values; null values excluded.</td>
</tr>
<tr>
<td>Null Count</td>
<td>Returns number of null values.</td>
</tr>
<tr>
<td>Non-Null Count</td>
<td>Returns number of values; null values excluded.</td>
</tr>
</tbody>
</table>

**Note:** Null values are empty values for which no data exists. Null values are not equal to zero.

### Using True Computed Item Totals

In the Pivot Section, break totals can be recalculated to have their value equal to the sum of their displayed detail cells. “True computed item totals” use aggregation according to the specified data function and do not rely on the computed item total formula. You also have the option to use the break total cell results derived from the computed item formula applied to the detail cell.

The “Computed” column is defined by the following formula:

\[(\text{Units} \% 50) + 1\]

where “%” represents modulo (remainder) operator. In other words, the formula is defined as:

Integer remainder of \((\text{“Unit column cell value”} / 50) + 1\)

For the Unit column values for each city within a state, the formula works as expected. For example, in the Oakland, CA cell, the formula is:

Units 910

Modulo (remainder) of 910 / 50 = 10

Add 1 to assign a value of 11 (shown above).

For the California “Total” row, the value shown is 41, which is the result of the following formula:

Total “Units” for California = 12390

Modulo of 123900/50 = 40

Add 1 to assign a value of 41

The Modulo of 41 is not the sum of the displayed cell values for all cities in California, instead it is the modulo formula applied only to the cell containing the “Unit” column city total for California.
To see a break cell total value of 145, use the True Computed Item feature total, which would reference the displayed values in the detail cells (this example assumes a Sum data function):

\[ 46 + 1 + 11 + 1 + 46 \ (= 145) \]

To use true totals in a break total cell:

1. Select Pivot Options from the Pivot menu.
   The General tab of the Pivot Options dialog box is displayed.
2. Select True Computed Item Totals and click OK.

**Using Surface Values in Data Functions**

Data functions, when applied to total rows or columns, can either apply calculations to *surface values* (the values displayed in the pivot table) or *underlying values* (the values from the original Results section).

When applied to surface values, data functions recalculate the values in the visible cells or surface of the pivot table. When applied to underlying values, data functions return to the unaggregated values beneath the pivot table and recalculate based on those values. When underlying values are used, the results often are displayed incongruous with the aggregate surface values of the chart element. In other words, a total of the underlying values does not match the total of the surface figures.

Consider a simple pivot table with two values of 20 and 30. Each of these is already a total of underlying values:

- \( 20 = 8 + 12 \)
- \( 30 = 10 + 20 \)

An average of the underlying values yields the result of:

\[ 12.5 = \frac{(8+12+10+20)}{4} \]

An average of the surface values yields a result of:

\[ 25 = \frac{(20+30)}{2} \]

To match surface-level values in your calculation, you can instead apply surface values to the totals derived from data functions. For example, if you use surface values for an average applied to a total, the total is converted to the average of the surface values in the corresponding element.

To use surface values, select Pivot, then Use Surface Values.

**Using Weighted Averages**

Weighted averages are useful for a variety of purposes, such as survey research or when you want to include demographic information in your pivot tables. For example, assume you took a survey of 100 people, 75 male and 25 female. But according to census data in that geographic region you should have surveyed 50 males and 50 females. The data you have is skewed toward males.
To correct for this, you assign a weight or *weighting factor* to correct for the sampling error in your survey. To calculate a weight you take the expected amount and divide it by the actual amount.

In the example, the men would have a weighting factor of:

\[ 50 \div 75 = 0.6666 \]

The women would have a weight of:

\[ 50 \div 25 = 2 \]

Any calculation would calculate each man as 0.6666 and each woman as 2.

Weighted averages can also be used to apply different levels of importance to a given item. Take, for example, a survey, which has multiple questions. The responses can be rated on a scale of 1 to 5. By assigning a weight to each question based on the level of importance (the higher the number the more important), and using that weight in calculating a weighted average, you can arrive at averages that are more meaningful.

To use weighted averages, you must add a column of data to the database. This data indicates the relative weight of each corresponding value in another column. The statistical calculation for weighted averages depends on the following mathematical formula:

<table>
<thead>
<tr>
<th>Customer ID</th>
<th>Units Sold</th>
<th>Type of Store</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1435</td>
<td>80</td>
<td>Electronics</td>
<td>8</td>
</tr>
<tr>
<td>1539</td>
<td>200</td>
<td>Computer</td>
<td>10</td>
</tr>
<tr>
<td>1634</td>
<td>60</td>
<td>Electronics</td>
<td>8</td>
</tr>
<tr>
<td>1213</td>
<td>900</td>
<td>Discount</td>
<td>2</td>
</tr>
</tbody>
</table>

To use weighted averages:

1. Ensure that a column of data with the weighted values exists in the database.
2. In the Query section, select the Topic item for which weighted values are needed.
3. Select Query, then Data Functions, and then Weight.

   The Reference dialog box is displayed.

4. Select the item that contains the weighted values and click OK.

   The item in the Request line is renamed to indicate it is a weighted value.

5. Process the query.

   The weighted values are returned in the Results section.

6. Go to the Pivot section and drag the Weighted item from Elements to the data layout.

   You may now use all of the various data functions on the weighted values.

   **Note:** Weighting functions work only in the Pivot section.
Adding Computed Items

Use the Add Computed Items command to create new elements in the Pivot section. Computed items enable you to build equations or apply functions to data values. Computed items are like normal data items and can be included in pivot tables or reused to compute other data.

To add a computed item:

1. **Select Pivot, then Add Computed Item.**
   - The Computed Item dialog box is displayed.

2. **In the Name field, type a name that describes the computation.**
   - The default name is Computed, which is numbered sequentially if there is more than one. If you assign a name to a computed item that is identical to a scalar function name, Interactive Reporting numbers the name starting with the number 2.

3. **Define the new data item by building an expression in the Definition text box.**
   - Use the operator buttons to insert arithmetic and logical operators at the insertion point.
   - Click **Reference** to display the Reference dialog box, and select Request items to place in the equation.
   - Click **Functions** to apply scalar functions using the Functions dialog box.
   - You can also type any portion of the equation or the entire equation directly into the Definition text box using JavaScript. The names are case sensitive, and you must replace spaces in item names with underscores ("_").

4. **If necessary, click Options to set a new data type for the item.**

5. **When the equation is complete, click OK.**
   - The computed item is added to the data layout and it is displayed as a column in the pivot table.

Considering Null Fact Values in Pivot Computed Items

Moving function calculations require that all displayed Fact cells must be considered in any computations. To consider null fact values in the Pivot section, see Selecting Pivot Table Elements.

- For Simple and Weighted Moving Averages, if any values are missing/null in the source “Data Column”, the Moving Average functionality compensates by subtracting the number of instances of null values from the “Window” divisor. For example, consider the following example for the “Sales” value for “Feb” which had a value of “Sales” value for “Feb” which is null:

- **Use Surface Value**—Recalculates the values in the visible cells or surface of the pivot rather than the values in the Results section.

- **True Computed Item Totals**—Recalculates break totals so that each total value is equal to the sum of their displayed detail cells. True computed item totals use aggregation according to the specified data function and does not rely on the computed item total formula. If you disable this option, the break total cell values are derived from the computed item formula.
applied to the detail cell. Average and Count aggregation Data Functions are not evaluated in True Total mode unless you enable Use Surface Values. If you do not enable Use Surface Values, the Average and Count aggregation are calculated using the count of the underlying Table/Result section data values instead of the displayed Pivot values.

- **Enable Null Facts In Computed Items**—Evaluates a null fact value (an empty cell value) as a zero fact value for non-Moving functions. For Moving Function calculations, where the presence of all displayed Fact cell must be considered in calculations, the following behavior occurs when null fact values are considered as zero. The default option to evaluate and show null values is disabled.

- For Simple and Weighted Moving Averages, if any values are missing/null in the source “Data Column”, the Moving Average functionality compensates by subtracting the number of instances of null values from the “Window” divisor.

- For Moving Maximum, Moving Minimum, Moving Difference, Moving Sum and Moving Median functions moving functions, if any values are missing/null in the source “Data Column”, the missing source values are omitted and the calculation Window size “shrunk” correspondingly.

- Exponential Moving Averages treat missing source Fact values as having a value of zero.

- For all Moving Functions, except Exponential Moving Averages, if all the values in the particular calculation Window are null or zero, the Moving Function value is also, by default, null or zero.

- For all Moving Functions, except Exponential Moving Averages, if the Moving Function calculation Window size exceeds the available number of input fact values, the returned set of Moving Function values should be null.

- When this option is disabled, the Pivot sections that receive their data from a relational Query section, ignore Fact cells with null data when calculating Computed Items. That is, the calculation is simply skipped for the null cell.

If you want to evaluate and show null values for a new Pivot section, select this option before creating the Pivot section.

**Pivot Section Aggregation and Surface Values**

This section describes how the Pivot section aggregates and displays data from underlying Table or Results sections. It also discusses the influence of the "Surface Values" property on any final displayed results.

There are three key component parts of the Pivot section display including: Detail Fact cells, Break Total cells and Computed Items cells. The behavior of these component parts is discussed in various typical Pivot usage scenarios.
**Fact Detail Cells**

A Fact detail cell is the core numeric data that you slice and dice dimensionally in your analysis. A variety of components effect how Fact detail cells behave and what sort of information you see when analyzing them.

Review the following:

- **Results/Table Section and Pivot Section**
- **Aggregation Data Functions**
- **Break Totals**
- **Computed Items**
- **Surface Values**
- **Pivot Section Levels**

**Results/Table Section and Pivot Section**

Fact detail cells are derived from the data set that you retrieve from your database and can be analyzed in both the Results and Pivot section. The Results section is a flat representation of raw, un-aggregated data from a database or external data source as shown below. This example shows sample Sales data for several cities in the USA.

<table>
<thead>
<tr>
<th></th>
<th>City</th>
<th>Units Sold</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Los Angeles</td>
<td>500</td>
</tr>
<tr>
<td>2</td>
<td>Los Angeles</td>
<td>400</td>
</tr>
<tr>
<td>3</td>
<td>Los Angeles</td>
<td>100</td>
</tr>
<tr>
<td>4</td>
<td>Los Angeles</td>
<td>100</td>
</tr>
<tr>
<td>5</td>
<td>Miami</td>
<td>300</td>
</tr>
<tr>
<td>6</td>
<td>Miami</td>
<td>200</td>
</tr>
<tr>
<td>7</td>
<td>Miami</td>
<td>100</td>
</tr>
<tr>
<td>8</td>
<td>Seattle</td>
<td>250</td>
</tr>
<tr>
<td>9</td>
<td>Seattle</td>
<td>50</td>
</tr>
<tr>
<td>10</td>
<td>Tacoma</td>
<td>100</td>
</tr>
</tbody>
</table>

A Pivot section is a flexible tabular display of aggregated data in Interactive Reporting that receives its source of data from either a Results or Table section. As an example, the same Table section City/Units Sold data from above can be converted to a simple Pivot section as shown below.

In the Pivot section, the Units Sold data has been aggregated; that is, there is only one occurrence of each City name shown in the Pivot section Row Labels. The Units Sold data for each City has been summed from the entries in the Table section above to show a single value for each City - these are the Pivot section Detail Facts.
Aggregation Data Functions

The method by which the underlying Table section values are aggregated/collected into Pivot section Facts can be varied by a user. As a default, the aggregation method is to sum the data (as shown in the Pivot example above); that is, the underlying Fact values are added together and the sum displayed as the Pivot Detail Fact cell value. There are several other aggregation Data Functions that can be applied to a Pivot Detail cell.

An aggregation Data Function can be applied to all Pivot Fact cells across a row or to the Fact cells down a column.

Break Totals

A Break Total is a user-enabled Pivot section row that can display summarized Fact information at the change of a level of Row Label information.

For example, in the above Pivot section, to see the sum of Unit Sales for all Cities, add a "Sum" Break Total line at the end of the City Row Label level as shown below:

<table>
<thead>
<tr>
<th>City</th>
<th>Units Sold</th>
</tr>
</thead>
<tbody>
<tr>
<td>Los Angeles</td>
<td>1,100</td>
</tr>
<tr>
<td>Miami</td>
<td>600</td>
</tr>
<tr>
<td>Seattle</td>
<td>300</td>
</tr>
<tr>
<td>Tacoma</td>
<td>100</td>
</tr>
<tr>
<td>Total</td>
<td>2,100</td>
</tr>
</tbody>
</table>

If you intend to use Break Totals in your Pivot section, be sure to enable surface values instead of using the underlying table data. In this way, Break Totals reflects the aggregation of displayed detail data.

If you use the “Add Totals” menu option to add a Break Total, the Break Total preserves any data function applied to a Fact Column.

Computed Items

A computed item is a user-created column of Fact data in a Pivot section that performs a user-specified calculation using fact data. As an example, below, a Computed Item column has been created to show the "Units Sold" fact amounts multiplied by 2. Each computed item column cell performs its calculations on the data that is used to create the corresponding cells of the fact column on which it is based:
Surface Values

Surface Values enable you to aggregate and calculate Pivot section data based on data displayed in the Pivot section as opposed to data from the underlying Results or Table section.

Pivot Section Levels

Pivot sections can contain multiple levels of Row Label and/or Column Label aggregations. As an example, a State level could be added to the Row Labels of the Pivot section shown above as follows:

<table>
<thead>
<tr>
<th>State</th>
<th>Units Sold</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA</td>
<td>1,100</td>
</tr>
<tr>
<td>FL</td>
<td>600</td>
</tr>
<tr>
<td>WA</td>
<td>300</td>
</tr>
<tr>
<td></td>
<td>100</td>
</tr>
</tbody>
</table>

Pivot Fact Detail Cell Behavior

See below for information on pivot fact detail cell behavior.

- Simple Pivot Section With Detail Cell Values Only (No Break Totals)—Surface Values Disabled
- Simple Pivot Section With Detail Cell Values Only (No Break Totals)—Surface Values Enabled
- Regular Break Total Line—Surface Values Disabled
- Regular Break Total Line—Surface Values Enabled
- Explicitly Specifying an Aggregation Data Function Across an Entire Break Total Row

Simple Pivot Section With Detail Cell Values Only (No Break Totals)—Surface Values Disabled

The example below shows a simple Pivot section with detail cell values only (no break totals and surface values have been disabled). (Unless otherwise specified, this Table section is used as the source data for Pivot section examples shown throughout this appendix.)
Review the Table section below that shows values of "Units Sold" for some sample cities in the USA. The visible Fact values for each City in the Pivot section are equivalent to the sum of each City's individual constituent "Units Sold" values in the underlying Table section.

In other words the Row Label of Los Angeles and the Pivot-displayed "Units Sold" value of 1,100 have been evaluated by summing each "Units Sold" value for Los Angeles in the underlying Table section (i.e. 500 + 400 + 100 + 100 = 1,100).

In addition to the sum method for aggregation in the Pivot section, you can explicitly specify a number of other aggregation methods, or Data Functions, in which to collect and represent the underlying Table section data as Facts in the Pivot section. These Data Functions include:

- **Average**—The displayed Pivot Fact value is the average (mean) value of the Fact values for this Pivot Row Label in the underlying Table section (i.e. the sum of the corresponding values in the Table section divided by the number of occurrences (rows) of these values).

- **Count**—The displayed Pivot Fact value is the sum of the number of occurrences of Fact values for this Row Label in the underlying Table section.

- **Maximum**—The displayed Pivot Fact value is the maximum value of all Fact occurrences in the underlying Table section for this Row Label.
The displayed Pivot Fact value is the minimum value of all Fact occurrences in the underlying Table section for this Row Label.

The effect of applying the above different aggregation Data Functions on Pivot Fact columns can be illustrated in the following Pivot section:

<table>
<thead>
<tr>
<th>Month</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>15</td>
</tr>
<tr>
<td>3</td>
<td>17</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>5</td>
<td>22</td>
</tr>
<tr>
<td>6</td>
<td>20</td>
</tr>
<tr>
<td>7</td>
<td>25</td>
</tr>
<tr>
<td>8</td>
<td>27</td>
</tr>
<tr>
<td>9</td>
<td>30</td>
</tr>
<tr>
<td>10</td>
<td>35</td>
</tr>
<tr>
<td>11</td>
<td>37</td>
</tr>
<tr>
<td>12</td>
<td>40</td>
</tr>
</tbody>
</table>

The "Units Sold" Fact column is evaluated using a Sum Data Function as above, whereas the "Average Of Units Sold" column uses the Average aggregation Data Function.

That is, the "Average Of Units Sold" Fact value for Los Angeles is evaluated according to the following formula:

The Sum of "Units Sold" values for Los Angeles in the underlying Table section divided by the number of occurrences (rows) of "Units Sold" values for Los Angeles in the underlying Table section

or

\( \frac{(500 + 400 + 100 + 100)}{4} = \frac{1100}{4} = 275 \)

The "Count Of Units Sold" value for Los Angeles is 4, which is evaluated by using the Count aggregation Data Function. It is equal to the count or number of individual "Units Sold" row occurrences for Los Angeles in the underlying Table section

The "Max Of Units Sold" for Los Angeles is 500, which is evaluated using the Maximum aggregation Data Function. This function returns the maximum single "Units Sold" value from all "Units Sold" values for Los Angeles in the underlying Table section.

The "Min Of Units Sold" for Los Angeles is 100, which is evaluated using the Minimum aggregation Data Function. This returns the minimum single "Units Sold" value from all "Units Sold" values for Los Angeles in the underlying Table section.
Simple Pivot Section With Detail Cell Values Only (No Break Totals)—Surface Values Enabled

If you enable the "Surface Values" property (by selecting Use Surface Values from the Pivot Option dialog. “Selecting Pivot Table Elements” on page 185) the Pivot section display from above changes to:

<table>
<thead>
<tr>
<th>Month</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Jan</td>
</tr>
<tr>
<td>2</td>
<td>Feb</td>
</tr>
<tr>
<td>3</td>
<td>Mar</td>
</tr>
<tr>
<td>4</td>
<td>Apr</td>
</tr>
<tr>
<td>5</td>
<td>May</td>
</tr>
<tr>
<td>6</td>
<td>Jun</td>
</tr>
<tr>
<td>7</td>
<td>Jul</td>
</tr>
<tr>
<td>8</td>
<td>Aug</td>
</tr>
<tr>
<td>9</td>
<td>Sep</td>
</tr>
<tr>
<td>10</td>
<td>Oct</td>
</tr>
<tr>
<td>11</td>
<td>Nov</td>
</tr>
<tr>
<td>12</td>
<td>Dec</td>
</tr>
</tbody>
</table>

The Fact columns have now been recalculated to use what would be a displayed (i.e. surface) summed, aggregated value for each City as a source of input data (i.e. the values that are actually displayed in the "Units Sold" column).

As a result, the "Average Of Units Sold" for Seattle references the sum of underlying "Units Sold" values for Seattle (that is, the same value that is displayed for Seattle in the "Units Sold" column). This value is 300 and there is only one value occurrence displayed for Seattle. Consequently, the "Average Of Units Sold for Seattle" is 300 divided by 1 which is 300.

Likewise, the "Count Of Units Sold" column uses the displayed "Units Sold" value for each City as a basis for calculation. Since there is always only one displayed value per City, the "Count Of Units Sold" for each City always has a value of 1.

The "Maximum Of Units Sold" and "Minimum Of Units Sold" column cell values are exactly the same as their equivalent cells in the "Units Sold" column. Since there is only one cell value available per City, by definition that value is always the Maximum or Minimum value available. Since the lowest level of data displayed in a Pivot section is always aggregated from the underlying Table section values, it is recommended that you disable surface values when applying a sum or other data function.

Regular Break Total Line—Surface Values Disabled

At any time, a "regular" Break Total line can be added to a Pivot section by selecting "Add Totals" from the Row Labels right-click speed menu (or the Pivot menu in the Section toolbar).

When a Break Total row is added, it is labeled "Total" as shown below:
Notice that, the regular Break Total row performs aggregation of each Break Total cell value from the underlying Table section data by using the aggregation Data Function that has been specified for that Fact column.

For example, the "Average Of Units Sold" Fact column Break Total cell has a value of 2100. This value has been evaluated by using the Average Data Function (which is the Data Function that has been specified already for that column) on the underlying Table section data. The equation is:

Sum all "Units Sold" values for all Cities in the underlying Table section and divide by the total number of occurrences of those values.

or

\[
\frac{500 + 400 + 100 + 100 + 300 + 200 + 100 + 250 + 50 + 100}{10} = \frac{2100}{10} = 210
\]

Likewise, The "Count Of Units Sold" Break Total value refers to the underlying Table section values and returns the total number of occurrences of "Units Sold" values there for all Cities. This amount equals 10.

The "Max Of Units Sold" Break Total value refers to the underlying Table section and returns the single maximum "Units Sold" amount for all Cities. This amount equals 500.

The "Min Of Units Sold" Break Total value refers to the underlying Table section and returns the single minimum "Units Sold" amount for all Cities. This amount equals 50.

### Regular Break Total Line—Surface Values Enabled

If the Surface Value property is enabled, the previous Pivot section, with regular Break Total line, is displayed as:

<table>
<thead>
<tr>
<th></th>
<th>Units Sold</th>
<th>Average Of Units Sold</th>
<th>Count Of Units Sold</th>
<th>Max Of Units Sold</th>
<th>Min Of Units Sold</th>
</tr>
</thead>
<tbody>
<tr>
<td>Los Angeles</td>
<td>1,100</td>
<td>1,100</td>
<td>1</td>
<td>1,100</td>
<td>1,100</td>
</tr>
<tr>
<td>Miami</td>
<td>600</td>
<td>600</td>
<td>1</td>
<td>600</td>
<td>600</td>
</tr>
<tr>
<td>Seattle</td>
<td>300</td>
<td>300</td>
<td>1</td>
<td>300</td>
<td>300</td>
</tr>
<tr>
<td>Tacoma</td>
<td>100</td>
<td>100</td>
<td>1</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Total</td>
<td>2,100</td>
<td>525</td>
<td>4</td>
<td>1,100</td>
<td>100</td>
</tr>
</tbody>
</table>

All cell values for each column in the Break Total line have been updated to show the aggregation of all Fact values that are displayed in that column using the aggregation data function specified for that column.
The Break Total value for the “Units Sold” column is shown to be 2100 which is the sum (i.e. the aggregation data function used for that column) of all displayed Fact detail cell values in the column (this value, 2100, happens to be the same as the value shown when surface values were disabled and the underlying Table section values were used, because of the mathematical nature of a sum operation).

For the “Average Of Units Sold” column – the Average aggregation data function is used for the Fact detail cells. As a result, this same data function is used to evaluate the Break Total cell value. The value of 525 is evaluated to be the sum of the displayed fact detail cells (= 2,100 = 1,100 + 600 + 300 + 100) divided by the number of occurrences of displayed fact detail cell values (4).

The “Count Of Units Sold” column uses the Count data function. As a result the Break Total value uses the Count data function operating against the number of occurrences of displayed Fact detail cells. Since there are four fact detail cell values in this column, the Break Total value is 4.

The “Max Of Units Sold” column uses the Maximum data function. As a result the Break Total value uses the Maximum data function to evaluate the largest number within the displayed Fact detail cells. This number is 1,100 which is the value displayed in the Break Total cell for this column.

The “Min Of Units Sold” column uses the Minimum data function. As a result, the Break Total value uses the Minimum data function to evaluate the largest number within the displayed Fact detail cells. This number is 1,100 which is the value displayed in the Break Total cell for this column.

**Explicitly Specifying an Aggregation Data Function Across an Entire Break Total Row**

With a regular Break Total line, as described above, the aggregation Data Function that is applied to each Break Total cell is the aggregation Data Function that is already applied to each Fact column.

However, you can explicitly specify an aggregation data function to apply to all cells on a Break Total row. To do this, select the entire Break Total row (by pressing the Alt and mouse left-click buttons simultaneously) and select the appropriate data function from the mouse right-click speed menu or the Pivot menu.

As an example, the row entitled “Total – Average” below, is a regular Break Total line which has had the Average aggregation Data Function explicitly applied to all of its cells (the row entitled “Total” is the regular Break Total row from above):

<table>
<thead>
<tr>
<th></th>
<th>Units Sold</th>
<th>Average Of Units Sold</th>
<th>Count Of Units Sold</th>
<th>Max Of Units Sold</th>
<th>Min Of Units Sold</th>
</tr>
</thead>
<tbody>
<tr>
<td>Los Angeles</td>
<td>1,100</td>
<td>275</td>
<td>4</td>
<td>500</td>
<td>100</td>
</tr>
<tr>
<td>Miami</td>
<td>600</td>
<td>200</td>
<td>3</td>
<td>300</td>
<td>100</td>
</tr>
<tr>
<td>Seattle</td>
<td>300</td>
<td>150</td>
<td>2</td>
<td>250</td>
<td>50</td>
</tr>
<tr>
<td>Tacoma</td>
<td>100</td>
<td>100</td>
<td>1</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Total</td>
<td>2,100</td>
<td>210</td>
<td>10</td>
<td>500</td>
<td>50</td>
</tr>
<tr>
<td>Total - Average</td>
<td>210</td>
<td>210</td>
<td>210</td>
<td>210</td>
<td>210</td>
</tr>
</tbody>
</table>
The "Total - Average" calculates the average value of "Units Sold" from the underlying Table section for all cells, that is, the Total - Average is the aggregated sum of all underlying "Units Sold" values divided by the number of occurrences of "Units Sold" values.

You can see how this concept works across all cells on a Break Total row by reviewing the example below. This example Break Total rows named "Total Sum", "Total - Count", "Total - Max", "Total - Min" which have the Sum, Count, Maximum and Minimum Data Functions respectively forced across their rows:

<table>
<thead>
<tr>
<th></th>
<th>Units Sold</th>
<th>Average Of Units Sold</th>
<th>Count Of Units Sold</th>
<th>Max Of Units Sold</th>
<th>Min Of Units Sold</th>
</tr>
</thead>
<tbody>
<tr>
<td>Los Angeles</td>
<td>1,100</td>
<td>275</td>
<td>4</td>
<td>500</td>
<td>100</td>
</tr>
<tr>
<td>Miami</td>
<td>600</td>
<td>200</td>
<td>3</td>
<td>300</td>
<td>100</td>
</tr>
<tr>
<td>Seattle</td>
<td>300</td>
<td>150</td>
<td>2</td>
<td>250</td>
<td>50</td>
</tr>
<tr>
<td>Tacoma</td>
<td>100</td>
<td>100</td>
<td>1</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Total</td>
<td>2,100</td>
<td>210</td>
<td>10</td>
<td>500</td>
<td>50</td>
</tr>
<tr>
<td>Total - Sum</td>
<td>2,100</td>
<td>2,100</td>
<td>2,100</td>
<td>2,100</td>
<td>2,100</td>
</tr>
<tr>
<td>Total - Average</td>
<td>210</td>
<td>210</td>
<td>210</td>
<td>210</td>
<td>210</td>
</tr>
<tr>
<td>Total - Count</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Total - Max</td>
<td>500</td>
<td>500</td>
<td>500</td>
<td>500</td>
<td>500</td>
</tr>
<tr>
<td>Total - Min</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
</tr>
</tbody>
</table>

You can override any cell in a Break Total row which has a forced data function. To override the Break Total Data Function, select the Fact cell in that column and the required data function. As an example, the "Count Of Units Sold" column from above has been selected and the Count data function has been explicitly assigned to this column. The resulting Pivot display is nearly identical to the example above except the entire "Count Of Units Sold" column has its Fact Detail and Break Total cells evaluated by the Count data function as described above:

<table>
<thead>
<tr>
<th></th>
<th>Units Sold</th>
<th>Average Of Units Sold</th>
<th>Count Of Units Sold</th>
<th>Max Of Units Sold</th>
<th>Min Of Units Sold</th>
</tr>
</thead>
<tbody>
<tr>
<td>Los Angeles</td>
<td>1,100</td>
<td>275</td>
<td>4</td>
<td>500</td>
<td>100</td>
</tr>
<tr>
<td>Miami</td>
<td>600</td>
<td>200</td>
<td>3</td>
<td>300</td>
<td>100</td>
</tr>
<tr>
<td>Seattle</td>
<td>300</td>
<td>150</td>
<td>2</td>
<td>250</td>
<td>50</td>
</tr>
<tr>
<td>Tacoma</td>
<td>100</td>
<td>100</td>
<td>1</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Total</td>
<td>2,100</td>
<td>210</td>
<td>10</td>
<td>500</td>
<td>50</td>
</tr>
<tr>
<td>Total - Sum</td>
<td>2,100</td>
<td>2,100</td>
<td>2,100</td>
<td>2,100</td>
<td>2,100</td>
</tr>
<tr>
<td>Total - Average</td>
<td>210</td>
<td>210</td>
<td>210</td>
<td>210</td>
<td>210</td>
</tr>
<tr>
<td>Total - Count</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Total - Max</td>
<td>500</td>
<td>500</td>
<td>500</td>
<td>500</td>
<td>500</td>
</tr>
<tr>
<td>Total - Min</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
</tr>
</tbody>
</table>

**Pivot Section Computed Items**

Computed items, are Fact columns that are created by the user and which use table data (or displayed Pivot data if Surface Values are enabled) as a basis for calculation. Each Pivot computed item is represented as a new Fact column in the Pivot display and Interactive Reporting provides a multitude of built-in data manipulation and evaluation functions for the user that facilitate the construction of computed item columns.
Here are some key points to remember when working with computed items in the Pivot section:

- To ensure consistency, the source of data for computed items should be governed by the type of computed item function rather than being governed "globally" by the Surface Values property.
- Computed Item functions that operate on aggregated data such as "Avg" and "ColMax" should always receive their input data from the underlying Results or Table data.
- Computed Item functions that expect a single value as input, such as Sqrt, should receive their data from the displayed surface values.
- Computed Items that refer directly to other Pivot section columns without wrapping in a Function reference (such as "Units Sold * 2" above, should receive this data always from displayed surface values.

**Computed Item Detail Cell Values**

A very simple computed item can be created using a Label dimension and the values from the Fact Detail cells. As an example of computed item display, consider the original simple "Units Sold" Pivot display below:

<table>
<thead>
<tr>
<th></th>
<th>Units Sold</th>
</tr>
</thead>
<tbody>
<tr>
<td>Los Angeles</td>
<td>1,100</td>
</tr>
<tr>
<td>Miami</td>
<td>600</td>
</tr>
<tr>
<td>Seattle</td>
<td>300</td>
</tr>
<tr>
<td>Tacoma</td>
<td>100</td>
</tr>
</tbody>
</table>

Several computed item columns can be added to this Pivot using the "Units Sold" Facts as a source of input data. These computed items are shown below as new columns to the right of the "Units Sold" column (surface values have been disabled in this example):

<table>
<thead>
<tr>
<th></th>
<th>Units Sold</th>
<th>Units Sold * 2</th>
<th>Sqrt Units Sold</th>
<th>ColMax Units Sold</th>
<th>Average Units Sold</th>
</tr>
</thead>
<tbody>
<tr>
<td>Los Angeles</td>
<td>1,100</td>
<td>2,200</td>
<td>33.16625</td>
<td>500</td>
<td>210</td>
</tr>
<tr>
<td>Miami</td>
<td>600</td>
<td>1,200</td>
<td>24.4949</td>
<td>500</td>
<td>210</td>
</tr>
<tr>
<td>Seattle</td>
<td>300</td>
<td>600</td>
<td>17.32051</td>
<td>500</td>
<td>210</td>
</tr>
<tr>
<td>Tacoma</td>
<td>100</td>
<td>200</td>
<td>10</td>
<td>500</td>
<td>210</td>
</tr>
</tbody>
</table>

These new Computed Item columns are evaluated as follows:

- Units Sold * 2 = Units Sold values multiplied by 2 (i.e. computed item calculation: formula is: Units_Sold * 2)
- Sqrt Units Sold = Square root of Units Sold values (i.e. computed item calculation formula: Sqrt (Units_Sold))
- ColMax Units Sold = Maximum value of Units Sold in a column (i.e. computed item calculation formula: ColMax (Units_Sold))
- Average Units Sold = Average value of Units Sold (i.e. computed item calculation formula: Avg (Units_Sold))
Computed Item Detail Cell Values—Without Surface Values

In general, when surface values are disabled, computed item columns receive their source "Units Sold". Fact values from the underlying Table section. However, the specific values from the underlying Table section that the computed item uses depends on the nature of the its calculation.

A calculation or computed item function that expects a single value as input uses the underlying values that correspond to the Pivot computed item column cell to be calculated, aggregated according to the aggregation data function applied to the computed item column.

In the previous example, the "Units Sold * 2" column implicitly requires a single cell value. If it is used in a computed item column with an aggregation data function of Sum, all underlying Table section values corresponding to this Pivot cell are summed and multiplied by 2.

If you review the "Units * 2" value for Los Angeles, you see that it is evaluated by summing the underlying Table occurrences for Los Angeles and multiplying this sum by 2.

The ColMax computed item function operates on a range of data as opposed to a single value. The Avg computed item function operates on the summed, aggregated values corresponding to a Pivot cell. In the case of Los Angeles, this would be the sum of underlying values for Los Angeles divided by the number of occurrences of these values.

This behavior can be most easily seen from the "ColMax Units Sold" column which shows a value of 500 for each cell. The ColMax function interrogates the range of values in an entire column and returns the maximum value found: 500. This is the maximum cell value in the entire "Units Sold" column in the underlying Table section (compared to 1,100 which is the maximum "Units Sold" value shown in the Pivot display).

Computed Item Detail Values— With Surface Values Enabled

Compare the above Pivot section display with the following example. In this example, surface values have been enabled.

<table>
<thead>
<tr>
<th></th>
<th>Units Sold</th>
<th>Units Sold * 2</th>
<th>Sqrt Units Sold</th>
<th>ColMax Units Sold</th>
<th>Average Units Sold</th>
</tr>
</thead>
<tbody>
<tr>
<td>Los Angeles</td>
<td>1,100</td>
<td>2,200</td>
<td>33.16625</td>
<td>1,100</td>
<td>525</td>
</tr>
<tr>
<td>Miami</td>
<td>600</td>
<td>1,200</td>
<td>24.4949</td>
<td>1,100</td>
<td>525</td>
</tr>
<tr>
<td>Seattle</td>
<td>300</td>
<td>600</td>
<td>17.32051</td>
<td>1,100</td>
<td>525</td>
</tr>
<tr>
<td>Tacoma</td>
<td>100</td>
<td>200</td>
<td>10</td>
<td>1,100</td>
<td>525</td>
</tr>
</tbody>
</table>

As you can see, the "ColMax Units Sold" and "Average Units Sold" Fact columns values have changed. With surface values enabled, the ColMax function refers to the displayed "Units Sold" Fact column for input data and the highest value in this column is 1,100 which is the value that is displayed in the "ColMax Units Sold" column above.

The Avg Computed Item function which is used to evaluate the "Average Units Sold" column takes its input from the sum of the displayed "Units Sold" Fact column values and divides this by the number of displayed occurrences of values in the "Units Sold" column - this is 2100/4 which equals 525 - the value displayed in the "Average Units Sold" column.
In general when you enable surface values for Pivot section computed items, they operate on the summed, aggregated values from the underlying Table section.

**Computed Items With Break Totals—No Surface Values**

If a Break Total row is added to the Pivot by using the "Add Totals" menu item described above, the following Pivot display results:

<table>
<thead>
<tr>
<th></th>
<th>Units Sold</th>
<th>Units Sold * 2</th>
<th>Sqrt Units Sold</th>
<th>ColMax Units Sold</th>
<th>Average Units Sold</th>
</tr>
</thead>
<tbody>
<tr>
<td>Los Angeles</td>
<td>1,100</td>
<td>2,200</td>
<td>33.16625</td>
<td>500</td>
<td>210</td>
</tr>
<tr>
<td>Miami</td>
<td>600</td>
<td>1,200</td>
<td>24.4949</td>
<td>500</td>
<td>210</td>
</tr>
<tr>
<td>Seattle</td>
<td>300</td>
<td>600</td>
<td>17.32051</td>
<td>500</td>
<td>210</td>
</tr>
<tr>
<td>Tacoma</td>
<td>100</td>
<td>200</td>
<td>10</td>
<td>500</td>
<td>210</td>
</tr>
<tr>
<td>Total</td>
<td>2,100</td>
<td>4,200</td>
<td>45.82576</td>
<td>500</td>
<td>210</td>
</tr>
</tbody>
</table>

Since the surface values have been disabled, the values in the Break Total ("Total") cells for each Computed Item column are evaluated by applying the computed item calculation formula to the underlying aggregated Table section "Units Sold" values summed for all Cities.

**Computed Items With Break Totals—Surface Values Enabled**

If a Break Total row is added to the Pivot by using the "Add Totals" menu and surface values are enabled, the following Pivot display results:

<table>
<thead>
<tr>
<th></th>
<th>Units Sold</th>
<th>Units Sold * 2</th>
<th>Sqrt Units Sold</th>
<th>ColMax Units Sold</th>
<th>Average Units Sold</th>
</tr>
</thead>
<tbody>
<tr>
<td>Los Angeles</td>
<td>1,100</td>
<td>2,200</td>
<td>33.16625</td>
<td>1,100</td>
<td>525</td>
</tr>
<tr>
<td>Miami</td>
<td>600</td>
<td>1,200</td>
<td>24.4949</td>
<td>1,100</td>
<td>525</td>
</tr>
<tr>
<td>Seattle</td>
<td>300</td>
<td>600</td>
<td>17.32051</td>
<td>1,100</td>
<td>525</td>
</tr>
<tr>
<td>Tacoma</td>
<td>100</td>
<td>200</td>
<td>10</td>
<td>1,100</td>
<td>525</td>
</tr>
<tr>
<td>Total</td>
<td>2,100</td>
<td>4,200</td>
<td>45.82576</td>
<td>1,100</td>
<td>525</td>
</tr>
</tbody>
</table>

With surface values enabled, the various Break Total cell values are evaluated using the displayed values of the "Units Sold" Fact column. For example, the "ColMax Units Sold" total line value has changed (from 500) to 1,100 since the maximum value displayed in the "Units Sold" column is 1,100. The "Average Units Sold" total line value has changed (from 210) to 525 since the displayed sum of "Units Sold" is 2,100 and the displayed number of Cities is 4 - leading to an average calculation of 2,100/4 = 525.

In either case, whether surface values are enabled or disabled, the computed item Break Total values are calculated by applying the computed item calculation on the Break Total value for the column on which the computed item calculation is based.

There exists no facility whereby computed item Break Totals are evaluated by summing (or some other aggregate data function) its constituent displayed Detail cell values. If such a facility were to exist, the "Total" cell for the "ColMax Units Sold" column above would be 4,400 (= 1,100 + 1,100 + 1,100 + 1,100 + 1,100) i.e. the sum of all Detail Fact values in this column.
# Pivot Menu Command Reference

The following table provides a quick reference to the commands available on the Pivot menu and lists any related shortcuts.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Keyboard Shortcut</th>
<th>Shortcut Menu</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add Selected Items</td>
<td>Adds the selected item as a Column Label, Row Label, or Fact.</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Remove Selected Items</td>
<td>Removes the selected item.</td>
<td>Del</td>
<td>*</td>
</tr>
<tr>
<td>Modify</td>
<td>Modifies the selected computed item.</td>
<td>Ctrl+M</td>
<td>*</td>
</tr>
<tr>
<td>Sort</td>
<td>Reorders the selected item by labels, by values, ascending, or descending.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add Totals</td>
<td>Adds the selected item to the Measures pane.</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Add Computed Item</td>
<td>Adds a new data item derived from calculations performed on an item.</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Add Cume</td>
<td>Adds cumulative totals to break totals by dimension and restarts them at each dimensional grouping.</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Data Function</td>
<td>Applies a prebuilt data function to the selected item.</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Drill Anywhere</td>
<td>Enables you to drill to any item.</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Drillup</td>
<td>Returns the original view of data that you drilled.</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Focus On Items</td>
<td>Updates the pivot table to include only the selected data.</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Hide Items</td>
<td>Hides the selected item.</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Show Hidden Items</td>
<td>Restores the selected hidden item.</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Show All Items</td>
<td>Updates the pivot table to include all items.</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Group Items</td>
<td>Groups the selected dimensions.</td>
<td>Ctrl+G</td>
<td></td>
</tr>
<tr>
<td>Ungroup Items</td>
<td>Ungroups the selected dimension.</td>
<td>Ctrl+U</td>
<td></td>
</tr>
<tr>
<td>Restore Name</td>
<td>Restores the original name of a renamed item.</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Refresh Data</td>
<td>Updates the data according to the selected option. Select between After Process, When Section Displayed, Manually, or Refresh Now.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pivot Options</td>
<td>Enable surface values, true computed totals, and null facts in computed items.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# Charting Data

In This Chapter

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chart Section</td>
<td>210</td>
</tr>
<tr>
<td>Charting Basics</td>
<td>210</td>
</tr>
<tr>
<td>Chart Terminology</td>
<td>210</td>
</tr>
<tr>
<td>Understanding Chart Dimensions</td>
<td>212</td>
</tr>
<tr>
<td>Creating Charts</td>
<td>213</td>
</tr>
<tr>
<td>Selecting a Chart Type</td>
<td>213</td>
</tr>
<tr>
<td>Two-dimensional Charts</td>
<td>214</td>
</tr>
<tr>
<td>Multidimensional Charts</td>
<td>221</td>
</tr>
<tr>
<td>Reference Lines, Baselines, and Trend Lines</td>
<td>229</td>
</tr>
<tr>
<td>Manipulating Chart Data</td>
<td>238</td>
</tr>
<tr>
<td>Changing Angle and Elevation (Rotating)</td>
<td>244</td>
</tr>
<tr>
<td>Working with Chart Elements</td>
<td>245</td>
</tr>
<tr>
<td>Chart Scrolling and Scaling</td>
<td>249</td>
</tr>
<tr>
<td>Customizing Chart Properties</td>
<td>253</td>
</tr>
<tr>
<td>Chart Menu Command Reference</td>
<td>260</td>
</tr>
<tr>
<td>General Chart Properties</td>
<td>261</td>
</tr>
<tr>
<td>Chart Labels Axis Properties</td>
<td>263</td>
</tr>
<tr>
<td>Chart Values Axis Properties</td>
<td>265</td>
</tr>
<tr>
<td>Bar Chart Properties</td>
<td>266</td>
</tr>
<tr>
<td>Area and Ribbon Chart Patterns Properties</td>
<td>267</td>
</tr>
<tr>
<td>Bar Chart Patterns Properties</td>
<td>267</td>
</tr>
<tr>
<td>Bar Chart Data Labels Properties</td>
<td>268</td>
</tr>
<tr>
<td>Line Chart Patterns Properties</td>
<td>268</td>
</tr>
<tr>
<td>Line Chart Axis Properties</td>
<td>269</td>
</tr>
<tr>
<td>Line Chart Data Labels Properties</td>
<td>269</td>
</tr>
<tr>
<td>Line Chart Properties</td>
<td>269</td>
</tr>
<tr>
<td>Pie Chart Patterns Properties</td>
<td>270</td>
</tr>
<tr>
<td>Pie Chart Data Labels Properties</td>
<td>270</td>
</tr>
<tr>
<td>Bubble Chart Properties</td>
<td>270</td>
</tr>
<tr>
<td>Bubble Chart Patterns Properties</td>
<td>271</td>
</tr>
<tr>
<td>Scatter Chart Patterns Properties</td>
<td>271</td>
</tr>
<tr>
<td>Set Legend On</td>
<td>272</td>
</tr>
<tr>
<td>Plot Area</td>
<td>272</td>
</tr>
</tbody>
</table>
Chart Section

The Chart section enables you to see meaningful summaries of your data. Graphic snapshots help you recognize patterns, trends, and other relationships that might not be apparent in columns and rows of tabular data.

The Chart section opens with an initial plot area for the chart. Because chart construction and manipulation is managed with the data layout, plotting, viewing, and reviewing are easy and intuitive. Charts respond dynamically to your commands. When you make a change in a charted item, you see your chart instantly redrawn to reflect the change. Experimenting with different combinations of data can be surprisingly informative.

Charting Basics

A chart is a graphic representation of data. Except for pie charts, all charts plot data with reference to a horizontal x axis and vertical y axis. Multidimensional charts sometimes plot data on an additional axis. A pie chart uses the metaphor of the pie as a whole to delineate the relative values of the parts or slices.

You construct a chart by dragging Request items from Elements to one of the data layouts. At least one y axis item must populate an data layout to plot a usable chart.

Chart Terminology

The following table defines the chart terminology used in Interactive Reporting:

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Axes</td>
<td>Straight lines on a chart that provide a framework for measurement and reference. Typically, the x axis and z axis are used to display label items, and the y axis shows values or facts (measurable items), such as units and amounts.</td>
</tr>
<tr>
<td>Values</td>
<td>Graphic indicators that represent data. Bar charts display values in either vertical or horizontal bars. Pie charts use wedge-shaped slices to represent values.</td>
</tr>
<tr>
<td>Plot Area</td>
<td>The area bounded by the axes is called the plot area. In the case of the pie chart, the plot area is defined by a circle representing the totality of all data items.</td>
</tr>
<tr>
<td>Term</td>
<td>Definition</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Planes</td>
<td>In all charts (except pie charts), planes provide background and graphed reference for charted values. Planes define horizontal, vertical, and background fields for a chart.</td>
</tr>
<tr>
<td>Legend</td>
<td>An information box containing color-keyed labels used to identify different data values represented on a chart.</td>
</tr>
<tr>
<td>Grid lines</td>
<td>Straight horizontal and vertical lines arranged in scaled increments that provide calibrated guidelines for value interpretation.</td>
</tr>
<tr>
<td>Bar</td>
<td>A linear measure of a value used in bar charts.</td>
</tr>
<tr>
<td>Slice</td>
<td>A spacial measure of a value used in pie charts.</td>
</tr>
<tr>
<td>Chart top title</td>
<td>Text description related to the entire chart.</td>
</tr>
<tr>
<td>Chart subtitle</td>
<td>Additional text used to describe the chart.</td>
</tr>
<tr>
<td>Y axis label</td>
<td>Text description of the vertical quantity axis.</td>
</tr>
<tr>
<td>Y axis values</td>
<td>Text or numbers which indicate specific values along the y axis</td>
</tr>
<tr>
<td>X axis label</td>
<td>Text description of the horizontal quantity axis.</td>
</tr>
<tr>
<td>X axis values</td>
<td>Text or numbers which indicate specific values along the x axis</td>
</tr>
<tr>
<td>Z axis label</td>
<td>Text description of the depth axis.</td>
</tr>
<tr>
<td>Z axis values</td>
<td>Text or number which indicate specific values along the z axis</td>
</tr>
<tr>
<td>Chart graphic</td>
<td>Central chart picture which contains Chart bars, background and also Chart axes.</td>
</tr>
<tr>
<td>Chart graphic border</td>
<td>Border which controls the position of the Chart's central bars, background and axes labels. The border can be resized and repositioned.</td>
</tr>
<tr>
<td>Chart border</td>
<td>Border which controls the general position of all other Chart components not controlled by the Chart graphic border. This border can be resized and repositioned.</td>
</tr>
<tr>
<td>Inserted text string</td>
<td>Optional text strings that can be placed anywhere within the Chart border.</td>
</tr>
<tr>
<td>Right y axis label</td>
<td>Optional label that can be placed to the right of the right-hand vertical edger of the Chart graphic.</td>
</tr>
<tr>
<td>Right y axis values</td>
<td>Optional values that can be displayed on the right hand vertical edge of the Chart graphic</td>
</tr>
<tr>
<td>Bar value</td>
<td>Optional text which represents numeric value of the bar.</td>
</tr>
</tbody>
</table>

Table 54  Chart Terminology (Pie Specific)

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pie slice</td>
<td>Individual y axis value representation for current x axis item.</td>
</tr>
<tr>
<td>Pie slice label</td>
<td>Text to identify x axis item.</td>
</tr>
<tr>
<td>Line to label</td>
<td>Line from pie slice label to pie slice to aid in slice identification.</td>
</tr>
</tbody>
</table>
Understanding Chart Dimensions

To understand the differences among charts, you have to distinguish between dimensions in space and dimensions of data. The two dimensions are distinct. Space can be represented as three dimensions along the x, y, and z axes. Corresponding to the X, Y and Z axes in the Chart Area. Data can either be represented in a two-dimensional or a three-dimensional (2-D or 3-D) space. In two dimensions, data is represented along the x axis and y-axis only. In three dimensions, data is projected back along the z axis also. Two dimensions of data must be represented in 2-D space. At least three dimensions of data are necessary to use the third spatial dimension Stack, Cluster or Depth (z axis). But three or more dimensions of data can be represented in 2-D space. For example, cluster and stack represent data categories in two spatial dimensions (X axis and Y axis only).

Using the Chart Data Layout

The Chart section opens with an initial plot area for the chart. Because you manage chart construction and manipulation is with the Chart data layout plotting, viewing and reviewing are easy and intuitive.

You construct a chart by dragging items from Elements to a data layout pane. At least one items must populate the data layout to plot a usable chart.

The data layout consists of the following items:

<table>
<thead>
<tr>
<th>Data Layout Pane</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X Axis Slice</td>
<td>The X-Axis is used for those items to be place don the x axis, which is a straight line on the chart. It is used as a qualitative data label for categorizing information. To place items on the x axis, use the X-Axis pane.</td>
</tr>
</tbody>
</table>
| Stack Cluster Depth | The Stack Cluster or Depth pane represent the third dimension of data, that is, the z axis that projects out toward you or a location in space. This axis can either be qualitative or quantitative. 
For a Stack each dimension of data of the charts is represented by only one bar. This bar consists of as many components as the data file has data rows, with the data from each row stacked onto the previous row. For example, a single bar can represent the amount of sales for CD-ROM drives in one year on top of a bar representing sales for other years. You can stack the bar charts vertically or horizontally. By stacking items and assigning a different color to each item, you can display trends among comparable or related items, or emphasize visually a sum of several indicators. 
For a Cluster, data extended in the third dimension is shown as clusters displayed in the foreground. This category creates a vertical column (and only a vertical column) for each data value. If the chart is showing multiple data series, the values are grouped based upon the category value. For example, use clustered bars to compare stores of different types. Alternatively, cluster bars can be used to compare two different values items, such as Amount of Sales and Units Sold. 
For the Depth, data extends the length of the chart along the z-axis. |
<table>
<thead>
<tr>
<th>Data Layout Pane</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fact</td>
<td>The Facts category indicates height in the coordinate system. It is used as a quantitative label as a way of categorizing information on the y axis.</td>
</tr>
<tr>
<td>Fact (Stack)</td>
<td>For the Fact (Stack) pane each dimension of numeric data is represented by only one bar, and shows the grouping along the y axis. This bar consists of as many components as the data file has numeric rows, with the numeric data from each row stacked onto the previous row. For example, a single bar can represent the amount of sales for CD-ROM drives in one year on top of a bar representing sales for other years. You can stack the bar charts vertically or horizontally. By stacking items and assigning a different color to each item, you can display trends among comparable or related items, or emphasize visually a sum of several indicators.</td>
</tr>
<tr>
<td>Fact (Depth)</td>
<td>For the Fact (Depth) pane, numeric data extends the length of the chart along the z-axis.</td>
</tr>
</tbody>
</table>

## Creating Charts

Interactive Reporting charts consists of two layout elements: graphical elements (for example, chart bars or pie slices) and axis labels. When you assign Request items to the data layout, they become values or labels in your chart.

The instructions below are a starting point for building charts. As you use and become familiar with the Chart section, you learn ways to create the exact type of chart that fits your needs.

> To create a chart using the Chart data layout:

1. Select Insert, then New Chart.
2. If the data layout is not visible, click Data Layout on the Section title bar.
3. Drag each line item from Elements to the data layout:
   - Drag values or facts (such as Units or Amount) into the y pane in the data layout to create bar charts, pie charts or ribbon charts.
   - Drag a label item (such as Name, Product, or State) into the x pane in the data layout to create a 2-D chart.
   - Drag a label item to the y pane in data layout to add a third dimension to your chart.
4. Select Format, then Chart Type, and then the [chart type].

**Note:** You can select and drag multiple items to the same data layout to add multiple values or labels for analysis. Items are hierarchically ordered in the sequence in which they are displayed in the data layout.

## Selecting a Chart Type

You can select a different chart type to show data in different views.
To select a chart type, select **Format**, then **Chart Type** and choose a chart type from the submenu.

Chart types include:

- Vertical Bar Charts
- Horizontal Bar Charts
- Stacked Bar Charts
- Clustered Bar Charts
- Pie Charts
- Area Charts
- Line Charts
- Ribbon Charts
- Combination (Bar/Line) Charts
- Scatter Charts
- Bubble Charts

**Two-dimensional Charts**

Pie and bar charts (of the non-stacked variety) lend themselves well to representing two dimensions of data. For example, imagine charting the amount of sales by product type. In pie charts, the two dimensions are represented by slices of a pie. In bar charts, the data is represented by bars along the x axis and y axis.

- “Pie Charts” on page 214
- “Scatter Charts” on page 216
- “Bubble Charts” on page 218
- “Two-dimensional Bar Charts” on page 220

**Pie Charts**

Of all charts, the pie chart is the easiest to understand. Pieces (slices) of the pie are drawn to represent the relative value of a measurable item category to the whole. Pie charts represent additional dimensions of data by further subdividing the pie.

In a Pie chart, Request items placed in the x axis represent itemized slices of the pie. Request items placed in the y pane of the data layout define the quantitative whole of the pie.
Creating a Pie Chart

To create a pie chart:

1. From the Chart drop-down list box, select Pie.
2. Drag a value from Elements to the y pane in the data layout.
   A Pie chart without slices is displayed.
3. Drag one or more label items from Elements to a data layout pane.
   The Pie chart is differentiated to reflect subcategories. A legend depicting details of the
   selection is displayed.

Note: Since pie charts plot data using only two axes, the z pane is disabled in the Data Layout
when creating a Pie chart.

Positioning Pie Slices

You can pull individual pie slices out of the pie chart.

To toggle the position of a pie slice, select a slice of the pie and choose Pull Out Slice on the
shortcut menu.

A check mark is displayed on the shortcut menu next to Pull Out Slice to indicate that this feature
is active. Select this option again to clear the check mark and restore the pie slice to its original
position.

Showing Positive and Negative Values

Pie slices show positive values by default.

To toggle the display of negative and positive values, select a slice of the pie and select Show
Negative Values on the shortcut menu.

A check mark is displayed next to Show Negative Values to indicate that negative values are
shown. Select this option again to clear the check mark and show positive values.

Showing Pie Percentages

To toggle the display of each pie slice value as a percentage, select a slice of the pie and select Show Pie Percentages on the shortcut menu.

A check mark is displayed next to Show Pie Percentages to indicate that percentages are displayed
on the chart (in parentheses next to the pie chart label). Select this option again to clear the check
mark and remove the percentages from view.
Adding Lines to Labels

To add a pointer line between an individual pie slice and its label, select the pie slice and choose Format, then Line To Label.

Note: You cannot change a Chart label name when a Pie Chart contains 0 or null values. To add a label name in this case, filter out or hide the 0 or null values in the underlying Results or Pivot section.

Rotating Pie Charts

To rotate the perspective angle or elevation of a pie chart:

1. Click Rotate on the shortcut menu.
   The rotate icon is displayed on the pie chart.
2. Click the rotate icon and move the dotted line to a new location.
   When you release the mouse button, the chart is redrawn to reflect the adjusted perspective. If you cannot drag the rotate icon in a certain direction, the chart has reached its farthest possible rotation in that direction.

Showing the Pie Outline

To show a border around the entire border of the pie, select Show Pie Outliner on the shortcut menu.

Scatter Charts

A scatter chart is useful for emphasizing scientific or statistical similarities rather than differences in your analysis. Scatter charts illustrate the relationship between pairs of numerical or quantitative values, which are combined into individual data points along the horizontal (y axis) and a vertical (x axis) axis. Data points are plotted in uneven intervals.

A scatter chart represents non-aggregated sources, that is, it retrieves data from the underlying Table/Reports section and does not reflect rolled up values (all other chart types retrieve their data from an aggregated source, and there is a one-to-one correspondence between Charts and Pivots). For this reason, the Pivot This Chart feature is not available for a scatter chart.

To demonstrate how a scatter chart plots data, see the following two figures. If two data series consisting of tons per acre by degree of latitude are shown in an Excel worksheet, the data is displayed as:
If the two data series are rendered in a chart, the data is shown as:

<table>
<thead>
<tr>
<th>A</th>
<th>B per acre 95</th>
<th>C</th>
<th>D per acre 96</th>
</tr>
</thead>
<tbody>
<tr>
<td>Degree of latitude</td>
<td>4750</td>
<td>Degree of latitude</td>
<td>5000</td>
</tr>
<tr>
<td>31</td>
<td>4750</td>
<td>31</td>
<td>5000</td>
</tr>
<tr>
<td>32</td>
<td>4800</td>
<td>32</td>
<td>5200</td>
</tr>
<tr>
<td>33</td>
<td>4900</td>
<td>33</td>
<td>5400</td>
</tr>
<tr>
<td>34</td>
<td>4850</td>
<td>34</td>
<td>5500</td>
</tr>
<tr>
<td>35</td>
<td>4800</td>
<td>35</td>
<td>5150</td>
</tr>
<tr>
<td>36</td>
<td>4900</td>
<td>36</td>
<td>5650</td>
</tr>
<tr>
<td>37</td>
<td>4450</td>
<td>37</td>
<td>5100</td>
</tr>
<tr>
<td>38</td>
<td>4250</td>
<td>38</td>
<td>4950</td>
</tr>
</tbody>
</table>

Scatter charts can only contain a pair of fact or numeric values which are placed in the Y Axis and X Axis in the data layout. If you add only one fact item to the data layout, no scatter chart is rendered. In addition, label values cannot be added to the Y Axis or X Axis of the data layout.

The zoom feature is available from the View menu when you need to zoom in a selected area of the plotted scatter chart.

The following feature limitations apply to scatter charts:

- Data functions are not available to scatter charts because this type of chart relies on non-aggregated data.
- The sort feature cannot be used for scatter chart items.
- The Pivot To Chart feature is not available.
- Drilling cannot be performed on a scatter chart.
- The focus feature cannot be used on scatter chart items.
The Hide feature hides the whole data series in a scatter chart, and an individual item cannot be hidden.

To create a scatter chart:
1. In the Section pane, select the Chart in which to generate the scatter chart.
2. Select Scatter from the Chart Type drop down list.
3. Drag a fact value from Elements to the Y Axis of the data layout.
   A pair of fact values must be added to the data layout to create a scatter chart. Multiple pairs of facts can be added. If only one fact value is added to the data layout, no chart is rendered.
4. Drag a fact value from Elements to the X Axis of the data layout.

To show the real value of a data point, use the tool tip and hover over the data point.

To show grid lines on the scatter chart, click the plot area of the chart and select Show X Axis Grid Lines or Show Y Axis Grid lines.

To zoom in on data value, click and hold your mouse button and drag to draw a selection rectangle around the data values that you want to zoom in.

You can also select View, then Zoom, and then Zoom In.

To zoom out on data values, click View, then Zoom, and then Zoom Out.

Once the view is zoomed out, you can pan areas to the left, top, right, and bottom of the chart. If the view is zoomed out to the maximum, no shortcut menu is available.

**Bubble Charts**

Bubble charts are typically used to show three dimensions of data in a two dimensional chart. This type of chart often lends itself to the display of financial data because specific values can be visually represented in your chart by different bubble sizes. It is similar to scatter chart allowing you to plot data as a collection of bubbles. Bubble charts plot three values:

- a value set on the x-axis
- a value set on the y-axis
- a value that defines the size or width dimension of a bubble in proportion to the amount of data

For example, the worksheet in the following picture contains values for three types of facts: number of products, dollar value of sales, and percentage size of market share.
In the following bubble chart, the size of the bubble corresponds to the values in the third column of the sample data (Market share %):

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of products</td>
<td>Sales</td>
<td>Market Share %</td>
</tr>
<tr>
<td>1</td>
<td>14</td>
<td>$12,200.00</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>$60,000.00</td>
</tr>
<tr>
<td>3</td>
<td>18</td>
<td>$24,400.00</td>
</tr>
</tbody>
</table>

Multiple data values can be plotted in the bubble chart.

Bubbles with zero size can render using some small bubble size to prevent them from disappearing. Also there is an option can be provided to hide zero-size values if necessary.

Bubbles with negative values can also be displayed. These type of values are derived from their real absolute value, and the real negative value is depicted in the data label (although based on the positive value). You can optionally select not to show negative values.

The zoom feature is available from the View menu when you need to zoom in a selected area of the plotted bubble chart.

The following feature limitations apply to bubble charts:

- Data functions are not available to bubble charts because this type of chart relies on non-aggregated data.
- The sort feature cannot be used for bubble chart items.
- The Pivot To Chart feature is not available.
Drilling cannot be performed on a bubble chart.

The focus feature cannot be used on bubble chart items.

The Hide feature hides the whole data series in a bubble chart, and an individual item cannot be hidden.

To create a bubble chart:

1. In the Section pane, select the chart in which to generate the bubble chart.
2. Drag a fact value from Elements to the Y Axis of the data layout.
3. Drag a fact value from Elements to the X Axis of the data layout.
4. Drag a fact value from Elements to the Size pane of the data layout.

To show the real value of a data point, use the tool tip and hover over the data point.

To show grid lines on the scatter chart, click the plot area of the chart and select Show X Axis Grid Lines or Show Y Axis Grid lines.

To zoom in on data value, click and hold your mouse button and drag to draw a selection rectangle around the data values that you want to zoom in

Once zoomed in, the text: “Partial View” is displayed in the upper left of the chart.

To zoom out on data values, click Full or Previous on the shortcut menu.

Once the view is zoomed out, you can pan areas to the left, top, right, and bottom of the chart. If the view is zoomed out to the maximum, no shortcut menu is available.

Two-dimensional Bar Charts

Bar charts are the most common type of business chart and are especially useful for comparative analysis when you want to focus on comparing values and place less emphasis on time. Use a bar chart to illustrate comparisons among individual items.

Two-dimensional bar charts are plotted using a single item in each of the x pane and y pane in the data layout. The z pane is not populated in 2-D bar charts.

To create a two-dimensional bar chart:

1. Select a bar chart format from the Chart drop-down list box.
   The default chart format is Vertical Bar.
2. Drag a label item from Elements to the data layout.
   Data labels are displayed on the horizontal axis in the Chart area.
3. Drag a value from Elements to the y pane of the data layout.
A chart is plotted that summarizes the selected value on the y axis as it relates to the subcategories of the label item x axis.

Interactive Reporting automatically scales the data represented on the y axis and adds appropriate labels. The Legend provides an index of label information with a coordinated color scheme.

To use a different 2-D chart format, select another 2-D chart from the Chart drop-down list box.

➤ To display bar values:

1. Select Show Bar Values on the shortcut menu.

2. Select the placement of values on the bar:
   - Inside
   - Over
   - Outside
   - None

➤ To show the gap between bars, click a bar and select Show Bar Gap on the shortcut menu.

➤ To show the borders of a bar, click a bar and select Show Bar Border on the shortcut menu.

Related Topics
Three-dimensional Bar Charts
Clustered Bar Charts
Stacked Bar Charts

Multidimensional Charts
Frequently, you want to represent more than two dimensions of data at a time. For example, you may want to see how the sales of product types break down by years or quarter. There are numerous ways to chart three or more dimensions of data. You can project data into the third dimension of space. You can also represent the data in two spatial dimensions.

- “About the 3-D View” on page 222
- “Three-dimensional Bar Charts” on page 222
- “Clustered Bar Charts” on page 223
- “Stacked Bar Charts” on page 224
- “Area Charts” on page 224
- “Ribbon Charts” on page 225
- “Line Charts” on page 225
About the 3-D View

By default, Interactive Reporting imparts a 3-D look to your chart objects. These objects are displayed in the chart space as 3-D objects with depth. That does not mean that you are plotting three dimensions of data or using three dimensions of space to represent data. It is simply a visual effect that can be turned off.

**Note:** If you turn off 3-D View, you cannot view charts that use a third dimension in space.

To toggle 3-D View, select **Format**, then **3-D View**.

A check mark is displayed next to the 3-D View option to indicate it is active. Select this option again to clear the check mark and turn off 3-D view.

**Note:** You can also select to view objects in 3-D using the Properties dialog box. For more information, see “Customizing Chart Properties” on page 253.

Three-dimensional Bar Charts

You can add more information to your bar chart by adding an additional item or items to the z pane in the data layout. Using multidimensional charts, you can show various relationships between three or more items in easy-to-understand bar chart formats.

Interactive Reporting plots the added data in rows that extend back along the z axis of the chart. Dragging a label item to the Stack, Cluster or Depth category (in this case, Year) plots an extra dimension.

To create a three-dimensional bar chart:

1. **Select a bar chart format from the Chart drop-down list box.**
   - The default chart format is Vertical Bar.
2. **Drag a label item from Elements to the x pane in the data layout.**
3. **Drag a label item from Elements to the z pane in the data layout.**
4. **Drag a value from Elements to the y pane in the data layout.**

A chart is plotted that summarizes the selected value y pane as it relates to the subcategories of the label items (x axis and z axis).
To display line values, select **Show Line Values** on the shortcut menu.

To display bar values:
1. Select **Show Bar Values** on the shortcut menu.
2. Select the placement of values on the bar:
   - Inside
   - Over
   - Outside
   - None

To show the gap between bars, click a bar and select **Show Bar Gap** on the shortcut menu.

To show the borders of a bar, click a bar and select **Show Bar Border** on the shortcut menu.

**Clustered Bar Charts**

You can change your chart perspective so that the z axis data extended in the third dimension is shown as clusters displayed in the foreground. This charting option is useful when the z axis bars are hard to distinguish in standard bar formats.

You can use clustered bar charts to juxtapose categories in one label item. For example, use clustered bars to compare stores of different types. Alternatively, cluster bars can be used to compare two different value items, such as Amount of Sales and Units Sold.

*Note:* You can only display clustered bar charts in vertical format.

To cluster bars representing divisions in label items (clustered on the z axis):
1. Select **Vertical Cluster Bar** from the Chart Type drop-down list box.
2. Drag a fact item from **Elements** to the y pane and label items to the z axis and x axis panes in the data layout.

To cluster bars representing two different value items (clustered on the y axis):
1. Select **Vertical Cluster Bar** from the Chart drop-down list box.
2. Drag two fact items to the y axis and a label item to the x axis.

To display bar values:
1. Select **Show Bar Values** on the shortcut menu.
2. Select the placement of values on the bar:
   - Inside
   - Over
Outside
None

To show the gap between bars, click a bar and select **Show Bar Gap** on the shortcut menu.

To show the borders of a bar, click a bar and select **Show Bar Border** on the shortcut menu.

**Stacked Bar Charts**

Another way to represent the third dimension of data is through stacking. In this way, a single bar on the chart can show data for more than one category of data. For example, a single bar can represent the amount of sales for CD-ROM drives in one year on top of a bar representing sales for other years. You can stack the bar charts vertically or horizontally.

Stacked bar charts show the relationship of parts to the whole. Stacking techniques differ depending on whether you are Stacking Categories within Data Labels or Stacking Categories of Numeric Data.

Stacked bar charts offer similar complexity to clustered bar charts by adding component value items within chart bars or areas. By stacking items and assigning a different color to each item, you can effectively display trends among comparable or related items, or visually emphasize a sum of several indicators.

To create a Stacked bar chart, you need more than one Request item in the Y-Facts in the data layout. Each value item adds a segment to the length of the bar.

To display bar values:

1. Select **Show Bar Values** on the shortcut menu.
2. Select the placement of values on the bar:
   - Inside
   - Over
   - Outside
   - None

To show the gap between bars, click a bar and select **Show Bar Gap** on the shortcut menu.

To show the borders of a bar, click a bar and select **Show Bar Border** on the shortcut menu.

**Area Charts**

Area charts are essentially bar charts with the discontinuous breaks removed along the horizontal axis. Data is not broken into discrete bars but is displayed in a continuous ebb and flow as defined against the Y-axis. Consequently, area charts are particularly useful for emphasizing the
magnitude of change over time. In addition, area charts can be used for the same purposes as bar charts.

Because area charts do not break data along the horizontal axis, they are most useful for charting three dimensions of data. The z pane should be used to either project data into a third spatial dimension, or to stack two categories of data in a stacked area chart.

Creating an Area Chart

To create an area chart:

1. Select Area from the Chart drop-down list box.
2. Drag a value item from Elements to y pane in the data layout.
3. Drag a label item from Elements to the z pane in the data layout.
4. Select Legend On Z from the Legend drop-down list box.

Setting the legend on the z axis properly distributes color.

Creating a Stacked Area Chart

To create a stacked area chart:

1. Select Stacked Area from the Chart drop-down list box.
2. Drag a value from Elements to the y pane in the data layout.

Ribbon Charts

A ribbon chart is very similar to a line chart but with a few visual differences. In a ribbon chart, items in the y axis determine the height of the line, and items in the x axis itemize the line sections. You can create multiple lines by adding items to the z pane.

To create a ribbon chart:

1. Select Ribbon from the Chart drop-down list box.
2. Drag a value item from Elements to the y pane in the data layout.
3. Drag a label item from Elements to x pane and one or more label items to the y pane in the data layout.

Line Charts

Line charts show trends in data at equal intervals and are effective for comparing highs and lows in a continuum. In a line chart, items in the y axis determine the height of the line, and items in x axis itemize the line sections. You can create multiple lines by adding items to z axis.
Line charts have one advantage over bar charts. They do not enable one set of data to obstruct the representation of another. Since lines are thin compared to bars, the data displayed in the front does not block out the data behind.

As a result, data that is not easily represented in bar or area charts work well in line charts. Many more dimensions of data can be superimposed without impairing the effectiveness of the chart.

**Note:** A line chart can have two different looks depending on if the chart was switched from a stacked or non stacked chart. If a line chart was switched from a stacked chart, the z axis items on the chart are stacked. If a line chart was switched from a non stacked chart, the z axis items are not stacked and are displayed as originally created.

➤ To create a standard line charts y axis:

1. Select **Line** from the Chart drop-down list box.
2. Drag a value item from **Elements** to the Facts pane in the data layout.
3. Drag label items from **Elements** to the X pane in the data layout.
4. Drag items from **Elements** to the Depth pane in the data layout.
5. Select **Legend On Facts (Depth)** from the Legend drop-down list box to distribute colors along the z axis.

➤ To show line values, click a line and select **Show Line Values** on the shortcut menu.

➤ To show marker border, click a line and select **Show Marker Border** on the shortcut menu.

**Time Aware Axis**

The Time Aware Axis feature allows you to show dates in chronological order plotted at specific intervals within minimum and maximum bounds. To do this, the Time Aware Axis feature turns a discrete X-Axis into a continuous time interval. The distance between adjacent axis items is proportional to their time value difference (a non Time Aware Axis shows all items using the same difference between them). The Time Aware feature only implements an alternative visualization of source data and does not affect the way how the data is aggregated and computed items are calculated. That is, the data processing in chart section which includes dividing data into categories and calculating fact data does not depend on whether the Time Aware feature is activated. In particular, the behavior of the “Chart This Pivot”, “Pivot This Chart” and “Add Computed Item” actions do not change.

For example, the data below represents the total sales per month:

<table>
<thead>
<tr>
<th>Date</th>
<th>Total Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/1/2005</td>
<td>$2,583,200</td>
</tr>
<tr>
<td>2/1/2005</td>
<td>$3,551,700</td>
</tr>
<tr>
<td>6/1/2005</td>
<td>$2,011,210</td>
</tr>
<tr>
<td>7/1/2005</td>
<td>$1,149,210</td>
</tr>
</tbody>
</table>
A Line Chart that has no Time Aware axis, shows the data as follows:

![Line Chart](image)

The four ticks corresponding to the month are arranged uniformly along the axis (the distance between them is the same). At first glance, it appears that the sales reduction trend becomes slower in the last month. If you look at the graph closer, there is no March to May period.

If you add a Time Aware axis, the same chart looks like this:

![Line Chart with Time Aware Axis](image)

This chart shows every month on the present time values; even month for which there is no data. Sales for the last month drop quickly.

The Line Charts is the main application area of this feature; however it is available in almost all existing chart types. Scatter/Bubble charts do not include this option because they are already “time aware” (essentially, value aware) by nature. Both have two fact axes which are continuous by definition.

The Time Aware axis can only be used under specific conditions:

- The X axis should have a date/time category to display, which becomes a dedicated Time Scale category.
- Only one category should be on the X-Axis. If you add multiple categories, even if they are date/time categories, the feature is rendered inactive.
- The Time Aware axis is not available for pie, scatter and bubble charts.
The Time Aware Axis is considered active if the conditions in the list above are met, and the Time Aware option on the Label Axis dialog is not explicitly disabled. You can turn on or off the feature. If you turn off the feature, the X axis remains discrete as in previous versions. By default the feature is turned off for Interactive Reporting documents older than Release 9.3. Charts created in Release 9.3 and later have the feature enabled.

**Note:** Since the Time Aware Axis assumes that all axis labels are in ascending order, the sort order option is disabled.

➤ To create a Time Aware axis:

1. Check the Time Aware option on the Label Axis tab of General properties.
2. Drag a date/time item from *Elements* to the X pane in the data layout.
3. Drag a value item from *Elements* to the Facts pane in the data layout.

### Combination (Bar/Line) Charts

Combination charts combine some of the strengths of bar charts with the advantages of line charts. Solid bars can be used for the most important data against which other dimensions are represented in lines. In this way, emphasis is given to a portion of data based on its importance. A combination chart is especially useful for comparing two numeric values, such as amount and units of sales.

➤ To create a combination chart that compares values:

1. Select **Bar-Line** from the Chart drop-down list box.
2. Drag two values from *Elements* to the y pane in the data layout.
3. Drag label items from *Elements* to x pane and z pane in the data layout.

➤ To create a combination chart that compares categories within a label:

1. Select **Bar-Line** from the Chart drop-down list box.
2. Drag a value item from *Elements* to the z pane in the data layout.
3. Drag one or more value items from *Elements* to the y pane in the data layout.
4. Drag a label to the x pane in the data layout.

**Note:** A combination chart is most effective when the y pane contains only two value items. It represents one value as bars and the other value as a line. When more than two values are present, the chart alternates between bars and lines in depicting the values (1st, 3rd, 5th … items are bars; 2nd, 4th, 6th … items are lines).
To display line values, select **Show Line Values** on the shortcut menu.

To display bar values:
1. Select **Show Bar Values** on the shortcut menu.
2. Select the placement of the values on the bar:
   - Inside
   - Over
   - Outside
   - None

To show the gap between bars, click a bar and select **Show Bar Gap** on the shortcut menu.

To show the borders of a bar, click a bar and select **Show Bar Border**.

To toggle fact values as bar or line, click an item in the Fact pane, and select either **Display as Line** or **Display as Bar**.

**Reference Lines, Baselines, and Trend Lines**

Use the Reference, Baseline and Trend line tools to visually reveal trends in your data set, and to make reasonable predictions about future values. These tools include:

- Reference Line—A horizontal or vertical line drawn in the diagram to indicate a user defined computed value.
- Trend line—A line connecting two or more data points representing a linear regression model of data. Generally, the trend line slants because it reflects the movement of a value’s increase or decrease over time.

**Reference Lines**

A Reference Line is a horizontal or vertical line drawn in the diagram to indicate a computed value. It is typically used to illustrate or compare a fixed value within a category of values, or an average, maximum or minimum of items. Reference lines can be drawn on the top of visible graphic elements of a diagram (for example, bars or lines). By default the lines are drawn on the top. Reference lines can be shown with text label, and reference line information in the legend.

There are two types of reference lines:

- Fact column—Fact based reference line to which a statistical function has been assigned (average, minimum or maximum). A fact based is oriented (horizontal or vertical) based on the chart type.
- Axis—A non-fact fixed constant value reference line. An axis based reference line is oriented based on the axis to which it is associated.
There can be several reference lines created for the same fact column having different statistical functions associated with each one. Similarly, multiple instances of axis based reference lines can exist for a particular axis. When the statistical function of a reference line is calculated, not only the data from current page, but also the data from all pages of a multi-page chart (or zoomed chart) are included. The category items that are hidden as a result of applying ‘Hide Items’ or ‘Focus on Item’ or ‘Drill Anywhere’ are not included.

If the fixed value assigned to the line is beyond the data range currently shown on fact axis and the axis set up as auto scaled, the data range is expanded automatically for the line to be able to appear.

By default a reference line has a text label associated with it showing auto-generated text. The axis based reference line shows the assigned number value, and the fact based reference line. The location of the text label is set automatically and cannot be adjusted by a user.

This example shows a fact based reference line associated with an Amount Sales fact column assigned to the Average function.

2D and 3D Reference Lines

Reference lines should be used primarily with two dimensional perspective, but are supported with a three dimensional (3D) perspective. If they are used with a 3–D perspective, several parallel lines are drawn along the Z-axis together with graphical data.

Reference Lines and Chart Types

The Stacked Bar, Stacked Area charts both allow axis-based and fact-based Reference lines. A single fact-bound Reference Line is drawn for summed stacked items.

A Pie chart cannot have any type of reference line.

The fact based reference line is always associated with a fact column. When the column is removed from the Chart data layout, the corresponding reference line is also removed. When the fact column is hidden or focused, the reference line is also hidden, or focused.
If the chart type is switched to another chart type, the Reference Lines is hidden. Switching back to the original chart type restores the lines.

**Adding and Modifying a Reference Line**

- To add a reference line, select a chart item that represents a data fact and on the shortcut menu, select *Add Reference Line*.

- To modify a reference line:
  1. Double-click the reference line.
     The Reference Properties dialog box is displayed.
  2. Select the Reference Line tab.
  3. Select *Fixed* to assign an axis based reference line, or a fact item from the Fact drop-down.
     For more information about adding fixed and fact based reference lines, see Reference Line Properties.
  4. To assign a statistical function to a fact based reference line, select a function from the Function drop-down list.
  5. Select any additional reference line properties.
  6. Select the Line Style tab, and select any line style properties.
     For more information, see Reference Line Style Properties.
  7. Select the Label Format tab, any select any label properties.
     For more information, see Reference Line Label Format Properties.
  8. Select OK.

- To delete a reference line, select the reference line and click Delete.

**Reference Line Properties**

Use the Reference Line dialog box to select the type of data to associate with the reference line, and to specify general reference line properties.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed</td>
<td>Enable to assign a fixed value to the reference line (axis based). This option is used in conjunction with the edit field and Axis field. When this option is enabled the “Fact” field name is changed to the “Axis” field name.</td>
</tr>
<tr>
<td>(edit box)</td>
<td>Specify the fixed value amount to use for the reference line. The amount must be numeric. It remains constant on the reference line, and does not depend on another item in the Fact pane of the data layout.</td>
</tr>
</tbody>
</table>
### Axis
Select the fact axis on which to position the reference line. Available axes:
- X axis
- Y axis
- secondary Y1 axis

**Note:** A reference line can be added to the X axis only if the axis continues as in the case of the Scatter and Bubble charts. If the fixed value assigned to the reference line is beyond the data ranged shown on the fact axis, and the axis is auto scaled, the data range is expanded automatically so the line can appear.

### Fact
Select the fact based column for the reference line from the drop-down. Available columns are based on the columns in the data set.

### Function
Select the function to apply to the fact based column. Available functions are:
- Average
- Maximum
- Minimum

### Bring to front
Positions the reference line in front of the chart item.

### Send to back
Positions the reference line in back of the chart item.

### Show in legend
Enable to show assigned text of the reference line and short line segment representing the actual color of the line in the legend.

### Show label
Enable to show the label text on the diagram.

### Reference Line Style Properties
Use the Reference Line Style Properties dialog box to define line styles.

#### Table 57  Reference Line Style Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatic</td>
<td>Enables the display of the default line color, width and pattern style.</td>
</tr>
<tr>
<td>None</td>
<td>Option is not available.</td>
</tr>
<tr>
<td>Custom</td>
<td>Enables you to select the line color, width and pattern style.</td>
</tr>
<tr>
<td>Color</td>
<td>Select to launch the palette and select another color. This option is only available for a custom line style.</td>
</tr>
<tr>
<td>Width</td>
<td>Select the width (weight) of the line. Available widths are from 1 point to 6 points. This option is only available for a custom line style.</td>
</tr>
</tbody>
</table>
## Reference Line Label Format Properties

Use the Reference Line Label Format dialog box to define line label (plot area) and legend text properties.

### Table 58 Reference Line Label Format Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Style</strong></td>
<td>Select the pattern of the line. Available styles are:</td>
</tr>
<tr>
<td></td>
<td>- Solid</td>
</tr>
<tr>
<td></td>
<td>- Dash</td>
</tr>
<tr>
<td></td>
<td>- Dot</td>
</tr>
<tr>
<td></td>
<td>- DashDot</td>
</tr>
<tr>
<td></td>
<td>- DashDotDot</td>
</tr>
<tr>
<td></td>
<td>This option is only available for a custom line style.</td>
</tr>
<tr>
<td><strong>(Text Format for Plot Area or Legend)</strong></td>
<td>Formats text for the plot area or legend from the drop-down.</td>
</tr>
<tr>
<td></td>
<td>If the “Use the same format in plot area and legend field” is enabled, this option is disabled.</td>
</tr>
<tr>
<td><strong>(Custom Format)</strong></td>
<td>Specify a custom format for the plot area or legend.</td>
</tr>
<tr>
<td></td>
<td>A custom format can combine constant text and generated strings for a statistical function or the value of the reference line.</td>
</tr>
<tr>
<td></td>
<td>Constant text accepts the following tags:</td>
</tr>
<tr>
<td></td>
<td>- [FC]—fact name</td>
</tr>
<tr>
<td></td>
<td>- [VL]—value of reference line</td>
</tr>
<tr>
<td></td>
<td>For example, the custom format: “Expected sales = [VL]” could return the results in the label or legend: “Expected sales = $26300000”. If the text for a tag cannot be generated (for example, the format is for an [FN] tag on an axis based reference line), the tag resolves in an empty string, and it is removed.</td>
</tr>
<tr>
<td></td>
<td>A custom format can include complex tags combining arbitrary text with one or more simple (and even complex) tags. A complex tag is bounded by curly braces ({}), for example, “[Sales [FN] = ][VL]”. If at least one of the tags inside the complex tag cannot be resolved, all complex tags result in an empty string.</td>
</tr>
<tr>
<td></td>
<td>A fact based reference line using the format above might be resolved in a “Sales Average = $126000” string, and for an axis based reference line it could be resolved in a “$126000” string.</td>
</tr>
<tr>
<td></td>
<td>The special delimiter tag, [DL] is available. This tag inserts a space in the resulting string if both the left and right tags are successfully resolved.</td>
</tr>
<tr>
<td></td>
<td>The complex tag, [; [DL] ] inserts custom text as a delimiter. For example, the format “(<a href="%5BFC%5D">FN</a>)[ ; [DL] ][VL]” may result in one of the following strings depending on conditions:</td>
</tr>
<tr>
<td></td>
<td>- Average(Sales)</td>
</tr>
<tr>
<td></td>
<td>- Average(sales) = $126000</td>
</tr>
<tr>
<td><strong>Auto format</strong></td>
<td>The following Default, Function and Value fields can be enabled to reset the label text to auto-generated text on the line. Each field adds predefined tags to the text format. If no auto-format is enabled you can enter a custom format in the edit box.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>Default</td>
<td>Show the default text: &quot;&lt;Function name&gt;(&lt;Fact name&gt;) = &lt;Value&gt;&quot;.</td>
</tr>
<tr>
<td>Function</td>
<td>Shows the statistical function in auto-generated text, for example: Average.</td>
</tr>
<tr>
<td>Value</td>
<td>Shows the value of the reference line, for example: Value = 0.7348.</td>
</tr>
</tbody>
</table>

**Baselines**

Use a baseline to shift the position of plotted data in the chart based on a threshold amount. Baselines are an effective tools to track actual progress against a targeted value. Baselines operate by changing the coordinate origins of the diagram. They gives the appearance of moving the X, Y or Y1 axis to another position. Data above the baseline amount appears in the top sector of the diagram separated by a break line from data below the baseline.

There a separate baseline settings for Y and Y1 axes. When the baselines is enabled, the horizontal axis (or horizontal plane for 3D charts) intersects both Y and Y1 axes at their baseline values.

In the following example, the baseline has been set at $150,000 for amount sales. Berlin, British Columbia, and Stockholm are above the baseline. Bayern and Nordhein Westfhalen are below the baseline.

To add a baseline:

1. **Select the chart and on the shortcut menu, select Properties.**
   The Chart Properties dialog box is displayed.
2. **Select the Values Axis tab.**
3. **Enable the Baseline field.**
4 Enter a baseline amount in the edit box.
5 Click OK.

**Trend Lines**

Trend lines are used to track trends in a data series graphically. Interactive Reporting supports trend lines modeled after linear regression analysis. Generally, the trend line is represented as a slanted line that crosses the diagram. For example, the trend line can demonstrate an increase or decrease of values over time. It may be accompanied with the calculated goodness of fit (R-squared) value.

Trend lines can be layered on top of the chart graphics (or Z axis for 3D charts), or positioned to the background. When data is processed to create the trend line, facts from all pages of the chart are included. Chart items hidden or focused explicitly by the user are not included.

Trend lines are always fact based, and only one trend line can be associated with a single fact column. In Scatter and Bubble charts, the trend line is bound to the data series.

![Trend Line Diagram](image)

**2D and 3D Trend Lines**

Trend lines should be used primarily with two-dimensional perspective, but are supported with a three-dimensional (3D) perspective. If they are used with a 3-D perspective, several parallel lines are drawn along the Z-axis together with graphical data.

**Trend Lines and Chart Types**

Trend lines can be added to most chart type including stacked charts (Bar and Area). The value of each stack is included when the trend equation is calculated. When the separate stackables of a full bar belong to different facts, then a single trend for all facts is drawn. The sum of separate facts (either positive or negative) is included when calculating the trend line equation.

Trend lines are most effective in Scatter, Bubble, and Time Aware charts.

A trend line with a single category on the X axis is preferable. In cases where there are multiple categories on the X axis, it might be difficult to analyze the trend if the categories are unrelated.
Pie charts cannot have trend lines.

**Adding and Modifying a Trend Line**

To add a trend line, select a chart item that represents a data fact and on the shortcut menu, select **Add Trend Line**.

To modify a trend line:

1. **Double click the trend line.**
   The Trend Line Properties dialog box is displayed.
2. **Select the** **Trend Line** **tab.**
3. **From the Fact drop-down box, select another fact item on which to place a trend line.**
4. **Select** **Bring to front**, **to move the trend line in front of the data item.**
5. **Send to back, to move the trend line behind the data item.**
6. **Select the** **Line Style** **tab.**
   The Trend Line Style dialog box is displayed.
7. **Select any line properties to modify.**
   For more information about line properties, see **Trend Line Style Properties**.
8. **Select the** **Label Format** **tab.**
   The Trend Line Label Format dialog box is displayed.
9. **Select any label formats to modify,**
   For more information about label formats, see **Trend Line Label Format Properties**.
10. **Click** **OK**.

To delete a trend line, select the trend line and click **Delete**.

**Trend Line Properties**

Use the Trend Line Properties dialog box to select general trend line properties.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fact</td>
<td>Select the fact item on which to assign the trend line. Multiple trend lines cannot be assigned to a single fact item.</td>
</tr>
<tr>
<td>Bring to front</td>
<td>Positions the reference line in front of the chart item.</td>
</tr>
<tr>
<td>Send to back</td>
<td>Positions the reference line in back of the chart item.</td>
</tr>
<tr>
<td>Show label</td>
<td>Enable to show the label text on the diagram.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Show in legend</td>
<td>Enable to show assigned text of the trend line and short line segment representing the actual color of the line in the legend.</td>
</tr>
</tbody>
</table>

**Trend Line Style Properties**

Use the trend Line Style Properties dialog box to define line styles.

**Table 60  Trend Line Style Properties**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatic</td>
<td>Enables the display of the default line color, width and pattern style.</td>
</tr>
<tr>
<td>None</td>
<td>Option is not available.</td>
</tr>
<tr>
<td>Custom</td>
<td>Enables you to select the line color, width and pattern style.</td>
</tr>
<tr>
<td>Color</td>
<td>Select to launch the palette and select another color. This option is only available for a custom line style.</td>
</tr>
<tr>
<td>Width</td>
<td>Select the width (weight) of the line. Available widths are from 1 point to 6 points. This option is only available for a custom line style.</td>
</tr>
<tr>
<td>Style</td>
<td>Select the pattern of the line. Available styles are:</td>
</tr>
<tr>
<td></td>
<td>• Solid</td>
</tr>
<tr>
<td></td>
<td>• Dash</td>
</tr>
<tr>
<td></td>
<td>• Dot</td>
</tr>
<tr>
<td></td>
<td>• DashDot</td>
</tr>
<tr>
<td></td>
<td>• DashDotDot</td>
</tr>
<tr>
<td></td>
<td>This option is only available for a custom line style.</td>
</tr>
</tbody>
</table>

**Trend Line Label Format Properties**

Use the Trend Line Label Format dialog box to set up the line label and legend text.

**Table 61  Trend Line Label Format Properties**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Text Format for Plot Area or Legend)</td>
<td>Format text for the plot area (line label) or legend text from the drop-down. If the “Use the same format in plot area and legend” field is enabled, this option is disabled.</td>
</tr>
</tbody>
</table>
### (Custom Format)

Specify a custom format for the plot area or legend. A custom format can combine constant text and generated strings to show the trend line name, the equation type, or a coefficient of determination (R-squared).

Trend line formats default formats that can be customized include:

- Trend((IFC))—Trend(<Fact name>)
- [ET]—Equation Type (Linear only)
- {R squared=[RS]}—Coefficient of determination (how good the fitness is), for example, R-squared=0.7349.

### Auto format

The following Default, Equation type and R-squared fields can be enabled to reset the label text to auto-generated text on the line. Each field adds predefined tags to the text format. If no auto-format is enabled you can enter a custom format in the edit box.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>Shows the default text: “Trend(&lt;Fact name&gt;)”</td>
</tr>
<tr>
<td>Equation Type</td>
<td>Shows the equation name in auto-generated text, for example: linear.</td>
</tr>
<tr>
<td>R-squared</td>
<td>Shows coefficient of determination (how good the fitness is, or how good the trend line conforms to the data) in a value range from 0–1, for example R-squared=0.7349.</td>
</tr>
</tbody>
</table>

### Manipulating Chart Data

Interactive Reporting offers a number of ways to manipulate the data in your chart for better analysis. Review these topics:

- “Using Different Scales to Compare Related Values” on page 238
- “Using Data Functions in Charts” on page 239
- “Adding Computed Items” on page 240
- “Sorting Chart Items” on page 241
- “Creating Pivot Tables from Charts” on page 242
- “Drilling into Charts” on page 242
- “Hiding and Focusing on Charted Data” on page 243
- “Changing Angle and Elevation (Rotating)” on page 244

### Using Different Scales to Compare Related Values

To chart comparison values or to combine two related indicators on the same chart, you may need to compensate for different numeric scales. For line, clustered bar, and combination charts, you can use a second Y-axis to represent values on a scalar function that differs from the scale of the first y axis.
For example, you might like to chart the sales of your two fastest growing product lines together to get an idea of how business is growing. However, while the growth rates are similar, the two product lines may sell at entirely different volumes. The chart does not provide much comparative information because each line needs to be charted at a different scale. By using different scales for the two y axes, you can correctly scale each value for the most effective presentation of the data.

To use a different scale for a value on the second y axis, double-click the one of the two values in the y pane in the data layout.

Interactive Reporting changes the scale of the y axis for the value automatically, and italicizes the value name in the data layout. Double-click the item again to return the scale to its default setting.

### Using Data Functions in Charts

You can change the way that values are calculated in a chart. For example, you may want to display average sales instead of total sales. You can apply data calculations to the values represented by one or more chart element. Different chart elements represent different types of values. For example, one bar can represent sums and another bar can represent averages.

Chart item totals added to your report are aggregates (literally, totals of totals), and can be recalculated using a Dashboard. When applied to totals, data functions can either apply the calculation to the "surface" or "underlying values".

When applied to surface values, data functions recalculate the values in the visible cells or "surface" of the Chart. When applied to underlying values, data functions return to the unaggregated values beneath the Chart and recalculate based on those values. When "underlying values" are used, the results often are shown incongruous with the aggregate "surface" values of the chart element. In other words, a total of the underlying values does not match the total of the surface figures.

To match surface-level Chart values in your calculation, you can instead use "surface" total functions. For example, if you apply a surface average to a total, the total is converted to the average of the surface values in the corresponding element.

To apply a data function:

1. **Select a label in the Chart area.**
2. **Select Chart, then Data Function, and then Function.**

<table>
<thead>
<tr>
<th>Data Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sum</td>
<td>Returns sum of all values. This is the default setting.</td>
</tr>
<tr>
<td>Average</td>
<td>Returns average of all values.</td>
</tr>
<tr>
<td>Count</td>
<td>Returns number of values.</td>
</tr>
<tr>
<td>Maximum</td>
<td>Returns highest value.</td>
</tr>
<tr>
<td>Data Function</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>------------------------------------------------------------------</td>
</tr>
<tr>
<td>Minimum</td>
<td>Returns lowest value.</td>
</tr>
<tr>
<td>% of Grand</td>
<td>Returns values as a percentage of all like values in the chart.</td>
</tr>
<tr>
<td>Non-Null Average</td>
<td>Returns average of values; null values excluded.</td>
</tr>
<tr>
<td>Null Count</td>
<td>Returns number of null values.</td>
</tr>
<tr>
<td>Non-Null Count</td>
<td>Returns number of values; null values excluded.</td>
</tr>
</tbody>
</table>

**Note:** Null values are empty values for which no data exists. Null values are not equal to zero.

## Adding Computed Items

You can create new chart elements by building equations to compute data items, or by applying functions to data items. Computed items are like normal data items, and can be included in charts or used to compute other data.

For example, you can modify the *Amount Sold* item by building an equation around it, multiplying it by the *Unit Price* item and renaming the resulting item *Revenue*. You can also apply a scalar function such as `Cume` to *Amount Sold* and return each individual value as a cumulative running total, or simply multiply *Amount Sold* by the local tax rate to find the tax owed on each sale.

---

To create a computed item:

1. **Select Add Computed Item from a Section menu (for example, Query, Results, and so on).**
   
The Computed Item dialog box is displayed.

2. **In the Name field, type a name that describes the computation.**
   
The default name is Computed, which is numbered sequentially if there is more than one. If you assign a name to a computed item that is identical to a scalar function name, Interactive Reporting numbers the name starting with the number 2.

3. **Define the new data item by building an expression in the Definition text box.**
   
Use the operator buttons to insert arithmetic and logical operators at the insertion point.

   - Click **Reference** to display the Reference dialog box, and select Request items to place in the equation.
   - Click **Functions** to apply scalar functions using the Functions dialog box.
     
     You can also type any portion of the equation or the entire equation directly into the Definition text box using JavaScript. The names are case sensitive, and you must replace spaces in item names with underscores (‘_’).

4. **If necessary, click the Options button to set a new data type for the item.**

5. **When the equation is complete, click OK.**
The computed item is listed in the data layout and is added to your chart.

**Sorting Chart Items**

Data in charts is sorted alphabetically by default. You can use the sort buttons on the Standard toolbar to perform simple sorts on selected items and reverse the sort order. In charts, however, you generally want to override the default alphabetical setting and sort dimensional data with reference to other data.

For example, if a chart lists each type of widget your company sells and the total amount sold of each, initially the widget types are ordered alphabetically. But this data becomes more meaningful when you instead sort the widget types with reference to the total produced by each. This approach enables you to rank each widget from the highest to lowest total sales.

You can use the Sort line in the Chart section to impose a sort condition for each dimensional data item in your chart. The Sort line includes three drop-down menus used to define the sort conditions. The contents of the menus vary depending on the data items in your chart.

**Note:** Sorting cannot be used for scatter or bubble chart items.

To specify a sort using the Sort line:

1. **If the Sort line is not already displayed, click Sort on the Section title bar.**
2. **Select an item to sort from the Sort drop-down list box.**
3. **Select a value from the By drop-down list box as a sort reference, or select Label to sort the item alphabetically.**
4. **If desired, select an aggregate function from the Using drop-down list box when sorting by values.**
   - The Using drop-down menu is not available when you sort by labels.
5. **If desired, click the ascending or descending Sort button on the Sort line.**
   - The Sort line stores a sort condition for each dimensional item included in the chart.

**Sort Items**

The sort drop-down menu lists the data items that can be sorted. Each dimensional item included in the chart (name and date) is listed in this menu. Dimensional items can include Pivot column and row labels.

**Reference Items**

The By drop-down menu lists items used as a basis for a complex sort condition (for example, sorting Cities by the revenue generated in each).

- **Label**—By default, dimensional data items is sorted alphabetically by name when you create your chart: this is equivalent to sorting by label. When selected, labels indicates that the item
chosen from the Sort list is sorted by label or name, rather than by reference to corresponding numeric data values in the chart.

- Value—Sorting by a numeric data item orders each value of the target item chosen from the Sort list by its corresponding numeric value in the Value list.

Sorting by values produces an entirely different sort order. For example, your chart may list each state in which your company has made sales revenue and the total cost-of-sales for each. The states are initially listed in alphabetical order. When you sort by cost-of-goods, the states are ranked in order by each corresponding cost-of-sales figure.

**Functions**

The Using drop-down menu contains aggregate statistical functions that are available when you sort by values. The sort aggregate functions are usually the same as the data functions available in a section. When you sort by values, labels are sorted by the corresponding numeric values of the referenced item (for example, sorting states by the sum total of the cost of goods sold in each state).

**Note:** Data Functions are not available for the scatter and bubble chart types because these types of charts do no use aggregated data (they retrieve their data from the equivalent of their underlying Table/Results section.)

**Creating Pivot Tables from Charts**

Once you have a final version of your chart, use the automatic pivot table-generator to create a *pivot table* based on the layout of your chart.

**Note:** The Pivot this Chart feature is not available for scatter and bubble chart types.

- To create a pivot table based on your chart, in the Chart section, select **Insert**, then **Pivot This Chart**.

**Drilling into Charts**

The drill into feature enables you to drill into items in the Chart section that are resident in the Results section without having to return to reprocess your query or locate the item in Elements. Drill into items are automatically added as new label items.

The advantage of this feature is that it instantly enables you to add items to the data set to reflect temporary or hypothetical situations. You can always suspend or delete the item to return to the original chart display.

The extent to which you can drill into your data depends on how the original query was built, since Drill Into retrieves data from the Results section based on the table hierarchy of items. This feature does not enable you to interactively query the database.
Note: You cannot drill into any items in a scatter or bubble chart because these types of charts use non-aggregated data.

To drill anywhere into a chart:

1. **Select one or more items for analysis and select Chart, then Drill Anywhere, and then Item.**

   Interactive Reporting redraws the chart drilled to the selected item. In the data layout, an item selected for drill-down is identified with a drill-bit icon.

2. **Select Chart, then Drillup to return to the original view of your chart.**

   Tip: If no options are available in the Drill Anywhere menu, all Request items have been used in the data layout.

   Note: Drill Anywhere is enabled on the General page of Data Model Options. To display the General tab, select DataModel, then Data Model Options.

**Hiding and Focusing on Charted Data**

A straightforward way to refresh your view of a chart is to single out items for closer focus or remove some of the charted elements. This enables you to concentrate on particular items of interest.

Note: You cannot use the Focus feature on scatter or bubble chart items. In addition, if you use the Hide feature for scatter or bubble charts, the whole data series is hidden and not an individual item.

**Focusing on Items**

To focus on a chart item:

1. **Select one or more objects on which to focus.**

   Focused items are displayed with a dotted outline.

2. **Select Chart, then Focus On Items.**

   The chart is redrawn to display only the chart object(s) selected. A drillbit icon is displayed in the data layout next to the item(s) on which you focused.
To return to the original chart display, select Chart, then Show All Items.

**Hiding Items**

To hide charted data:

1. **In the chart, select the object.**
   The item is displayed with a dotted outline.

2. **Select Chart, then Hide Items.**
   Interactive Reporting redraws the chart with the selected objects removed. A drillbit icon is displayed in the data layout next to the item(s).

**Restoring Hidden Items**

To restore hidden chart items, select Chart, then Show Hidden Items.

**Changing Angle and Elevation (Rotating)**

By default, charts are displayed in a three-dimensional, shortened perspective from above and to the right. You can alter the perspective from which a chart is viewed by rotating it within the Contents pane.

**Tip:** Before you can rotate a chart, the background plane must be visible. To display the background plane, click the plot area of the chart and click Properties on the shortcut menu. Next click Show Back Plane in the General tab of the Chart Properties dialog box.

To rotate the angle of vision or change the elevation of a bar, line or area chart:

1. **Place the cursor at the top right corner of the chart.**
   The cursor changes to indicate that you can rotate the chart.

2. **Hold down the mouse button and rotate the chart.**
   When you release the mouse button, the chart is redrawn to reflect the adjusted perspective. If you cannot rotate further in a certain direction, the chart has reached its farthest possible rotation in that direction.

To rotate the perspective angle or elevation of a pie chart:

1. **Click Rotate on the shortcut menu.**
   The rotate icon is displayed on the pie chart.

2. **Drag the bars on the rotate icon to a new location to rotate the chart.**
When you release the mouse button, the chart is redrawn to reflect the adjusted perspective. If you cannot drag the rotate icon in a certain direction, the chart has reached its farthest possible rotation in that direction.

**Working with Chart Elements**

In the Chart section, you can easily reorganize or reposition data to reconfigure your charts and highlight different relationships between the same items. You can drag items to a different order within an data layout, drag label items between data layout panes designated for labels, or delete items from the data layout your chart is redrawn to reflect your changes.

You can also work directly with elements in the Chart area. Most elements are selected by clicking the element in the chart or the element’s label in the legend. For axis labels, a change to one axis label changes all labels along that axis.

Review the following:
- Selecting Chart Elements
- Displaying Axis Grid Lines
- Inserting Text
- Chart Legends

**Selecting Chart Elements**

- To select chart elements as graphic objects in the Chart area:
  1. Place the cursor over the edge of an item in the Chart area.
     - The cursor changes to a move cursor.
  2. Select the object.
     - A gray outline is displayed around the item. You can move the selected chart item anywhere in the Chart area or resize it using the handles. Also, check the shortcut menu for additional options.

**Displaying Axis Grid Lines**

Axis grid lines are straight lines on a chart that provide a framework for measurement and reference. By default, grid lines are white, but the color can be customized. Typically, the x axis and Stack, Cluster and Depth (z axis) are used for label items and the Fact, Fact (Stack) and Fact (Depth) (y axis) shows values or facts (measurable items), such as units and amounts.

You can view or hide these axis gridlines depending on the chart you are designing.

- To toggle the display of axis grid lines:
  1. Click anywhere within the main plot area of the chart.
2 Select **Show X Axis Grid Lines** (or **Show Y Axis Grid Lines**) on the shortcut menu.

A check mark is displayed next to the selected option to indicate that the grid lines are visible. Select this option again to clear the check mark and remove the gridlines from the Chart area display.

**Adding a Second Y Axis**

To chart comparison values or combine two related indicators on the same chart, you may need to compensate for different numeric scales.

For example, you might like to chart the sales of your two fastest growing product lines together to get an idea of how business is growing. However, while the growth rates are similar, the two product lines may sell at entirely different volumes. If this is the case, the chart won't provide much comparative information because each line needs to be charted at a different scale.

By charting each item to a different axis, you can include both items correctly in the same chart. Each item in the values panes can be assigned to one of two axes, each of which can be auto- or manually scaled.

- To add a second Y-axis:
  1. Double-click an item in the values panes.

     A second Y-axis is added to the right of the chart, and is auto-scaled to match the item. The item is displayed in italics in the data layout.

  2. If desired, change the scale or modify the second axis using the Chart Values Axis Properties tab of the Chart Properties dialog box.

    **Tip:** You can add a second Y-axis to clustered bar, combination bar-line and line charts.

**Inserting Text**

You can insert text anywhere around or within the chart to further explain or emphasize a chart component.

- To insert text:
  1. Select **Insert Text** on the shortcut menu.

     The Set Inserted Text dialog box is displayed.

  2. Type the text in the text box and click **OK**.

     The text is displayed in the location where you initially invoked the Insert Text command.

**Changing Chart Backplanes**

By default the backplane of a chart is light gray
To change the color of the backplane:

1. Select the backplane of the chart.
2. Select the Fill Color icon.
   The Fill Color palette is displayed.
3. Select a color from the Fill Color palette.

Chart Legends

You can select the axis along which to distinguish your data by setting the chart legend on that axis. This is a great way to view values on the selected axis without rearranging the values in the data layout. A chart legend can be set on either the X Axis, Y-, or Z-axis. You can also reposition or resize a legend to take advantage of either the horizontal or vertical space within the chart area.

**Note:** Chart legend color settings are preserved when the chart type, chart legend axis, and number of chart axis label values are changed.

The following three examples show how to place the legend on different axes to alter the appearance and data shown by the same chart. In the first example, the legend has been set on the X Axis. In the second example, the legend has been set on the Facts axis. In the third example, the legend has been set on the Z axis:

![Chart Example](image)
To set the axis used for a chart legend from the Format menu:

1. **Select Format, then Set Legend On.**
   
   The Set Legend On dialog box is displayed.

2. **Select the axis on which to set the legend and click OK.**
   
   Colors are redistributed to highlight the data associated with the selected item and color coordinates corresponding labels in the Legend.

To set the axis used for a chart legend from the toolbar, select the axis on which to position the legend.

The selected Legend pull-down menu displays to indicate the new axis position on which the legend has been assigned. Interactive Reporting redistributes colors to highlight the data associated with the selected item and color coordinates corresponding labels in the Legend.

To resize a chart legend:

1. **Click a border on the legend.**
Sizing handles are displayed on each corner of the selected legend.

2 Drag a sizing handle until the legend is the desired size.

➢ To wrap text in a chart legend, select Wrap Legend Text on the shortcut menu.

Legend text wrapping is only allowed when:

● The longest text of items arranged in one column does not fit the width of the legend.
● Sufficient vertical space is available in the legend.

➢ To show the border around the legend, select Show Legend Outline on the shortcut menu.

By default, the legend outline is not shown.

Chart Scrolling and Scaling

You may want to scroll through data, enlarge your chart to better work with chart details, or change the perspective or angle from which a chart is viewed. Review the following sections for information on:

● Scrolling through Chart Data
● Rotating and Elevating Charts
● Smart Scaling
● Auto Resizing Charts
● Fitting Charts to Screen

Scrolling through Chart Data

When you add a Catalog dimension to the data layout of a Chart, the rendered Chart contains all bars and labels corresponding to the total number of unique items in the dimension. If the number of items added is large and as a result, all items cannot display in one view, you can scroll the entire contents of a chart either vertically or horizontally using the scrollbars. The horizontal scrollbar controls the scrolling of X Axis items, and the vertical scrollbar controls the scrolling of Stack, Cluster and Depth items.

Optionally, you or an administrator can define a specific number of items to show for each view of the chart. If the number of total bars is greater than the view size, a full page of bars displays. For example, assume each bar has three views and you want to display four bars (such as: A-B-C-D). When you first display the chart only bars A-B-C display, then when you scroll to the right bars B-C-D display.

If not all bars of a chart display in one view, the you can display the text “Partial View” as an indicator in the top left corner of the Contents pane. You can show or hide this indicator on a per chart basis by enabling or disabling the “Show partial view indicator” field on the General tab of Chart Properties.
The following rules apply to Chart scrolling:

- The chart legend displays only items visible in the current view.
- Scrollbars are not available for the Pie chart type.
- If all bars can display in one view, the scrollbars are disabled.

### Rotating and Elevating Charts

By default, charts are displayed in 3-D foreshortened perspective from above and to the right. You can alter the perspective from which a chart is viewed by rotating it.

Before you can rotate the chart, the background plane must be visible.

To rotate the angle of vision or change the elevation of a bar, line, or area chart:

1. **Select Chart, then Properties.**
   
   The Chart Properties dialog box is displayed.

2. **Click the General tab to view the General properties page.**

3. **In the Planes area, select the Show Back Plane check box and click OK.**

4. **Place your cursor at the top right corner of the chart.**
   
   The cursor changes to indicate that you can rotate the chart.

5. **Hold down the left mouse button and drag the chart to rotate it.**

   When you release the mouse button, the chart is redrawn to reflect the adjusted perspective. If you cannot rotate in a certain direction, the chart has reached its farthest possible rotation in that direction.

### Smart Scaling

Smart Scaling describes a method for ensuring that chart text label components are intelligently placed in relation to each other so that there is no overlapping, visual obstructions or clipping. Smart Scaling governs:

- Initial chart displays size
- Chart behavior during resizing
- Ability to move or reposition charts

You apply Smart Scaling to prevent overlapping and clipping of objects. Understanding the initial display, and using the Do Not Remove Text and Minimum Font as modifiers are key to the Smart Scaling algorithm.

Smart Scaling is enabled on the General tab of the Chart Properties. By default, Smart Scaling is enabled.
**Initial Display**

Smart Scaling relies on a display hierarchy or a draw order when it determines what is displayed. The display hierarchy is based on the number of components and the initial size of the chart boundary. The display hierarchy for the Chart sections is as follows.

Bar, Line and Area charts have this display order:

- Planes
- Bar/Lines/Ribbons
- Stack, Cluster and Depth values
- X axis values
- Fact, Fact (Stack) and Fact (Depth) axis values
- Legend
- Titles
- Axis
- Inserted text

Pie charts have this display order:

- Slices
- Slice Labels
- Legend
- Titles
- Inserted text

The display order of Scatter and Bubble Charts are:

As you add or modify various Chart label components, Smart Scaling uses the hierarchy to determine which components to add or omit when placing items.

For example, in a Pie Chart where several small slices are in the same vicinity of each other, some of the pie slice labels might typically overlap due to simple space allocations. If you have Smart Scaling enabled, some of the slice labels are simply not drawn when crowded conditions are encountered. If a current label overlaps a label, then the label is skipped and the next label is drawn. You can correct this by moving the label slice. If there is enough room for other labels to display unobstructed when the label slices are repositioned, additional labels are displayed. You can also enable the “Line to Label” feature to enhance the slice-label relationship, and identify and place items easier.

**Retaining Text**

When a chart object is reduced disproportionately to other objects in the bounding area and Smart Scaling is enabled, text may be eliminated based on the display hierarchy. You can prevent text from being removed by enabling the Do not remove text option available on the General
tab of Chart properties. In the case where a chart object is reduced too much, text is preserved, but may overlap other object or appear distorted

**Minimum Font Size**

When the overall size of a chart is modified, Interactive Reporting attempts to redraw chart components using the object’s font size. If it is not feasible to redraw the component in the space available, the font size is reduced by one point, and another drawing is attempted. This process is repeated until the component can fit within the boundaries defined for it, or until a minimum font size is reached.

You specify the minimum font size on the “General” tab of Chart Properties. The default value for the minimum font size is 8.

The components which are modifiable by the minimum font size are:

- y axis, x axis and z axis
- pie slice values
- legend

When the component is first reduced, it continues to show the original font size and not the user defined font size. If the minimum font size is reached for the supported components and it is still too large to fit with obstructing or being obstructed, then the item is removed from display.

When Smart Scaling is used in conjunction with Auto Resize, changes to the Chart size are limited to all those methods which do not involve the movement of the chart border.

If you select the handles of the chart border and reposition them as the method to resize the Chart when Smart Scaling and Auto Resize are enabled, Auto Resize is automatically disabled. The Smart Scaling property remains enabled.

**Moving Chart Components**

Almost all chart components can be moved independently of each other within the borders of the chart once is has been drawn. With the exception of bar values and axes values, all other chart components can be moved.

With Smart Scaling, the chart title, subtitle, axes labels and legend cannot be moved over each other and cannot be moved over the axes values text and the chart graphic.

If Smart Scaling is not enabled, a Chart component can be moved anywhere within the border of the chart. If the component is layered on top of a component, it overrides that component in the display.

To enable Smart Scaling:

1. Select Chart, Properties.
2. In the Chart Properties dialog box, click the General tab, and then Smart Scaling.
3. To use a minimum font size, add a font size in the Min Font Size box and click OK.
Auto Resizing Charts

You can use the following options to resize a chart:

- Drag the handles of the chart border, which redraws the chart to fit the new border dimensions,
- Enable the “Auto Resize” option on the General tab of Chart Properties. This feature automatically expands the chart border to fit the maximum height or width of the Contents pane.
- Enable the “Auto Resize” property and adjust the size of the entire Interactive Reporting Studio application by dragging the corner the application window. This redraws the chart components automatically.
- Enable the “Auto Resize” property and adjust the size of the Contents pane by adjusting the Section/Elements border. This redraws the chart components automatically.
- Enable the “Auto Resize” property and show or hide the Outline and or Sort line.

Resizing a chart can result in the overlap of chart components, especially when a chart is reduced in size. To prevent this overlap, you can use the Smart Scaling and Minimum Font Size features (both described above). By default Auto Resizing is enabled.

To enable Auto Resize:

1. Select Chart, then Properties.
   The Chart Properties dialog box is displayed.
2. Click the General tab to view the General properties.
3. Select the Auto Resize check box and click OK.

Fitting Charts to Screen

When the dimensions of a chart are smaller or larger than the actual size of the Contents pane, and you wish to maintain relative proportions between the chart and the actual size of the pane, use the “Fit to Screen” icon on the Chart toolbar.

To fit a chart to the Contents pane, click 📐

Customizing Chart Properties

Use the Properties command on the Chart menu to customize how your Chart appears. Review these topics for information:

- “Customizing Chart Patterns and Labels” on page 254
- “Auto-Arranging Chart Pie Labels” on page 255
- “Applying Chart Color Schemes for Data Points” on page 256
Customizing Chart Patterns and Labels

In addition to the generic label axis properties that apply to all charts, you can change properties of individual chart items, such as patterns, colors, and data labels. Review the following sections for information:

- Changing Chart Color Schemes and Fill Patterns
- Changing Chart Data Labels
- Changing the Color of Chart Elements, Lines, and Text

Changing Chart Color Schemes and Fill Patterns

You can redistribute chart colors to emphasize specific charted items listed in the data layout and change chart focus. Colors are differentiated along a specific axis.

To change the pattern or color scheme of a chart:

1. Double-click a chart element in the legend, or select an element in the chart (such as a bar or a pie slice) and select Chart, then Properties.
A Properties dialog box is displayed that may contain one or more tabs (Patterns, Axis, Data Labels), depending on the format of the active chart.

2 Select the Patterns tab to display the Patterns tab.

3 Select the fill pattern and color for the chart segment, line, or marker and click OK.
   - **Automatic**—Sets the fill pattern and color to the default setting. The default setting for fill pattern is Solid. The default setting for color varies between chart formats.
   - **None**—Sets a transparent fill pattern and color for the selected element.
   - **Custom**—Enables you to select a foreground color and fill pattern for the chart. Choices include Solid, Hollow, Horizontal, Vertical, Cross, Diagonally Up, Diagonally Down, Diagonally Cross.
     
     For line charts, you can also select the width, style, and color of lines, and the size, style, border colors, and fill colors of markers.

**Changing Chart Data Labels**

You can change the type of data used in the labels in your chart. The choices available depend on the chart format.

➤ To change a data label:

1 Double-click a chart element in the legend, or select an element in the chart (such as a bar or a pie slice) and select Chart, then Properties.

2 Click the Data Labels tab to display the Data Labels tab.

3 Select the fill pattern and color for the chart segment, line, or marker.
   - **Pie Charts**—Shows pie slice labels, values, percentages, negative values, and lines that connect a data label to its pie slice.
   - **Bar Charts**—Shows individual bar values and the placement of the values inside the bar.
   - **Line Charts**—Shows values and ignore null values, and select the placement of the values relative to the line.

For line charts, use the Axis page to specify whether to plot values on the primary (left) axis or secondary (right) axis.

**Auto-Arranging Chart Pie Labels**

If you reposition a pie label manually and the slices are later rearranged, the position of the label remains unchanged. To reposition the label automatically with the slice, enable the Auto Arrange option on the label context menu.
To auto-arrange chart pie labels, select **AutoArrange** on the shortcut menu.

## Applying Chart Color Schemes for Data Points

This section explains legacy, standard and custom color options available in the Chart section.

You can choose either legacy or standard color scheme to represent graphical data point (for example, bars, markers, etc) on charts. The color scheme is a cyclic sequence of colors which determines what color to assign by default to each data point on chart. Both legacy and standard color scheme consists of seventeen colors as shown below

**Note:** Live charts use the color scheme defined in Default Formats. When the Live Chart is created, the color scheme is defined in Default Formats, but the color scheme persists only at the chart level. If the user changes the default color scheme later, this action does not affect existing charts.

- **Legacy chart color palette:**
  ![Legacy chart color palette](image)

- **Standard chart color palette:**
  ![Standard chart color palette](image)

- **Custom chart colors.**

## Standard and Legacy Chart Color Palette

By default Interactive Reporting bases the color scheme used in charts on the standard colors in the Standard Chart Color Palette. This palette consists of a legacy color palette (pre 8.4 color palette) and a default color palette as shown below:

![Legacy color palette](image)

The legacy color palette consists of the first six rows in the color palette.
The standard color palette consists of the last two rows. There are seventeen chart default colors and the chart background color is the last color on the bottom row.

The numbers to the right of the color scheme are the hexadecimal numbers for the rgb codes in the standard color palette.

**Using Legacy Chart Colors**

When you open a chart created in Release 8.3 and earlier, the chart uses the default colors available in the legacy color palette. This feature is enabled automatically for all 8.3 and earlier charts when the Legacy check box is enabled on the Choose Color Scheme dialog box. In addition, when this option is checked, all newly added charts use the default colors in the Legacy Color Palette that were used in 8.3 and earlier. When the Legacy option is unchecked, which is a default value, all newly added charts use the default colors in the Standard Color Palette. This value is saved with other application level parameters.

**Note:** Legacy chart colors cannot be applied to Bubble and Scatter charts.

- To apply a legacy chart color scheme:
  1. **Select the chart, and on the short cut menu, select Properties.**
     - The Chart Properties dialog box is displayed.
  2. **On the General tab, select Color Scheme for Data Points...**
     - The Choose Color Scheme dialog box is displayed.
  3. **Select Legacy and click OK.**

   **Note:** The color scheme for the legacy chart palette cannot be changed. To change the color scheme, select “Custom” instead of Legacy.

If you modify the chart (for example, you perform a drill down, or change the chart type), the colors used are the default color sets depending on the Legacy option. For example, if the check box is not checked, the new default colors are used, otherwise, the old default colors (pre 8.4) are used.

In the example below, the bar chart uses the default colors from the legacy color scheme:
Using Standard Chart Colors

By default, charts created globally in Release 8.4 and later use the standard colors available in the Standard Color Palette. In this case, the Legacy option check box is unchecked on the Choose Color Scheme dialog box.

To apply a standard chart color scheme:

1. **Select the chart, and on the shortcut menu, select Properties.**
   
   The Chart Properties dialog box is displayed.

2. **On the General tab, select Color Scheme for Data Points...**
   
   The Choose Color Scheme dialog box is displayed.

3. **Select Standard and click OK.**

   **Note:** The color scheme for the standard chart palette cannot be changed.

In the following example, the bar chart uses the colors from the standard color scheme:
Using a Custom Chart Color Scheme

If you want to use a chart color scheme other than those offered in either the Legacy or Standard Chart color schemes, you can design and apply a custom chart scheme. A custom chart color scheme persists only to the chart to which it has been applied.

To create a custom color scheme:

1. Select the chart, and on the shortcut menu, select Properties.
   The Chart Properties dialog box is displayed.
2. On the General tab, select Color Scheme for Data Points...
   The Choose Color Scheme dialog box is displayed.
3. Select Custom.
4. From the color palette, double-click the chart color scheme item to change.
   An extended color palette is displayed.
5. From the extended color palette, select the color to apply for the color scheme, and click OK.
   You can change any of the seventeen colors displayed on the Choose Color Scheme dialog box.

In the following example, a custom color scheme has been selected for the bar chart:

Changing the Color of Chart Elements, Lines, and Text

Use the Format toolbar to change the line, fill, and text color of a chart element or text and add emphasis to your chart.

Note: If the Format toolbar is not visible, select View, then Toolbars, and then Formatting.
To change line color:
1 Select a plot point in the line.
2 On the Format toolbar, open the Line Color list and select a color from the palette.

To change the fill color of a chart element:
1 Select the chart element.
2 On the Format toolbar, open the Fill Color list and select a color from the palette.

To change the color of text:
1 Select a label or other text element.
2 On the Format toolbar, open the Text Color list and select a color from the palette.

**Chart Menu Command Reference**

The following table provides a quick reference to the commands available on the Chart menu and lists any related shortcuts.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Keyboard Shortcut</th>
<th>Shortcut Menu</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sort</td>
<td>Sorts the selected column values in ascending order (alphabetical or numeric).</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Data Function</td>
<td>Applies a prebuilt data function to the selected item.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add Computed Item</td>
<td>Enables you to add a new data item derived from calculations performed on an item.</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Modify Computed Item</td>
<td>Enables you to modify a computed item.</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Remove Selected Items</td>
<td>Removes the selected items.</td>
<td>Del</td>
<td>*</td>
</tr>
<tr>
<td>Drill Anywhere</td>
<td>Enables you to drill to any item.</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Drillup</td>
<td>Returns the original view of the data you drilled.</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Focus On Items</td>
<td>Updates the chart to include only the selected items.</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Hide Item</td>
<td>Hides the selected item from view.</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Show Hidden Items</td>
<td>Restores the selected hidden item.</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Show All Items</td>
<td>Updates the chart to include all items removed by focusing.</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Group Items</td>
<td>Groups the selected items.</td>
<td>Ctrl+G</td>
<td>*</td>
</tr>
<tr>
<td>Ungroup Items</td>
<td>Ungroup the selected grouped item.</td>
<td>Ctrl+U</td>
<td>*</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
<td>Keyboard Shortcut</td>
<td>Shortcut Menu</td>
</tr>
<tr>
<td>---------------</td>
<td>------------------------------------------------------------------------------</td>
<td>-------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Restore Name</td>
<td>Restores the original name of a renamed item.</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Refresh Data</td>
<td>Updates the data according to the selected option. Select between After Process, When Section Displayed, Manually, or Refresh Now.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Properties</td>
<td>Opens the Properties dialog box for the chart or selected chart element.</td>
<td></td>
<td>*</td>
</tr>
</tbody>
</table>

**General Chart Properties**

General chart properties control the display of various object, such as titles, legends, and borders in the Chart area. They are also used to specify default plane and rotation settings.

General chart properties control the display of various objects, such as titles, legends, and borders, in the Chart area. They are also used to specify: scaling, scrolling, sizing, zooming, default plane and rotation settings.

- To adjust general chart properties:
  1. **Select Chart, then Properties.**
     The Properties dialog box is displayed.
  2. **Click the General tab to display the General properties page.**
  3. **Change any of the following properties:**
     - Show title
     - Show subtitle
     - Show legend
     - Show border.
     - Show partial view indicator,
     - 3-D objects
     - Print/PDF all views
     - Auto Resize
     - Smart Scaling
     - Display all text with smart scaling—
     - Minimum Font Size
     - Color Scheme for Data Points....
     - Planes
     - Rotation Auto
     - Horizontal/Vertical rotation degrees
     - Rotation
- Auto Layout
- Legend position

4. Select OK.

<table>
<thead>
<tr>
<th>Table 64 General Chart Properties</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Option</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>Show title</td>
<td>Toggles the display of the chart title as entered in the text field.</td>
</tr>
<tr>
<td>Show subtitle</td>
<td>Toggles the display of a subtitle as entered in the text field.</td>
</tr>
<tr>
<td>Show legend</td>
<td>Toggles the display of the color-coded chart legend in the Contents pane.</td>
</tr>
<tr>
<td>Show border</td>
<td>Toggles the display of a border around the chart. The border controls the general position of all other Chart components not controlled by the Chart graphic border. This border can be resized and repositioned.</td>
</tr>
<tr>
<td>Show partial view indicator</td>
<td>Toggles the display of the “Partial View” indicator in the top left corner of the Contents pane. The indicator shows when not all bars are displayed in the current view. The indicator is displayed on all output media, including the Interactive Reporting Studio, EPM Workspace, PDF and print.</td>
</tr>
<tr>
<td>3-D objects</td>
<td>Toggles the three dimensional display of a chart object, including pie slices, chart bars, side planes, and the perspective view.</td>
</tr>
<tr>
<td>Print/PDF all views</td>
<td>Toggles the print of the entire rendered chart. The setting you specify here is not applied to the Export to PDF option in the EPM Workspace.</td>
</tr>
<tr>
<td>Auto Resize</td>
<td>Toggles the resizing of the graphic proportions within the Contents pane automatically whenever the application window is resized.</td>
</tr>
<tr>
<td>Smart Scaling</td>
<td>Toggles the Smart Scaling feature, which governs how Chart components are initially displayed, how they behave during Chart resizing and how the Chart components can be moved or repositioned.</td>
</tr>
<tr>
<td>Display all text with smart scaling</td>
<td>Toggles the do not remove text. When this feature is enabled, Smart Scaling (see above) does not remove any text (it does not clip or overlap text if an object is resized or repositioned to a minimum font size). This feature is checked for new Chart sections, and unchecked for 8.3 releases and earlier that have Smart Scaling enabled. Inserted text and legends are not affected by your selection in this field. Smart Scaling may shrink these objects, but they are never scaled to the point where they are eliminated entirely.</td>
</tr>
<tr>
<td>Minimum Font Size</td>
<td>Toggles the minimum font size feature for text labels when the overall Chart size is changed and Smart Scaling (see above) is enabled. If it is not possible to fit a component in the available space, the font size is reduced by one point and a redraw is attempted. The font size reduction process repeats until the component can fit within the component’s boundary or until the font size specified here is reached. If the component is still too large to fit without obstructing another object, or it is obstructed, then the component is removed from the display. The default minimum font size is 8.</td>
</tr>
<tr>
<td>Color Scheme for Data Points....</td>
<td>Launches the Choose Color Scheme dialog box from which you can select the chart color scheme for data points. Available color schemes are:</td>
</tr>
<tr>
<td></td>
<td>- Standard—default color scheme used for charts created in Release 8.4 and later.</td>
</tr>
<tr>
<td></td>
<td>- Legacy—color scheme used for charts in Release 8.3 and earlier</td>
</tr>
<tr>
<td></td>
<td>- Custom—color scheme defined by the user.</td>
</tr>
<tr>
<td></td>
<td>To enable all charts to use either a standard, legacy or custom color scheme, select the desired color scheme by choosing Color Scheme Data option on the Default Fonts and Styles box for chart.</td>
</tr>
<tr>
<td></td>
<td>Also see, “Applying Chart Color Schemes for Data Points” on page 256</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| **Planes**           | Show horizontal plane - Toggles the display of the horizontal plane.  
                      | Show vertical plane - Toggles the display of the vertical plane.  
                      | Show back plane - Toggles the display of the back plane.          |
| **(Rotation) Auto**  | Enables the default rotation for new bar charts to 3° horizontally by 2° vertically when a component is added to the X axis. If a component is added to the Z axis, a rotation of 15° horizontally by 20° vertically is used. If this option is disabled, the rotation specified in the Horizontal \ Vertical rotation degrees fields. |
| **Horizontal degrees/ Vertical degrees** | Horizontal degrees / Vertical degrees—Shifts the perspective angle or elevation (-60 to 60 degrees) for all chart types except pie, which has its own rotation, and is only visible when the chart type is set to pie. |
| **Rotation (Pie Charts)** | Pie Rotation—Rotates the pie (-90 to 90 degrees).  
                      | Pie Height—Modifies the pie height (0 to 90 degrees).          |
| **Auto Layout**      | Enables the layout of scatter and bubble chart components automatically. In this case, scatter and bubble chart components cannot be repositioned manually. |
| **Legend position**  | Sets the position of the legend in scatter and bubble charts. Available positions include:  
                      | • Top  
                      | • Left  
                      | • Bottom  
                      | • Right  
                      | This option is especially usefully when the Auto Layout property has been enabled. |

### Chart Labels Axis Properties

Chart labels axis properties control the display for the X Axis and Z-axis labels, tickmarks, and values for items in the data layouts.

- To adjust chart label axis properties:
  1. Select Chart, then Properties.
  2. Select the Label Axis tab.
  3. Modify the properties shown as desired and click OK to apply your changes.

<table>
<thead>
<tr>
<th>Table 65 Chart Labels Axis Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Option</strong></td>
</tr>
<tr>
<td>(Axis dropdown)</td>
</tr>
<tr>
<td>Show axis label</td>
</tr>
<tr>
<td>Show drill path in labels</td>
</tr>
<tr>
<td>Option</td>
</tr>
<tr>
<td>--------------</td>
</tr>
<tr>
<td><strong>Time Aware</strong></td>
</tr>
<tr>
<td><strong>Auto range</strong></td>
</tr>
<tr>
<td><strong>Min</strong></td>
</tr>
<tr>
<td><strong>Max</strong></td>
</tr>
<tr>
<td><strong>Auto frequency</strong></td>
</tr>
</tbody>
</table>
| **Show tickmark** | Check to show tickmarks. To manually specify recurring intervals at which to display a limited number of tick marks, enter the numeric value in the At every edit box. If you are working with a Time Aware axis, you can place the tickmark on the:  
  * Value  
  * Year  
  * Month  
  * Week  
  * Day  
  * Hour  
  * Minute  
  * Second |
| **Show label** | Check to show labels. To manually specify recurring intervals at which to display a limited number of labels, enter the numeric value in the At every edit box. If you are working with a Time Aware axis, you can specify to place the label on the:  
  * Value  
  * Year  
  * Month  
  * Week  
  * Day  
  * Hour  
  * Minute  
  * Second |
| **Expand label box** | Increases the label box size for an x axis label (y and z labels are unaffected) in an individual Chart. It is not available for pie charts. This feature is used primarily for sequential labels, for example, time, day or number labels. For regular name-based labels, this feature should be disabled so that the user can view each label name. |
### Chart Values Axis Properties

Chart values axis properties control the display of axis labels, tickmarks, values, and position for the Facts category in the data layout.

To adjust chart value axis properties:

1. Select **Chart**, then **Properties**.
2. Select the **Values Axis** tab.
3. Modify the properties as desired and click **OK**.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>X Axis, Y Axis</strong></td>
<td>Sets the axis (X or Y) for the axis scale parameter (see below). This option is available for scatter and bubble charts only.</td>
</tr>
<tr>
<td><strong>Show left axis label</strong></td>
<td>Toggles the display of a descriptive axis label as entered in the text field for the main value axis.</td>
</tr>
<tr>
<td><strong>Show right axis label</strong></td>
<td>Toggles the display of a descriptive axis label as entered in the text field for the secondary value axis.</td>
</tr>
<tr>
<td><strong>Show tick marks at intervals</strong></td>
<td>Toggles the display of tick marks along the axis.</td>
</tr>
<tr>
<td><strong>Show values at intervals</strong></td>
<td>Toggles the display of numeric scale values along the axis.</td>
</tr>
<tr>
<td><strong>Show values at right</strong></td>
<td>Toggles the display of numeric scale values at the right.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Logarithmic Scale</td>
<td>Plots the line chart with a logarithmic (base 10) values axis.</td>
</tr>
<tr>
<td>Baseline</td>
<td>Enable to shift the position of plotted data in the chart based on a baseline (threshold) amount. Data above the baseline amount appear in the top sector of the diagram separated by a break line from those items below the baseline.</td>
</tr>
<tr>
<td>(Baseline Amount)</td>
<td>Specify the baseline amount.</td>
</tr>
<tr>
<td>Axis Scale (Left and Right)</td>
<td>Set the axis scale between selected minimum and maximum values, or click Auto to auto-assign the scale.</td>
</tr>
<tr>
<td>Interval</td>
<td>Enter an interval value to separate axis values, or click Auto to auto-assign the interval.</td>
</tr>
</tbody>
</table>

## Bar Chart Properties

Bar chart properties control the general attributes for the various bar chart formats, including:

1. To adjust bar chart properties:
   1. Select **Chart**, then **Properties**.
   2. Select the **Bar Chart** tab.
   3. Modify the properties as desired, and click **OK**.

### Table 67  Bar Chart Properties

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show values on bars</td>
<td>Toggles the display of data values for all bars.</td>
</tr>
<tr>
<td>Cluster bars by items in &quot;Y&quot; axis</td>
<td>Uses the item assigned to the y axis to define a bar cluster. For example, if you have &quot;Quarter&quot; on the y axis, each cluster consists of four bars. (clustered bar charts only).</td>
</tr>
<tr>
<td>Cluster bars by items in &quot;Z&quot; axis</td>
<td>Uses the item assigned to the z axis to define a bar cluster. For example, if you have &quot;Product Line&quot; on the z axis, each cluster consists of a bar for each product line in your data. (clustered bar charts only).</td>
</tr>
</tbody>
</table>

### Table 68  Bar Chart Properties

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show values on bars</td>
<td>Toggles the display of data values for all bar lines.</td>
</tr>
<tr>
<td>Show line values</td>
<td>Toggle the display of numeric values on each bar line.</td>
</tr>
<tr>
<td>Ignore null values for lines</td>
<td>Ignores null values by not plotting a line to a marker representing null values in the Bar Line or Line chart. Null values can disrupt the appearance of values in a line chart. Zero’s (0) are included.</td>
</tr>
<tr>
<td>Shift Points to Left</td>
<td>Places line chart plot points at left of bars (bar-line charts only).</td>
</tr>
<tr>
<td>Shift Points to Center</td>
<td>Places line chart plot points centered on bars (bar-line charts only).</td>
</tr>
<tr>
<td>Stack bars along &quot;Y&quot; axis</td>
<td>Uses the item assigned to the &quot;Y&quot; axis to define a stacked bar line. For example, if you have &quot;Quarter&quot; on the &quot;Y&quot; axis, each stacked bar line consists of two lines.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Cluster bars along &quot;Y&quot; axis</td>
<td>Uses the item assigned to the &quot;Y&quot; axis to define a bar line cluster. For example, if you have &quot;Quarter&quot; on the &quot;Y&quot; axis, each cluster consists of four lines.</td>
</tr>
</tbody>
</table>

### Area and Ribbon Chart Patterns Properties

Use Pattern tab to select the design of your Area and Ribbon charts.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatic</td>
<td>Sets the default fill pattern for the Area or Ribbon chart. The default pattern is solid.</td>
</tr>
<tr>
<td>None</td>
<td>Sets a transparent fill pattern for the Area or Ribbon chart. Only the borders of the Area or Ribbon chart element are displayed.</td>
</tr>
<tr>
<td>Custom</td>
<td>Enables you to select your own fill pattern for the Area or Ribbon chart. Valid patterns can be selected from the Fill Pattern pull-down list.</td>
</tr>
<tr>
<td>Fill Pattern</td>
<td>Sets the fill scheme of the Area or Ribbon chart, such as solid, hollow or horizontal. To select a different style, click another fill pattern from the pull-down list.</td>
</tr>
<tr>
<td>Foreground</td>
<td>Displays the part of the area or bar chart view nearest to the user that will be represented.</td>
</tr>
<tr>
<td>Preview</td>
<td>Displays the pattern as it will be displayed to the user when the settings have been applied.</td>
</tr>
</tbody>
</table>

### Bar Chart Patterns Properties

Use the Bar Chart Patterns tab to apply a pattern to a bar chart element.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatic</td>
<td>Sets the default fill pattern for the bar chart. The default pattern is solid.</td>
</tr>
<tr>
<td>None</td>
<td>Sets a transparent fill pattern for the bar chart. Only the borders of the chart element are displayed.</td>
</tr>
<tr>
<td>Custom</td>
<td>Enables you to select your own fill pattern for the bar chart. Valid patterns can be selected from the Fill Pattern pull-down list.</td>
</tr>
<tr>
<td>Fill Pattern</td>
<td>Sets the fill pattern of the bar chart, such as solid, hollow or horizontal. To select a different style, click another fill pattern from the pull-down list.</td>
</tr>
<tr>
<td>Foreground</td>
<td>Toggles the display the part of the bar chart view nearest to the user that will be represented.</td>
</tr>
<tr>
<td>Preview</td>
<td>Displays the pattern as it will be displayed to the user when the settings have been applied.</td>
</tr>
</tbody>
</table>
Bar Chart Data Labels Properties

Use the Data Labels tab to toggle data labels on bar chart elements.

Table 71  Bar Chart Data Labels Properties

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>Suppresses the display of the numeric value of each bar chart element in a data label.</td>
</tr>
<tr>
<td>Show values</td>
<td>Displays the numeric value of each bar chart element in a data label.</td>
</tr>
<tr>
<td>Inside top</td>
<td>Inserts the data label at the top of the bar.</td>
</tr>
<tr>
<td>Inside middle</td>
<td>Inserts the data label in the middle of the bar.</td>
</tr>
<tr>
<td>Inside bottom</td>
<td>Inserts the data label at the bottom of the bar.</td>
</tr>
</tbody>
</table>

Line Chart Patterns Properties

Use the Line Chart Patterns tab to apply a pattern to elements and markers.

Table 72  Line Chart Patterns Properties

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatic</td>
<td>Sets the default color, width and style pattern for the line.</td>
</tr>
<tr>
<td>None</td>
<td>Sets a transparent color, width and style for the line.</td>
</tr>
<tr>
<td>Custom</td>
<td>Enables you to select your own color, width and style pattern.</td>
</tr>
<tr>
<td>Color</td>
<td>Sets the color of the line. To select a different color, click the color bar. A grid of colors is displayed so that you can select another color for the line.</td>
</tr>
<tr>
<td>Width</td>
<td>Sets the width (weight) of the line. To select a different width, click another point size from the pull-down list.</td>
</tr>
<tr>
<td>Style</td>
<td>Sets the style of the line, such as a solid, dotted or dashed line. To select a different style, click another style from the pull-down list.</td>
</tr>
<tr>
<td>Preview</td>
<td>Displays the line or marker pattern as it will be displayed to the user when the settings have been applied.</td>
</tr>
</tbody>
</table>

Table 73  Line Chart Patterns Properties

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatic</td>
<td>Sets the defaults color, size, border and fill color pattern for the marker.</td>
</tr>
<tr>
<td>None</td>
<td>Sets a transparent color, width and style for the line for the marker.</td>
</tr>
<tr>
<td>Custom</td>
<td>Enables you to select your own color, width and style pattern for the marker.</td>
</tr>
<tr>
<td>Style</td>
<td>Sets the symbol for the marker, such as diamond-shaped, circular, rectangular or triangular. To select a different style, click a style from the Style pull-down list</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Size</td>
<td>Sets weight of the marker. To select a different weight, click a point size from the another point size from the pull-down list.</td>
</tr>
<tr>
<td>Border Color</td>
<td>Sets the color of the marker’s borders. To select a different color, click the color bar. A grid of colors is displayed so that you can select another color for the border.</td>
</tr>
<tr>
<td>Fill Color</td>
<td>Sets the fill color of the marker. To select a different color, click the color bar. A grid of colors is displayed so that you can select another fill color.</td>
</tr>
</tbody>
</table>

### Line Chart Axis Properties

Use the Axis tab to plot values on the primary or secondary axis.

**Table 74  Line Chart Axis Properties**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary (left) axis</td>
<td>Toggles to plot (locate) values based on the primary or left axis.</td>
</tr>
<tr>
<td>Secondary (right) axis</td>
<td>Toggles to plot (locate) values based on the secondary or right axis.</td>
</tr>
</tbody>
</table>

### Line Chart Data Labels Properties

Use the Data Labels tab to toggle data labels for line chart elements.

**Table 75  Line Chart Data Label Properties**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show values</td>
<td>Displays the numeric value of each bar chart element in a data label.</td>
</tr>
<tr>
<td>Ignore Null Values</td>
<td>Toggle to include null values in the chart. Null values disrupt the line, area or ribbon.</td>
</tr>
<tr>
<td>Above</td>
<td>Inserts the data label above the marker.</td>
</tr>
<tr>
<td>Below</td>
<td>Inserts the data label below the marker.</td>
</tr>
<tr>
<td>Left</td>
<td>Inserts the data label to the left of the marker.</td>
</tr>
<tr>
<td>Right</td>
<td>Inserts the data label to the right of the marker.</td>
</tr>
</tbody>
</table>

### Line Chart Properties

Line chart properties control the general attributes of the line chart.
Table 76  Line Chart Properties

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ignore Null Values</td>
<td>Toggle to include null values in the chart. Null values disrupt the line, area or ribbon.</td>
</tr>
<tr>
<td>Fill area under ribbon</td>
<td>Toggles between ribbon and area charts, creating the latter by filling the area under a ribbon. This option is activated by default.</td>
</tr>
</tbody>
</table>

Pie Chart Patterns Properties

Use the Pie Chart Patterns tab to apply a pattern to a pie chart slice.

Table 77  Pie Chart Patterns Properties

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatic</td>
<td>Sets the default fill pattern for the slice. The default pattern is solid.</td>
</tr>
<tr>
<td>None</td>
<td>Sets a transparent fill pattern. Only the borders of the chart element are displayed.</td>
</tr>
<tr>
<td>Custom</td>
<td>Enables you to select your own fill pattern. Valid patterns can be selected from the Fill Pattern pull-down list.</td>
</tr>
<tr>
<td>Fill Pattern</td>
<td>Sets the fill pattern of the pie chart, such as solid, hollow or horizontal. To select a different style, click another fill pattern from the pull-down list.</td>
</tr>
<tr>
<td>Foreground</td>
<td>Toggles the display the part of the chart view nearest to the user that will be represented.</td>
</tr>
<tr>
<td>Preview</td>
<td>Displays the pattern as it will be displayed to the user when the settings have been applied.</td>
</tr>
</tbody>
</table>

Pie Chart Data Labels Properties

Use the Data Labels tab to toggle data labels on each pie chart element.

Table 78  Pie Chart Data Label Properties

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show labels</td>
<td>Toggles the display of pie slice labels.</td>
</tr>
<tr>
<td>Show values</td>
<td>Toggles the display of the numeric value of each pie slice.</td>
</tr>
<tr>
<td>Show percentages</td>
<td>Toggles the display of the percentage value of each pie slice in a selected number format.</td>
</tr>
<tr>
<td>Show lines to labels</td>
<td>Toggles the display a line running from a data label to a selected pie slice.</td>
</tr>
<tr>
<td>Show negative values as positive</td>
<td>Toggles the representative display of negative values as pie slices, as opposed to not including them.</td>
</tr>
</tbody>
</table>

Bubble Chart Properties

Use the Bubble Chart Properties tab to set specific bubble chart options.
To set bubble chart properties:

1. **Select Chart, then Properties.**
   
The Properties dialog box is displayed.

2. **Select the Bubble tab.**

3. **Modify the properties as desired and click OK.**
   
   - **Show negative bubble**—Display bubbles with negative values. These type of values are derived from their real absolute value, and the real negative value is depicted in the data label (although based on the positive value).
   
   - **Show zero bubbles**—Display bubbles with zero size values as a visual reference.
   
   - **Max bubble size (%)**—Set the maximum size of the bubble in the grid area. This size is measured as a percentage value. The default value is 25%.

### Bubble Chart Patterns Properties

Use the Bubble Chart Patterns dialog to select the border color and fill color for data points represented as a bubble chart.

To set bubble chart pattern properties:

1. **Select Chart, then Properties.**

2. **Click the border color and select a color from the color palette.**

3. **Click the fill color and select a color from the color palette.**

4. **Click OK.**

### Scatter Chart Patterns Properties

Use the Scatter Chart Patterns Properties dialog to select the style, size and fill colors for data points in the scatter chart.

To set scatter chart pattern properties:

1. **Select Chart, then Properties.**

2. **Select the style of the data point:**
   
   - **Diamond**
   
   - **Square**
   
   - **Circle**
   
   - **Triangle**

3. **Select the point size of the data point.**
   
   Available points are between 1–6.
Click the fill color and select a color from the color palette.

Click OK.

Set Legend On

A chart legend can be placed on different axes.

Select the axis on your chart on which you want to set the legend and click OK.

Colors are redistributed to highlight the data associated with the selected item and color coordinates corresponding labels in the Legend.

Plot Area

The area bounded by the axes is called the plot area. In the case of a pie chart, the plot area is defined by a circle representing the totality of all data items.

Chart Bar

A linear measure of a value used in bar charts.

Axis Labels

Request items, which represent dates or names, are usually assigned to one of the label panes. Labels are the scalar axis definitions that reference the information in your chart. Depending on whether you create a two-dimensional or three-dimensional chart, you assign other data to one or two labels panes.

For example, assume you create a pie chart to see the proportional sales totals for all your regional sales managers. After adding the sales dollars to the values pane to create the pie chart, assume you divide it in slices by placing the employee name in the x pane. In this example, the employees are the axis labels.

The panes at the top right is the z pane, and the pane at the bottom right is the x pane. Depending on the type of chart you create, you can use one or both of these panes, or the panes could switch around with the data values panes.

Stacking Categories within Data Labels

Stacked bar charts are an excellent way to depict how categories "stack up" against each other. For example, assume you want to see the sales by quarter for each product. To do this, you could place the product type on the x axis and use a vertical stack to illustrate a breakdown of sales by
To stack label categories in a bar chart:

1. Select Vertical Stacked bar or Horizontal Stack bar from the Chart Type pull-down list.
2. Select the target category to subdivide in the vertical or horizontal stack.
3. Drag a value item from Elements to the z pane in the data layout.
4. Drag a label item from Elements to x pane in the data layout.
5. Drag the targeted item for the stacking from Elements to y pane in the data layout.

The chart creates stacks for each category of the targeted item. The height of all the stacks represents the grand total of all the charts.

**Stacking Categories of Numeric Data**

You can stack numeric categories of data on top of each other to illustrate their relative size. For example, a stack can illustrate the relative size of retail prices with respect to wholesale prices. By stacking these figures on top of each other, you can analyze their respective size.

*Note:* Data values stacked differently than data labels because sub-divisions in labels can be meaningfully represented while values cannot. For example, you can the product year by 2001 and 2002 and represent each year as a stack. However, data values (such as Amount of sales) cannot be meaningfully subdivided in the same way. With data values, it is more meaningful to compare one value item (such as amount of sales) with another value item (such as units sold).

**Vertical Bar Charts**

Bar charts compare items within a set. You can create vertical bar charts in two or three dimensions, and you can cluster or stack the bars. In a vertical bar chart, items in the y pane form the chart bars, and items in the x and z panes itemize the bars. Vertical bar charts are the default chart type.
Horizontal Bar Charts

Bar charts compare items within a set. You can create horizontal bar charts in two or three dimensions, and you can cluster or stack the bars. In a horizontal bar chart, items in the x-categories pane form the chart bars, and items in the y pane and z pane itemize the bars.
Report Section

In the Report section, the Report Designer helps you to easily develop a complete range of reports, from small ad-hoc reports to mission-critical operational reports. After you create your database query, you can use this visual layout capabilities of the section to drag and drop columns, expressions, charts, pivot tables, logos, and graphic objects to quickly design and customize your reports.

Report Section Elements

The Report section differs slightly from other sections in order to provide you with as robust a report designer as possible. Key differences include:

- **Expression Line**—Accessed by way of the Expression button on the Section title bar, the Expression line enables you to build common computed expressions using JavaScript.
- **Elements**—Contains all of the drag-and-drop elements that you use to create a custom report, including:
  - **Query**—Contains all the sections associated with a selected query, including Results, Pivot, and Chart sections. You can drag individual Results columns from the Results and Table sections to a report page, as well as entire Pivot and Chart sections to create *Smart Reports*. 

In This Chapter

Report Section ........................................................................................................... 275
Creating a Custom Report.......................................................................................... 280
Working with a Report Page..................................................................................... 292
Setting Up a Report.................................................................................................... 293
Enhancing Report Data.............................................................................................. 296
Using Multiple Data Sources in a Report................................................................. 303
Creating Smart Reports............................................................................................. 304
Formatting Report Items........................................................................................... 304
Converting Detail Reports from Versions Earlier than 6.0........................................ 305
Report Menu Command Reference........................................................................... 307
Graphics—Contains standard vector graphics text labels, and pictures (bitmaps only). Vector graphics include: line, horizontal line, vertical line, rectangles, round rectangles, and ovals. For vector graphics, select Report, then Insert Graphics. For a .BMP, .JPEG, .GIF, .JPG or .PNG, select Report, then Picture.

When an Interactive Reporting document (BQY) is created and saved using 9.3.x releases, all images are saved in centralized format (the Resource Manager). As a result, the images cannot be read or displayed in prior releases of Interactive Reporting. This is because the relocation and rationalization of the images are not understood by releases prior to 9.3.x.

The exception is images from a document saved in a pre 9.3 Release, which have been referenced once: not copied, merged or duplicated, that is, placed in Dashboard or Report workspace as a Picture graphic in traditional fashion.

To retain compatibility with earlier releases, you can run the RevertImageResources script. This script provides a facility to undo the relocation and image merging, and thereby return the Interactive Reporting document to the pre-9.3 pre-Resource Manager format. This script is included with the 9.3. Release of Dashboard Development Services. It can also be downloaded from the Hyperion Developer Network. This script can only be used on the desktop, and not in the EPM Workspace, as it relies on the COM feature of Interactive Reporting Studio.

Certain new features make documents ineligible for conversion to the non-Resource Manager format: for example if the 9.3 Interactive Reporting document has Bubble and Scatter charts, then these are lost when the Interactive Reporting document is opened in the older version of Interactive Reporting Studio. For information on running the RevertImageResources script, see the Hyperion System 9 BI+ 9.3. Readme.

Fields—Contains predefined fields that can be dragged to various areas of the report to enhance the look and feel of the report. Fields include items such as date and time, the report name, or page number.

To use any of these elements, simply drag them to the desired report component band.

Data Layout—Consists of the Groups data layout and Table data layout. The Table data layout is divided into the Dimension and Fact panes.

Groups Data Layout—Defines the overall or highest levels used to group data in a report. When you designate an item to serve as a group header (also known as a break value), you are instructing Interactive Reporting to organize the rest of the data in repeating collections of records according to the group header.

Dimensions pane in Table Data Layout—Includes descriptive information such as a column in a table that is included in the body of the report.

Facts pane in Table Data Layout—Includes the measurable or quantifiable data as a column in a table included in the body of the report. Interactive Reports calculates and inserts subtotals for each fact column.

Interactive Reporting quantifies values by group header and dimension. If you have a descriptive numeric value that should not be calculated, such as Retail Price or Target Sales, use it as a group header or table dimensions instead of a fact.
Report Section Toolbar

The Report Section toolbar provides icons that enable you to quickly maneuver multiple report objects.

- **Align**—Aligns several objects at the same time. Objects are aligned to the first object you select. Select the first object, then hold down the Control key and select the remaining objects. Click the arrow on the Align icon and select an alignment option: left, center, right, top, middle, or bottom.

- **Make Same Size**—Resizes the selected objects to the same size. Objects are resized to match the first object you select. Select the first object, then hold down the Control key and select the remaining objects. Click the arrow on the Make Same Size icon and select a resizing option: width, height, or both.

- **Layer**—Stacks a single object in relative position to other objects. Layer includes four rearrangement options: Send To Front, Send To Back, Bring Forward, and Send Backward. Use this feature to layer multiple objects so that only the sections of the objects you want visible are shown.

- **Spring Objects**—Maintains relative vertical spacing between dynamic objects. That is, you can spring one object to another so that if the first object is moved, increased or diminished, the second object moves in the same flow. To select an object, select the object, hold down the Control key and select the remaining objects. Click the Spring Objects icon to spring the objects. To remove spring from objects, select the objects, and click Spring Objects again.

- **Zoom**—Sets the magnification level of the report. Options include: whole page, page width, or a percentage of magnification based on 100%.

<table>
<thead>
<tr>
<th>Tool</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Rulers" /></td>
<td>Rulers</td>
<td>Horizontal and vertical rulers help you line up items based on precise units of measure. To select the ruler measurement, click the measure indicator at the intersection of the rulers. You can select inches, centimeters, or pixels. To toggle the ruler, select Report, then Rulers. A check mark next to the Ruler menu item indicates that the ruler is turned on.</td>
</tr>
<tr>
<td><img src="image" alt="Grid" /></td>
<td>Grid</td>
<td>The layout grid automatically snaps all objects to the closest spot on the grid. To toggle the grid, select Report, Grid. A check mark next to the Grid menu item indicates that the grid is turned on.</td>
</tr>
<tr>
<td><img src="image" alt="Design Guides" /></td>
<td>Design Guides</td>
<td>Design guides are similar to grids in that objects automatically snap to line up with the design guides. Design guides are placed at user-specified locations in the work area. You can use both horizontal and vertical design guides. A design guide is displayed as a blue line across the work area, and you can drag the guide to any new location. If rulers are turned on, you can click and drag design guides from both the horizontal and vertical rulers. To toggle a design guide, select Report, then Design Guides. A check mark next to the Design Guides menu item indicates that design guides are turned on.</td>
</tr>
</tbody>
</table>
Report Section Title Bar

The Report Section title bar consists of single-click buttons that enable you to show or hide a specific Report Designer function. These buttons include:

- **Expression**—Toggles the Expression Builder bar, which enables you to build common computed expressions and JavaScript expressions.
- **Groups**—Toggles the Group data layout, which enables you to specify report categories, such as Quarter or Item description.
- **Sort**—Toggles the Sort bar, which enables you to rank data.
- **Table**—Toggles the Table data layout, which enables you to insert facts and dimensions into report bands.
- **Arrows**—The back and forward arrow keys enables you to move back and forward between sections.

**Tip:** All data layouts and lines on the section title bar can be detached, dragged, floated and docked from any fixed position.

Expression Line

Use the Expression Line to apply and display aggregate functions and computing fields. For each item in a report, you can display its JavaScript syntax and modify it to fit your own needs.

![Expression Line](image)

The Expression line includes the following components:

- **Data Function**—Shows available data functions that can be applied to table columns.
- **X**—Cancels and removes a formula. This icon is displayed only when you enter syntax.
- **+**—Accepts and applies the formula. This icon is displayed only when you enter syntax.
- **Edit bar**—Used to display, enter, and edit a JavaScript expression.

<table>
<thead>
<tr>
<th>Expression Line Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Function</td>
<td>Shows available data functions that can be applied to table columns.</td>
</tr>
<tr>
<td>X</td>
<td>Cancels and removes a formula. This icon is displayed only when you enter syntax.</td>
</tr>
<tr>
<td>+</td>
<td>Accepts and applies the formula. This icon is displayed only when you enter syntax.</td>
</tr>
<tr>
<td>Edit Bar</td>
<td>Used to display, enter, and edit a JavaScript expression.</td>
</tr>
</tbody>
</table>
To toggle the Expression line, select $\Sigma$ Expression.

To add a expression:

1. Select $\Sigma$ Expression. The Expression Line is displayed.
2. Select the item to which to apply the expression.
3. Select a data function from the drop-down, or write your own formula in the Expression Line.
4. Click ✔

**Expression Syntax**

The following table lists the default syntax for Report section items.

<table>
<thead>
<tr>
<th>Report Element</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table Dimension</td>
<td><code>CurrBreak.Value[&quot;Column Name&quot;]</code></td>
</tr>
<tr>
<td></td>
<td>&quot;Column Name&quot; refers to the dimensional column name. The column name can be modified.</td>
</tr>
<tr>
<td>Table Fact</td>
<td><code>Tables[&quot;Section&quot;].Columns[&quot;Column Name&quot;].Sum[currBreak]</code></td>
</tr>
<tr>
<td></td>
<td>&quot;Section Name&quot; refers to either the Results or Table section. &quot;Column Name&quot; refers to the fact column name. Sum refers to the return of underlying values and can also be an applied Data functions such as <em>Avg</em>, <em>Max</em>, and <em>Min</em>.</td>
</tr>
<tr>
<td>Group Label</td>
<td><code>CurrBreak.Value[&quot;Column Name&quot;]</code></td>
</tr>
<tr>
<td></td>
<td>&quot;Column Name&quot; refers to the dimension or fact column name.</td>
</tr>
<tr>
<td>Group Fact</td>
<td><code>Tables[&quot;Section&quot;].Columns[&quot;Column Name&quot;].Sum[currBreak]</code></td>
</tr>
<tr>
<td></td>
<td>&quot;Section Name&quot; refers to either the Results or Table section. &quot;Column Name&quot; refers to the fact column name. To apply a break at the parent level (the highest level), use the following syntax: <code>Tables[&quot;Section Name&quot;] .Columns[&quot;Column&quot;].Sum[currBreak. Parent]</code></td>
</tr>
</tbody>
</table>

**Snapping Together Expression Syntax**

You can concatenate report expression syntax with other JavaScript expressions to customize the content of labels, facts, and dimensions.

For example, if you wanted a group header to show a union of the Total label and the amount, you could insert a field in the Expression line to concatenate Total with the table fact syntax.
Or, you could insert a field to show the sum (or any other data function) of the column divided by the sum of the parent level expressed as a percentage.

```
Format(Tables("Results").Columns("Amount").Sum(currBreak) / Tables("Results").Columns("Amount").Sum(currBreak.parent), "0%")
```

Tip: In a Report section, you often concatenate some text with a value. For example, you might want to see “Total Sales: $1,234.56” in a group footer instead of just the number. However, since you have concatenated text to the beginning of the number, you cannot set the number format in the traditional way (by using the Properties dialog box). Since you have a concatenated object, you need to set the number format using JavaScript.

To set a number format on a concatenated item, use the following syntax:

```
Format(number, mask)
```

where `number` is the value to format and `mask` is the number format mask to apply. Using the example above, your completed expression would be displayed as follows:

```
"Total Sales: " + Format(Tables("Results").Columns("Sales").Sum(currBreak), "$#,##0.00")
```

### Creating a Custom Report

The Report section uses tables as the basic building blocks of custom reports. Tables contain columns of dimensions and facts as determined by the Results items you place in the Dimensions and Facts panes in Table data layout.

1. **Select Insert, then New Report.**
   
   Interactive Reporting creates a new Report section and inserts a blank table in the Body band of the report in the Content pane.

2. **If desired, select Report, then Section Boundaries to view the bands for the report components.**
   
   If the Table data layout is not visible, click Table on the Section title bar to open the Table data layout.

3. **Drag Results items from the Query Section pane to the Table data layout.**
Tip: The Table data layout has two panes – Dimensions and Facts. Use the Dimensions pane to build dimension (label) columns, such as Month, Region, or Product Line. Use the Facts pane to build the facts (numeric values) in the report, such as Amount Sales or Units Sold. Results and Table items added to the Facts pane are totaled automatically.

Report Components

The Report Designer offers a variety of dynamic tools for constructing the report you want to create with all of the components you want it to contain. Understanding how these components perform and how they integrate with Report section elements is fundamental to building a successful report.

Report pages are structured areas, or bands, of information. Each band contains a different report component and can be customized to include other report elements such as graphic objects, predefined fields, computed fields, charts, and pivot tables.

A components of the report are:

- **Body**—Contains one or more tables whose contents are based on the items listed in the Table Dimensions and Table Facts panes of the data layout. Every report has a body that holds a table of data. Tables can be based on the same or different results sets within the document.

- **Report Group Headers**—Categorize data into repeating collections of records organized according to the header band.

- **Report Header/Footer**—Fully customizable summary bands of information. Report headers print only on the very first page of the report. Report footers print only on the very last page.

- **Page Header/Footer**—Contain data that is repeated on every page, such as page numbers. These bands are also fully customizable.

Related Topics

“Adding Report Group Headers” on page 283

“Inserting Report Headers and Footers” on page 284

“Inserting Page Headers and Footers” on page 286

Table Column Formatting Options

The following table lists the column formatting options available in Interactive Reporting.

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Changing the Column Name Heading (Title)</td>
<td>To modify the column title, click Expression on the Section title bar, then select the column title that you want to modify. In the Expression line, type the new title between the quotation marks (“ ”) and click the check mark.</td>
</tr>
<tr>
<td>Options</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Hide/Show Column Name Heading (Title)</td>
<td>To toggle the display of column titles, select Column Titles on the shortcut menu. A check mark is displayed next to Column Titles to indicate that column titles are visible. Select this option again to clear the check mark and hide column titles from view.</td>
</tr>
<tr>
<td>Remove Column</td>
<td>To remove a selected column from the report (and Table data layout), select Remove Column on the shortcut menu. This option is available for dimension and fact columns. Items should be removed with caution as computed items may draw data values from the deleted item.</td>
</tr>
<tr>
<td>Hide/Show Column Total</td>
<td>To toggle the display of column totals, select Show Column Totals on the shortcut menu. A check mark is displayed next to Column Totals to indicate that column totals are visible. Select this option again to clear the check mark and hide column totals from view.</td>
</tr>
<tr>
<td>This option is available only for fact columns.</td>
<td></td>
</tr>
<tr>
<td>Select Column</td>
<td>Click anywhere inside the column.</td>
</tr>
<tr>
<td>Suppress Duplicates</td>
<td>To suppress the duplicate values for a column, select the column and select Format, then Suppress Duplicates. Use this feature if you want to display only the first instance in the column of a duplicate value when individual database records include redundant information. This feature is especially useful if records are associated with the same date, location or customer.</td>
</tr>
<tr>
<td>Text Wrap</td>
<td>To wrap text within a column, select the column and select Format, then Text Wrap.</td>
</tr>
<tr>
<td>Sort Ascending/Descending</td>
<td>To sort column values, click Sort Ascending or Sort Descending on the Table menu, shortcut menu, or standard toolbar.</td>
</tr>
<tr>
<td>Reordering Columns</td>
<td>If you want the columns in the table to be displayed in a different order, drag the item in order of how you want it to be displayed in the Table data layout. You can also select the column in the actual report and drag it to a new position in the report.</td>
</tr>
</tbody>
</table>

**Inserting Additional Tables in a Custom Report**

A custom report may contain multiple tables, each originating from the same or different result sets in the document. Multiple tables in one band are a powerful way to compare values.

1. **To add multiple tables to one report:**
   1. Select Report, then Insert Table.
      
      The cursor changes to a crosshair.

2. In the Body band, left-click and hold your left mouse button to position your cursor where you want to insert the table, and then drag to create the table.

   A numbered table is inserted.

3. **Drag the Results items from the Query sections of the Sections pane to the Table2 data layout.**

   Each new table that you insert has its own corresponding dimensions and facts pane in the Table data layout.
Adding Report Group Headers

A report group is the topmost level at which you can structure data in a report. When you drag an item from Elements into the Groups data layout, Interactive Reporting supplies a report group header band automatically and adds a label inside the band that identifies the group.

For example to show purchases by state, each state would serve as a report group header in the report. You can place multiple items in one header or add a multiple levels of group headers.

This step is not necessary to create a report, but it may help you to better organize the content of the report. You can show additional headings, graphics, and totals in the report group header band.

To create a report group header:

1. If the Groups data layout is not visible, click Groups on the Section title bar to open the Groups data layout.
2. Drag the item from Elements to the Report Group1 field in the Groups data layout.

   The selected item automatically populates the group header.

To modify a report group header:

1. Select Report, then Section Boundaries to display header/body/footer regions.
2. Drag the an item from Elements to the group header band.
3. Resize and format the group header as desired.

   Tip: Modify a report group when you want to show additional headings, graphics, and totals in the group header band.

To add a text label to report group header:

1. Select Report, then Section Boundaries to display header/footer regions.
2. Drag the Text Label tool from the Graphics Folder in Elements to the Group Header.
   
   A Text Label box is displayed.
3. Highlight the text in the Text Label box and type the information that you want to be displayed in the box.
4. Click Properties on the shortcut menu to change the border, background or alignment of the text label.
   
   The Properties dialog box is displayed.
5. Select the tab that corresponds to the style you want to change and click OK.
6. Move the text label by clicking and dragging it to the desired position.

   Tip: Use text labels to identify values shown in a numeric group header.
To repeat report group headers:

1. Select the group header band that you want to repeat.
2. Click Repeat Header on the shortcut menu.

The selected group header displays across pages whenever the body band is broken across pages.

**Tip:** Repeat report groups if the body portion of your report section spills across multiple pages. When you repeat a report group, the group header displays on each body page.

### Adding Report Group Footers

In addition to header bands, Report Groups can also have footer bands.

To turn on a footer band for a group:

1. Select the group in the Group data layout.
2. Right-click on the group to display the shortcut menu.
3. Select Footer from the shortcut menu.

There should now be a footer band for each group band. The footer band is displayed after all sub-group and body bands.

**Tip:** The footer band for a group is a good place to add group totals.

### Inserting Report Headers and Footers

Report headers and footers are treated as normal report areas enabling full customization of the contents, including the introduction of other report elements (Pivots, Charts), text labels and computed fields. Typically Report headers and footers are summarizing bands of information. Report headers print only on the first page of a report. Report footers print only on the last page of a report.

**Tip:** Do not confuse report headers and footers with report *group* headers. Report group headers categorize data into repeating collections of records in a header band, based on the Results columns in the Groups data layout.

To view and create a report header or footer:

1. Select Report, then Section Boundaries to view the bands for the report components.
2. Select Report, then Headers and Footers, then *Option*.

You can view the report header, the report footer, or both. The Report Header band is displayed on the first page of the report. The Report Footer band is displayed on the last page.
To insert a report header, select Report Header.

To insert a report footer, select Report Footer.

To insert both report headers and report footers, select Show Both.

The Report Header band is displayed on the first page of the report. The Report Footer band is displayed on the last page of the report.

3 Type the information to be displayed in the report header or footer, or drag objects from the Catalog pane to the Report Header or Report Footer band.

The selected item populates the report header or footer automatically.

Review the following table for information on how to insert items into the Report Header or Report Footer band.

<table>
<thead>
<tr>
<th>Item to Insert</th>
<th>Steps to Take</th>
</tr>
</thead>
<tbody>
<tr>
<td>text</td>
<td>Select Text Label from the Graphics folder, and drag the text label to the header/footer area.</td>
</tr>
<tr>
<td>lines, rectangles, or oval graphic</td>
<td>Select the graphic from the Graphics folder and drag it to the report header/footer area, or select Report, then Insert Graphic.</td>
</tr>
</tbody>
</table>
| picture graphic                | Select the picture from the Graphics Catalog in Elements and drag it into the report header or footer area. 
You can also select Report, then Picture, use the left mouse button to draw the area in which you want the picture to be displayed, and select an image from the Select Image dialog box.
When an Interactive Reporting document (BQY) is created and saved using 9.3.x releases, all images are saved in a centralized format (the Resource Manager). As a result, the images cannot be read or displayed in prior releases of Interactive Reporting. This is because the relocation and rationalization of the images are not understood by releases prior to 9.3.x.
The exception is images from a document saved in a pre 9.3 Release, which have been referenced once: not copied, merged or duplicated, that is, placed in Dashboard or Report workspace as a Picture graphic in traditional fashion.
To retain compatibility with earlier releases, you can run the RevertImageResources script. This script provides a facility to undo the relocation and image merging, and thereby return the Interactive Reporting document to the pre-9.3 pre-Resource Manager format. This script is included with the 9.3. Release of Dashboard Development Services. It can also be downloaded from the Hyperion Developer Network. This script can only be used on the desktop, and not in the EPM Workspace, as it relies on the COM feature of Interactive Reporting Studio.
Certain new features make documents ineligible for conversion to the non-Resource Manager format: for example if the 9.3 Interactive Reporting document has Bubble and Scatter charts, then these are lost when the Interactive Reporting document is opened in the older version of Interactive Reporting Studio. For information on running the RevertImageResources script, see the Hyperion System 9 BI+ 9.3. Readme. |
| Smart Chart/Pivot report       | Select a prebuilt Pivot report or Chart and drag it into the report header, group header, body, or footer area. 
Note the following:
  ● Any Chart/Pivot dropped into a header/footer that is “owned” by data is focused by that piece of data.
  ● An embedded Chart object retains the Chart size, component placement and property settings (Smart Scaling, Auto Resize, minimum font size) of the original chart in the Chart section. Any changes you make to the size of the embedded chart in the Report section is not transferred to the originating chart. |
<p>| empty field (for a computed field) | Select a field from the Fields folder and drag it to the report header, group header, body, or footer area, or select Report then Insert Field. |</p>
<table>
<thead>
<tr>
<th>Item to Insert</th>
<th>Steps to Take</th>
</tr>
</thead>
<tbody>
<tr>
<td>common report values (such as page number or page total)</td>
<td>Select a field from the Fields folder and drag it to the report header or footer area, or select Report, then Insert Predefined Fields and click a field from the drop-down list box. Normally, fields are used in page headers and footer sections.</td>
</tr>
</tbody>
</table>

**Inserting Page Headers and Footers**

Page headers and footers are treated as normal report areas enabling full customizing of the contents, including the introduction of other report elements (Pivots, Charts), text labels and computed fields. Typically, Page headers and footers enable you to specify data that is repeated on every page, such as a page number.

➤ To insert a page header or page footer:

1. **Select Report, then Section Boundaries** to view the bands for the report components.
2. **Select Report, then Headers and Footers, then Option.** You can view a page header, a page footer, or both.
   - To insert a page header, select **Page Header**.
   - To insert a page footer, select **Page Footer**.
   - To insert both page headers and page footers, select **Show Both**.
3. **Type the information to be displayed in the report header or footer, or drag objects from Elements to the Page Header or Page Footer band.** The selected item automatically populates the report header or footer. Review the following table for information on how to insert items into the Page Header or Page Footer band.

*Tip:* You can hide the page header on the first page of a report and show it for the rest of the pages in your report by typing the following code in the “Expression Line” for the page header object:

```java
if (PageNm == 1)
    '{
else
    {"Query Processed: "+Format(newDate(), "d-mmm-yyyy")};
```

**Inserting Page Breaks**

You can insert a page break before or after a report group header.

➤ To toggle page breaks:

1. **Select Report, then Section Boundaries** to view the bands for the report components.
2. **Select a report group header.**
3 Select Insert, then Page Break Before or Page Break After.

A check mark is displayed next to the selected page break option to indicate that it is active. Select this option again to clear the check mark and remove the page break.

<table>
<thead>
<tr>
<th>Item to Insert</th>
<th>Steps to Take</th>
</tr>
</thead>
<tbody>
<tr>
<td>text</td>
<td>Select Text Label from the Graphics folder, and drag the text label to the header/footer area.</td>
</tr>
<tr>
<td>lines, rectangles, or oval graphic</td>
<td>Select the graphic from the Graphics folder and drag it to the report header / footer area, or select Report, then Insert Graphic.</td>
</tr>
<tr>
<td>picture graphic</td>
<td>Select the picture from the Graphics Catalog in Elements and drag it into the report header or footer area. You can also select Report, then Picture, use the left mouse button to draw the area in which you want the picture to be displayed, and select an image from the Select Image dialog box. When an Interactive Reporting document (BQY) is created and saved using 9.3.x releases, all images are saved in a centralized format (the Resource Manager). As a result, the images cannot be read or displayed in prior releases of Interactive Reporting. This is because the relocation and rationalization of the images is not understood by releases prior to 9.3.x. The exception is images from a document saved in a pre 9.3 Release, which have been referenced once: not copied, merged or duplicated, e.g. placed in Dashboard or Report workspace as a Picture graphic in traditional fashion. To retain compatibility with earlier releases, you can run the RevertImageResources script. This script provides a facility to undo the relocation and image merging, and thereby return the Interactive Reporting document to the pre-9.3 pre-Resource Manager format. This script is included with the 9.3. Release of Dashboard Development Services. It can also be downloaded from the Hyperion Developer Network. This script can only be used on the desktop, and not in the EPM Workspace, as it relies on the COM feature of Interactive Reporting Studio. Certain new features make documents ineligible for conversion to the non-Resource Manager format: for example if the 9.3 Interactive Reporting document has Bubble and Scatter charts, then these are lost when the Interactive Reporting document is opened in the older version of Interactive Reporting Studio. For information on running the RevertImageResources script, see the Hyperion System 9 BI+ 9.3. Readme.</td>
</tr>
<tr>
<td>Smart Chart/Pivot report</td>
<td>Select a prebuilt Pivot report or Chart and drag it into the report header, group header, body, or footer area. Note that any Chart/Pivot dropped into a header/footer that is &quot;owned&quot; by data is focused by that piece of data.</td>
</tr>
<tr>
<td>empty field (for a computed field)</td>
<td>Select a field from the Fields folder and drag it to the report header, group header, body, or footer area, or select Report, then Insert Field.</td>
</tr>
<tr>
<td>common report values (such as page number or page total)</td>
<td>Select a field from the Fields folder and drag it to the report header or footer area, or select Report, then Insert Predefined Fields and click a field from the dropdown list box. Normally, fields are used in page headers and footer sections.</td>
</tr>
</tbody>
</table>

Adding Other Report Elements

You can resize report band items and use the tools available on the Format toolbar to change properties such as number, font, alignment, border, and background.

You can also add graphic items and fields from Elements to enhance your report. For example, use the Text Label tool to add text labels that identify values for numeric report group headers. Review the following topics for information on:
Interactive Reporting provides a complete set of graphic elements to assist you in designing presentation-quality reports (see Table 81, “Report Section Graphic Elements,” on page 289). Lines, rectangles and ovals are considered vector graphics. Vector graphics consist of commands for creating the measures and shape of each line, rectangle, arc, and so on. The resolution of the output device defines the look of the vector graphic, since a vector graphic has no fixed resolution. This property enables you to resize the vector graphic without changing its resolution. Since a vector graphic contains only the instructions for creating an image, it requires less disk space.

Pictures are considered bitmap graphics. A bitmap graphic consists of pixels, which have the appearance of small points. In computer memory, a pixel is shown as one or more bits, containing instructions for color, density and appearance for each pixel shown. As a result, bitmaps use more disk space and have a fixed resolution. If you resize a bitmap image, it may distort the original image.

When an Interactive Reporting document (BQY) is created and saved using 9.3.x releases, all images are saved in a centralized format (the Resource Manager). As a result, the images cannot be read or displayed in prior releases of Interactive Reporting. This is because the relocation and rationalization of the images is not understood by releases prior to 9.3.x.

The exception is images from a document saved in a pre 9.3 Release, which have been referenced once: not copied, merged or duplicated, e.g. placed in Dashboard or Report workspace as a Picture graphic in traditional fashion.

To retain compatibility with earlier releases, you can run the RevertImageResources script. This script provides a facility to undo the relocation and image merging, and thereby return the Interactive Reporting document to the pre-9.3 pre-Resource Manager format. This script is included with the 9.3. Release of Dashboard Development Services. It can also be downloaded from the Hyperion Developer Network. This script can only be used on the desktop, and not in the EPM Workspace, as it relies on the COM feature of Interactive Reporting Studio.

Certain new features make documents ineligible for conversion to the non-Resource Manager format: for example, if the 9.3 Interactive Reporting document has Bubble and Scatter charts, then these are lost when the Interactive Reporting document is opened in the older version of Oracle Hyperion Interactive Reporting Studio. For information on running the RevertImageResources script, see the Hyperion System 9 BI+ 9.3. Readme.

To insert a graphic object from Elements:
1. Select Elements and expand the folder containing the desired graphic object.
2. Select the graphic object and drag it to the Contents pane.

To insert a graphic object from the Report menu:
1. Select Report, then Insert Graphic.
2 Select a graphic object from the menu that is displayed.
3 Click the Contents pane to insert the graphic.

<table>
<thead>
<tr>
<th>Table 81 Report Section Graphic Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Element</strong></td>
</tr>
<tr>
<td>Line</td>
</tr>
<tr>
<td>Hz Line</td>
</tr>
<tr>
<td>Vt Line</td>
</tr>
<tr>
<td>Rectangle</td>
</tr>
<tr>
<td>Round Rectangle</td>
</tr>
<tr>
<td>Oval</td>
</tr>
<tr>
<td>Text Label</td>
</tr>
<tr>
<td>Picture</td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Visible</td>
</tr>
<tr>
<td>Locked</td>
</tr>
</tbody>
</table>

To insert a graphic object, use one of the following options:
- Expand the folder in Elements that contains the desired graphic object and drag it to the Contents pane.
- Select Report, then **Insert Graphic**, select a graphic object from the menu that is displayed, and click the Contents pane to insert the graphic.

### Working with Fields

Interactive Reporting provides a computable field and predefined fields to assist you constructing and producing reports (see Table 82, “Report Section Fields,” on page 289). You can drag and drop these fields anywhere within the report page, body, report group header, report header/footer, and page header and footer. You can also customize how numbers, dates, time, and text fonts are displayed in your report. For numbers, you can use commas, decimals place, dollar signs or percentage symbols. For text items, you can select fonts styles and sizes. For Query and Result filters, you can select specific filters.

<table>
<thead>
<tr>
<th>Table 82 Report Section Fields</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Field Name</strong></td>
</tr>
<tr>
<td>Field</td>
</tr>
<tr>
<td>Query Filter</td>
</tr>
<tr>
<td>Field Name</td>
</tr>
<tr>
<td>---------------</td>
</tr>
<tr>
<td>Result Filter</td>
</tr>
<tr>
<td>Query SQL</td>
</tr>
<tr>
<td>Page Number</td>
</tr>
<tr>
<td>Number of Pages</td>
</tr>
</tbody>
</table>
| Page X of Y   | **Note:** All column data types default to *string* unless they are manually changed.  
Note the following behavior of this field when the section is deployed in EPM Workspace:  
• If you intend to use this field in a report deployed in Oracle Hyperion Enterprise Performance Management Workspace, any Report section which is more than one page shows as ‘Page 1 of y’. It continues to show the pages as such (with an unknown ‘y’ value) until the end user navigates to the last page of the section.  
• If an end user navigates to the last page using the “SHIFT+toolbar button” combination, all pages are generated and the page # toolbar icon tool tip shows the correct page position and total page value. Subsequent navigation continue to show correctly the current and total page numbers. These values persist for the entire user session.  
• If the Report sections has fields in the section which show ‘Page x of y’, the fields match the page # toolbar icon tool tip.  
All pages are generated if the Report section is exported to XLS or PDF formats.  
• If the Interactive Reporting document is saved without Results which the Report section references, processing the Interactive Reporting document causes a generation of the first page only.  |
| Last Saved    | Inserts date on which the report was last saved in MM/DD/YYYY format.       |
| Last Printed  | Inserts date on which the report was last printed in MM/DD/YYYY format.     |
| Date          | Inserts and stamps the current date in MM/DD/YYYY format.                  |
| Time          | Inserts and stamps current time in HH:MM:SS format.                        |
| Date & Time   | Inserts and stamps date and time in MM/DD/YYYY and HH:MM:SS format.        |
| Date Now      | Inserts the current date in MM/DD/YYYY format.                            |
| Time Now      | Inserts the current time in HH:MM:SS format.                              |
| Date & Time Now | Inserts the current date and time in MM/DD/YYYY and HH:MM:SS format.    |
| File Name     | Inserts an Interactive Reporting document (BQY) name.                     |
| Path Name     | Inserts the full path name of the document.                               |
| Report Name   | Inserts the report name.                                                  |

**Working with Computed Fields**

A computed field contain a single value based on a calculation. You can create customized JavaScript expressions or modify JavaScript syntax for an existing report element. A computed field is repeated based on the report component in which it is inserted.
To create a computed field:

1. Select **Report**, then **Insert Field**.

   The cursor changes to a crosshair.

2. Left-click and hold your left mouse button to position your cursor in the report component where you want to insert the field, and then drag to create the table.

   A field labeled Empty Field is inserted.

3. Click the empty field to select it, and then enter JavaScript syntax in the Expression line.

   You can copy and paste JavaScript syntax from existing report items into the Expression line to build the expression. To do this, select an existing item and copy the syntax in the Expression line. Then select the new field and paste the syntax into the Expression line.

4. When you have defined the equation, click the check icon.

   The field displays the computed value.

A computed field repeats itself based on the band, header, or footer in which you insert it as shown below:

<table>
<thead>
<tr>
<th>Report Element</th>
<th>Repeats</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report Group Header</td>
<td>Every Header band</td>
</tr>
<tr>
<td>Report Body</td>
<td>Every Body band</td>
</tr>
<tr>
<td>Report Header</td>
<td>Very first page</td>
</tr>
<tr>
<td>Report Footer</td>
<td>Very last page</td>
</tr>
<tr>
<td>Page Header</td>
<td>Every page</td>
</tr>
<tr>
<td>Page Footer</td>
<td>Every page</td>
</tr>
</tbody>
</table>

**Inserting Filter Values**

You can select and show filters applied in the Query and Results sections to aid in the visual construction and production of the report.

To use a filter value in a report:

1. Select **Report**, then **Insert Predefined Field**, then **Query Filter** or optionally select **Result Filter**.

   The cursor changes to a crosshair.

2. Left-click and hold your left mouse button to position your cursor in the report component where you want to insert the filter field, and then drag to create the field.

   The Insert Filter Values dialog box is displayed.

3. In the Query or Result Sections list, select the section that contains the filters that you want to show.

   A list of applied filters populates the Filter Names list.
4 **Select the Filter Name that you want to use from the Filter Names list.**

To show the operator used in the expression, select the Include Filter Operator check box.

5 **Click OK.**

### Modifying Filter Fields

- To modify the Filter field name or the values being shown, double-click the **Filter** field and modify the statement on the Expression line.

The JavaScript expression for the Filter Value field is displayed in the Expression line.

### Working with a Report Page

The Report section gives you complete control of your report page setup and provides a number of features that assist you in designing effective, high-quality reports. Review these topics for information:

- “Displaying Rulers” on page 292
- “Using Grids” on page 292
- “Using Design Guides” on page 293

### Displaying Rulers

Horizontal and vertical rulers help you line up items based on precise units of measure. Available units of measurement include inches, centimeters, and pixels, which you select by clicking the Measurement icon at the intersection of the top and left rulers.

- To toggle the display of a ruler, select **Report**, then **Ruler**.

A check mark is displayed next to Ruler to indicate the ruler is visible. Select this option again to clear the check mark and remove the ruler from view.

### Using Grids

Use the layout grid to snap all objects to the closest grid spot automatically.

- To toggle the display of the grid, select **Report**, then **Grid**.

A check mark is displayed next to Grid to indicate the grid is visible. Select this option again to clear the check mark and remove the grid from view.
Using Design Guides

Design guides are horizontal and vertical lines that you place in your report to help you line up objects. Design guides are similar to grids in that objects automatically snap to align to the design guides.

If rulers are visible, click the ruler and drag one or more design guides from both the horizontal and vertical rulers.

To toggle the display of design guides, select Report, then Design Guides.

A check mark is displayed next to Design Guides to indicate they are visible. Select the option again to clear the check mark and remove the design guides.

Setting Up a Report

Use the Report Page Setup command to specify report page parameters including, page size, margins, and columns. Review these topics for information:

- “Specifying Page Size” on page 293
- “Specifying Page Margins” on page 294
- “Setting Up Page Columns” on page 294

Specifying Page Size

When deciding on page size for your report, consider the type of output you want. Do you want to print the report or view it on screen? You can select printer or custom dimensions for your report pages.

To specify page size:

1. Select Report, then Report Setup.
   The Report Setup dialog box is displayed.
2. Click the Page Size tab to view the Page Size page.
3. Select the page dimension option you want to use and click OK.
   - Use Printer Dimensions—Determines the printable area based on your printer. If you want to only print the report, select this option.
   - Use Custom Dimensions—Enables you to specify the width and height of your report page.
     If you export reports to HTML pages, you can use this setting to control the number of “pages” that are included in each HTML file. Each page is exported to a single HTML file by default. To ensure that all pages in your report are contained in a single HTML file, set the page height to a large number.
Specifying Page Margins

Page margins are the blank space that borders the report area on your page. Since the report area can hold a variety of objects, such as page numbers, text, and lines, change the margin area to accommodate the size of your report area. Margins are set for an entire report.

To specify page margins:

1. Select Report, then Report Setup.
   The Report Page Setup dialog box is displayed.
2. Click the Margins tab to view the Margin page.
3. Enter the Top, Bottom, Left and Right margin sizes and click OK.
   Margins are measured in units of inches, pixels or centimeters. To change the margin units, click the Measurement icon located above the Content pane until you find the measurement unit that you want to use.

Margin Measurements

Margins are set either in inches, pixels, or centimeters. Default margins for all four sides are:

<table>
<thead>
<tr>
<th>Type of Measurement</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inches</td>
<td>.5</td>
</tr>
<tr>
<td>Pixel</td>
<td>.48</td>
</tr>
<tr>
<td>Centimeters</td>
<td>1.28</td>
</tr>
</tbody>
</table>

Note: The default number of pixels depends on your screen resolution.

To change how margins are measured:

Click the Measurement icon above the Contents pane until you find the desired measurement standard.

Measurement standard code definitions are: in= inches, px=pixels, and cm=centimeters.

Setting Up Page Columns

You can set up a multicolumned report and have data flow from the bottom of one column to the top of the next column. You can specify as many columns as you want.

Page columns can be displayed in a number of layouts so that they are both visually attractive and easy to follow.

For example, you can set up report columns to look like newspaper-style columns (snaking columns) where data wraps seamlessly from the bottom of one column to the top of the next.
To set up page columns:

1. **Select Report, then Report Setup.**
   - The Report Setup dialog box is displayed.

2. **Click the Column tab to view the Column page.**

3. **Select the number of columns that you want to include on the page.**

   - If you want to include more than four columns on the page, enter the number of columns in the Other field.

   - The column width and spacing are determined automatically based on the number of columns specified, the page size, and the page margins. Column width and spacing measurements are set in either inches, pixels, or centimeters.

**Column Measurements**

Column width and spacing measurements are set in either inches, pixels or centimeters. The default width and spacing dimensions for columns are:

<table>
<thead>
<tr>
<th>Type of Column Measurement</th>
<th>Column Number(s)</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inches</td>
<td>1</td>
<td>8 &quot;</td>
</tr>
<tr>
<td>Inches</td>
<td>2</td>
<td>3.75&quot;</td>
</tr>
<tr>
<td>Inches</td>
<td>3</td>
<td>2.33&quot;</td>
</tr>
<tr>
<td>Inches</td>
<td>4</td>
<td>1.63&quot;</td>
</tr>
<tr>
<td>Spacing</td>
<td></td>
<td>0.5&quot;</td>
</tr>
<tr>
<td>Pixels</td>
<td>1</td>
<td>768 px</td>
</tr>
<tr>
<td>Pixels</td>
<td>2</td>
<td>360 px</td>
</tr>
<tr>
<td>Pixels</td>
<td>3</td>
<td>224 px</td>
</tr>
<tr>
<td>Pixels</td>
<td>4</td>
<td>156 px</td>
</tr>
<tr>
<td>Spacing</td>
<td></td>
<td>48 px</td>
</tr>
<tr>
<td>Centimeters</td>
<td>1</td>
<td>20.33 cm</td>
</tr>
<tr>
<td>Centimeters</td>
<td>2</td>
<td>9.53 cm</td>
</tr>
<tr>
<td>Centimeters</td>
<td>3</td>
<td>4.13 cm</td>
</tr>
<tr>
<td>Centimeters</td>
<td>4</td>
<td>1.27 cm</td>
</tr>
<tr>
<td>Spacing</td>
<td></td>
<td>1.27 cm</td>
</tr>
</tbody>
</table>
Enhancing Report Data

You can apply filters, sorts, computations, and break totals to refine the data in your reports. Review the following:

- Sorting Report Items
- Adding Computed Items
- Creating Computed Columns
- Applying Data Functions
- Adding Totals
- Hiding and Focusing on Reported Data

Sorting Report Items

Use the Sort icons to quickly sort a report group header or table column.

➤ To sort items quickly:

1
2. Drag the table column that you want to sort.
3. Click the Ascending or Descending icons on the standard toolbar.

➤ To apply sort conditions using the Sort line for report group labels:

1. If the Sort line is not already displayed, click **Sort** on the Section title bar.
   The Sort line is displayed below the Section title bar.
2. Click a report group label in the Content pane.
   The Sort line should read Report Groupx, where x = 1, 2, 3, and so on.
3. Drag a report group label from the content area to the Sort line.
4. Double-click the name of the report group label on the Sort line to toggle between ascending and descending sort order.

➤ To sort table columns:

1. If the Sort line is not displayed, click **Sort** on the Section title bar.
   The Sort line is displayed below the Section title bar.
2. Click anywhere within the table column (dimension or fact) in the Content pane.
3. Drag table dimensions or table facts from the Content pane to the Sort line.
4. Reorder the Sort items to determine the nested sort order.
5. Double-click the item in the Sort line to toggle between ascending and descending sort orders.
Adding Computed Items

Use the Add Computed Items command to build equations to compute totals, or to apply functions to existing values. Since a report may derive its values from a wide range of data sources (relational database queries, OLAP queries, imported data sets, and local-join queries), you select the Request item that you want to use from the applicable Results section.

To add a computed item:

1. Select a table column and select Report, then Add Computed Item
   The Computed Item dialog box is displayed.
2. In the Column Name field, type a name that describes the computation.
   The default name is Computed.
3. In the Tables list, select the results set on which to apply the computed item.
4. In the Columns list, select the column from which to compute the new data item.
   Note that the columns available for reference are only the columns that exist in the results set that is the source for the rest of the table.
5. Modify the existing syntax of the table column you selected or define a new computed item by building an expression in the Formula pane.
   Interactive Reporting uses JavaScript to compute data items. The JavaScript engine supported in Interactive Reporting does not accept European or non-US number formats for computing data items. More information about JavaScript can be found in Oracle Hyperion Interactive Reporting Object Model and Dashboard Development Services Developer’s Guide, Volume I: Dashboard Design Guide.
6. Click OK to apply the definition to the computed item column.
   A new computed column is added to the Facts pane in the Table data layout for the selected table.

To add a free form computed item:

1. Select a table column and select Report, then Add Computed Item or click Add Computed Item on the shortcut menu.
   The Computed Item dialog box is displayed.
2. In the Column Name field, type a name that describe the free form computation.
3. Click the Formula field.
   The Formula field is activated.
4. Define a new computed item column by building an expression in the Formula field and click OK.

Creating Computed Columns

When looking for answers to basic business questions, raw data cannot always provide the information. For example, while a database might track sales figures, sometimes this information
is much more valuable with reference to more complicated calculations such as cost of sales or profit margins.

You can anticipate these requirements, designing the internal database structure so that it makes such calculations as the raw data is collected. But at times you might not find the exact information you need because no one has needed it before. That is where Computed data items are most useful. You use these items to generate new information, usually from data that is already stored in the database. If the database does not offer this information, you can use Interactive Reporting to compute it.

**Note:** Equations in the Report section are written in JavaScript syntax. When creating new data item columns, the Computed Item dialog box is not used. Instead, equations are defined in the Expression Line.

To create a computed column:

1. **Build a Tabular report.**

2. **Drag an item from Elements to the Table Facts pane of the data layout.**

   You cannot modify items added to the Table Dimensions pane of the data layout. Either add an item to the Table Facts pane or modify the item in the Results section.

   When selecting an item from the Catalog, select an item that is part of the equation for the computed item. For example, if Unit Sales is part of the equation, then drag Unit Sales to the Expression bar.

3. **Define the expression in the Expression Line.**

   Undock the Expression Line and resize the window if the equation extends beyond the viewing portion of the window.

   Change the data function applied to the item by selecting another data function from the pull-down list on the Expression Line.

   To apply an arithmetic equation to the expression, enter the applicable operators.

   Drag additional items from Elements to the Expression Line as needed. The JavaScript syntax for items are generated automatically.

4. **When the equation is complete, click the green check mark on the Expression Line.**

   The equation is calculated and the values are displayed in a column in the table.

In the following example, the report determines which Store Managers receive $2,000 and $3,000 bonuses. A Level 1 Bonus of $2,000 is granted if the store’s revenue exceeds $5,000.00. A Level 2 Bonus is granted if the store’s revenue exceeds $10,000.00.
Applying Data Functions

A Dashboard enables you to determine the nature of the values represented in a Table Fact column in the Report section. Data functions retrieve underlying values and recalculate the value according to the type of data function that you specify. For example, the totals for a column are sums (the default data function) of the values in a column. You can use other data functions to change the totals to averages instead of sum.

Interactive Reporting provides a variety of prebuilt data functions that can be accessed from the Expression Line. You can also define your own data functions using JavaScript.

To apply a prebuilt data function in the Report section:

1. Select a table fact.
2. Select Report, then Data Function, and then Function.

The totals in the column are recomputed to reflect the selected data function.

The following table lists the prebuilt data functions available in the Report section.

Table 83  Prebuilt Data Functions in Report section

<table>
<thead>
<tr>
<th>Data Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Column Name</td>
<td>Returns the table column name.</td>
</tr>
<tr>
<td>Sum</td>
<td>Returns sum of underlying values. This is the default data function.</td>
</tr>
<tr>
<td>Average</td>
<td>Returns average of underlying values.</td>
</tr>
<tr>
<td>Count</td>
<td>Returns number of underlying values.</td>
</tr>
<tr>
<td>Maximum</td>
<td>Returns highest of underlying values.</td>
</tr>
<tr>
<td>Data Function</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Minimum</td>
<td>Returns lowest of underlying values.</td>
</tr>
<tr>
<td>% of Category</td>
<td>Returns group total percentage of the value.</td>
</tr>
<tr>
<td>Null Count</td>
<td>Returns number of nulls among underlying values.</td>
</tr>
<tr>
<td>Non-Null Count</td>
<td>Returns number of underlying values; null values are excluded.</td>
</tr>
<tr>
<td>Non-Null Average</td>
<td>Returns average of underlying values; null values are excluded.</td>
</tr>
</tbody>
</table>

To set a data function to the default sum total:

1. Select a column that has been computed with a data function.
   - A pull-down list of valid data functions displayed.
2. Click  on the Standard toolbar.

**Adding Totals**

By default tables show the total of any column placed in the Table Facts area of the data layout. However, sometimes you want the total to be in different places in the report: in a group header, group footer, or somewhere else in the body. You can do this by dragging a column from the section catalog and dropping it directly into the desired report section. Any column that you drag directly into the report is aggregated using the sum function based on the current break level. Note that you do not need to have the column exist in a table; you can drag any column in your results directly into the report.

Frequently, you may want to display different types of the same underlying values displayed side by side such as the average, maximum, or count values from the original data as shown below. In this case, data functions are particularly useful. By dragging the same column from the section catalog next to the original column and applying a data function, different aspects of the original data are revealed.

**Applying Break Totals**

To apply a break total to a column in a report table:

1. Select a column and select Report, then Break Total.
   - The Insert Break Total dialog box is displayed.
2. Select a break column from the At Every Break in drop-down list box.
3. Select the data function you want to apply from the Break Total Function drop-down list box.
   - See “Applying Data Functions” on page 5-35 for a list of data functions that you can use with break totals.
4. Select one or more columns on which to display the break total and click OK.
Using True Totals for Computed Items

Both report totals and break totals for tables can be calculated based on the column’s data function applied to surface values, or the underlying formula of the column. This option is controlled by the “True Total Computation” feature.

Underlying values refer to values from the original Results section. Surface values refer to values in the actual report section. The two approaches produce values that may be displayed incongruous with the values in the report.

To understand this difference between underlying and surface values, consider a simple table with two values of 20 and 30. Each of these is already a total of underlying values (20 = 8 + 12 and 30 = 10 + 20). An average of the underlying value yields the result of 12.5 = (8 + 12 + 10 + 20) / 4). An average of the surface value yields the result of 25 = (20 + 30) / 2).

The True Total Computation enables you to standardize how total values in a computed column are calculated. When this feature is enabled, all total values are calculated based on the column’s data function applied to the surface values. When this feature is disabled, all totals are calculated based on the underlying formula of the column.

Pre Release 8.3.2 tables (legacy) tables use the calculation method where one total is calculated by the underlying computed item formula for its computation, and the other total applies the column’s data function to the surface values. You can change this behavior by also using the True Total Computation feature.

For legacy computed item columns, the “True Total Computation” feature shows the text “Legacy Total Computation” on the shortcut menu. This indicates that the column is using the inconsistent legacy total/break total computations.

To use true totals for the table total or break total in a computed column:

1. Select the computed column to which you want to apply the true total.

2. On the shortcut menu, select True Total Computation.

When the True Total Computation is deselected, totals and break totals are computed using the underlying formula of the column.

To use true totals for the legacy table total or break total in a computed column:

1. Select the legacy computed column to which you want to apply the true total.

2. On the shortcut menu, select Legacy Total Computation.

When you select a column, “Legacy Total Computation” is displayed on the shortcut menu. This means that the column is using an inconsistent total/break total computation.

Selecting the Legacy Total Computation from the shortcut menu disables the feature.

An Alert dialog box indicates that the change cannot be undone: “This action cannot be undone. Are you sure you want to use True Total Computations for Total and Break Totals?”

3. Click Continue.

True Totals are enabled for both Totals and Break Totals of this column.
The next time that you display the shortcut menu, “True Total Computation” is displayed as an option in place of “Legacy Total Computation”. At this point, you can no longer apply the legacy behavior to this column.

**Hiding and Focusing on Reported Data**

A straightforward way to refresh your view of a report is to single out items for closer focus or remove some of the reported elements. This feature enables you to concentrate on particular items of interest. Hiding a group item removes the item from the report, but not from the data layout. Focusing a group item removes all items except the focused item from the report, but not the data layout.

**Focusing on Items**

To focus on a report item:

1. Select a group header and select **Report**, then **Focus On Item**.

   Interactive Reporting updates the report to focus on the data. A drill icon is displayed in the data layout next to the column you selected.

2. Select **Chart**, then **Focus On Items**.

   The chart is redrawn to display only the chart object(s) selected. A drill icon is displayed in the data layout next to the item(s) on which you have focused.

**Show All Items**

Use this feature when you need to return the display to the original view.

To return to the original chart display, select **Chart**, then **Show All Items**.

**Hiding Items**

Use this feature when you need to hide a selected item.

To hide a group item, select a group header and select **Report**, then **Hide Item**.

You can only select one group item to hide at a time.

Interactive Reporting updates the report to hide the data. A drill icon is displayed in the data layout next to the column that you selected to hide.

To restore all hidden groups, select **Report**, then **Show Hidden Items**.

All items listed in the data layout are fully restored to view. The drill icon is displayed.
Using Multiple Data Sources in a Report

You can create reports that contain data from multiple queries that use a wide range of data sources (relational database queries, OLAP server queries, imported data sets, and local join queries).

In a sense, the data in a single table correlates data across queries in a report, retrieving all rows from the table on the “left” and any rows from the table on the “right” that have matching values. Unlike actual joined topics in the Query section, the tables are not linked in the same sense. In the Report section, data is only in the report.

The rules for using multiple data sources in a single report are:

- Report group headers require a common value that belongs to all results or table sets be included in the report. If you introduce a unique value belonging to one results or table set, nothing is returned for any band below that group header.
- Table Dimensions items can originate from only one data source.
- Values from multiple data sources can be included in Table Facts.
- The Results column that you place as a table determines the Facts used. For example, you have two Results sets:

<table>
<thead>
<tr>
<th>Results 1</th>
<th>Region</th>
<th>Sales</th>
<th>Results 2</th>
<th>Region</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>East</td>
<td>10</td>
<td></td>
<td>East</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>West</td>
<td>20</td>
<td></td>
<td>West</td>
<td>200</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>South</td>
<td>400</td>
</tr>
</tbody>
</table>

If you place “Region” from Results 1 into the Table Dimension and then drop Results 1: Sales and Results 2: Units into the Table Facts, you get the following table:

<table>
<thead>
<tr>
<th>Region</th>
<th>Sales</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>East</td>
<td>10</td>
<td>100</td>
</tr>
<tr>
<td>West</td>
<td>20</td>
<td>200</td>
</tr>
</tbody>
</table>

- Note that the data from Results 2: South is not included.
- Smart charts and pivot tables added to a multiple data source report are subject to the same restrictions as described above.

To build a multiple data source report:

1. **Build the queries that you want to include in the report:**
   - Verify item data types and associated data values in source documents so that you know how to correlate them when creating the reports.
   - Build the Request line, and add server and local filters, data functions, and computations to the query as needed.
2 Click **Process**.

3 Select **Insert**, then **New Report**.
   A new report section is displayed.

4 From **Elements**, expand the file button of the Results or Table set.

5 If necessary, click **Groups** on the Section title bar to open the Group data layout.
   A group header categorizes data into repeating collections of records in a header band.

6 If necessary, click **Table** on the Section title bar to open the Table data layout.
   The Table data layout shows two panes: Table Dimensions and Table Facts. Use the Table Dimensions pane to build the column dimensions (labels) in the report. Use the Table Facts pane to build the numeric values in the report.

7 Build the group header band by dragging the item(s) from the Results sections from **Elements** to the Groups data layout.

8 Build the column dimensions of the report, drag the item(s) from the Query sections from **Elements** to the Table Dimensions in data layout pane.

9 Build the column numeric values of the report by dragging the item(s) from the Query sections from **Elements** to the Table Facts in data layout pane.

### Creating Smart Reports

Smart reports enable you to embed charts and pivot tables into a report body. These reports show only the data that is relevant to the report section in which they are placed. For example, if the report is grouped by year and you insert a chart in the report body, the chart replicates automatically so that there is a chart for each year of data in the report. Each chart contains data specific to that year.

**Tip:** Smart reports are refreshed whenever you reprocess the query. Also, if you change a chart or pivot table in its home section, the Smart report is updated and the changes are reflected in the Report section.

➢ To embed a chart or pivot in a report:

1 From **Elements**, drag a pivot table or chart to either the report group header, page header, footer, or report body.

2 Size the pivot table or chart by clicking the embedded report until handles are displayed.
   The pointer changes to a two-sided arrow.

### Formatting Report Items

The following table lists formatting techniques you can use in the Report section.
Table 84  Report Formatting Options

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>To modify the way numbers, dates and currency are displayed, select the item and select Format, then Number. When the Number tab are displayed, apply any desired changes.</td>
</tr>
<tr>
<td>Style</td>
<td>To modify the appearance of a report item, select the item and select Format, then Style, and then Option. Options include bold, underline, italic, and so on</td>
</tr>
<tr>
<td>Alignment / Justify</td>
<td>To modify the way an item is justified within a row or column, select an item, select Format, then Justify, and then Option. Options include left, center, right, top, middle, and bottom.</td>
</tr>
<tr>
<td>Font</td>
<td>To modify the font of a report item, select the item and select Format, then Font. When the Font tab of the is displayed, apply any desired changes.</td>
</tr>
<tr>
<td>Keep Together</td>
<td>Instructs Interactive Reporting not to split a band when a break is encountered. When a break is encountered, the entire band is moved to the next page.</td>
</tr>
<tr>
<td>Keep With Next</td>
<td>Instructs Interactive Reporting to keep bands within a group together when paginating a report. If the lower band cannot fit on the page when the report is paginated, both bands are moved to the following page.</td>
</tr>
</tbody>
</table>

Converting Detail Reports from Versions Earlier than 6.0

You can use the Report Designer to modify Detail reports created in documents prior to version 6.0.

Version 6.0 and later opens Detail reports in read-only mode. To work with the contents of a Detail report, first convert it to the new Report Designer format.

**Note:** You can convert Detail reports to Report Designer format only if the results set is saved with the document. If your results set is not saved with your document, reprocess the query and then select File, then Save Results With document.

To convert a Detail report to Report Designer format:

1. **In the Section catalog, select the Detail report.**
   The detail report is displayed in the Content pane and a Detail menu is displayed on the menu bar.

2. **Select Detail, then Convert To Report.**
   The Detail report is converted to the new Report Designer format and inserts a new Report section into the document. The original Detail report is left unchanged.

While every effort has been made to make the converted report as close as possible to the original Detail report design, you may need to clean up some reports, especially those that use computed items, multiple page headers/footers, or complex formatting.
Detail reports can be exported from the current version of the product to all of the previously supported formats, including Excel (.xls), Lotus 1-2-3 (.wks), tab-delimited text (.txt), comma-separated text (.csv), and HTML (.htm).

**Display Differences**

The page margins of a report created by the conversion process may be displayed smaller than in the original Detail report. This is because the new Report section is a WYSIWYG display that includes the unprintable area that exists outside of report margins. In version 5.x, Detail reports did not display or store information about the unprintable area. You can manually adjust the margin sizes in the converted report if needed.

If a Detail report contains multiple page header and footer sections, the current version of the product resolves these to a single page header and footer section during the conversion process. The height of the headers and footers in the converted report is equal to the combined heights of all headers and footers in the original Detail report. This may cause graphic and text objects in the headers and footers to overlap. You should manually adjust the properties, size, and/or position of these objects as necessary.

**Conversion of Detail Report Categories**

Detail Report categories from Detail reports are converted into report groups in the new Report section. The converted report displays Report Group Header and Footer sections if the corresponding category header and footer bands are visible in the original Detail report. The height of these header and footer sections remains the same. Detail report category labels are converted into JavaScript-based computed fields. Display properties remain unchanged.

**Conversion of Data Area**

To preserve as much of the original data and layout as possible, the conversion process translates the Detail report body information to a single table object in the body section of the converted report. This may lead to a loss of fidelity when converting complex or non-tabular style Detail reports.

**Conversion of Facts**

During report conversion, Detail report facts are added to the right of the dimension columns in the body section table. Number formats applied to Detail report facts are migrated to the converted report. If no number format exists, then the default number formats are applied to date, time, and number fields.

Because version 6.x uses JavaScript as the scripting, the conversion process does not automatically convert any computed facts contained in the Detail report. For computed items located in the Detail report body, the expression is omitted from the version 6.x table object. The conversion process may attempt to replace the computed column with another fact. If that occurs, simply delete the extraneous column.
For computed items located in a group header or footer, the code of the old scripting language expression is placed in a text field in the associated header/footer area to aid in manual conversion. Before you correct these placeholder items, Hyperion displays an error message in the field which reads `<string>(1):unterminated string literal`. The Detail report’s Surface Math option has no equivalent in the converted report and is ignored.

**Conversion of Smart Reports**

Version 6.x is able to convert Detail reports containing multiple *smart* charts and pivot tables. Some converted smart charts may look different or even invisible once converted and displayed in the 6.x Report section. This is due to some changes in the chart section that affect the way that charts are plotted within a given sized area. As a result, embedded charts may need to be resized manually by the user.

**Conversion of Graphic Objects**

Certain graphic object properties are no longer supported and are ignored during the conversion process. These properties include the shadow, sunken button, and raised button fill effects.

The width of line objects in Detail reports is expressed in whole pixels. When a report is converted, the pixel line widths are converted to the nearest point equivalent.

**Report Menu Command Reference**

The following provides a quick reference to the commands available on the Report menu and lists any related shortcuts.

<table>
<thead>
<tr>
<th>Table 85</th>
<th>Report Menu Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Command</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>Design Guides</td>
<td>Toggles the display of design guides.</td>
</tr>
<tr>
<td>Grid</td>
<td>Toggles the display of grid lines.</td>
</tr>
<tr>
<td>Rulers</td>
<td>Toggles the display of rulers.</td>
</tr>
<tr>
<td>Section Boundaries</td>
<td>Toggles the display of section boundaries (bands).</td>
</tr>
<tr>
<td>Page Margins</td>
<td>Toggles the display of page margins</td>
</tr>
<tr>
<td>Sort</td>
<td>Sorts the selected column values in ascending order (alphabetical or numeric).</td>
</tr>
<tr>
<td>Data Function</td>
<td>Applies a prebuilt data function to the selected item.</td>
</tr>
<tr>
<td>Add Computed Item</td>
<td>Opens the Insert Computed Item dialog box.</td>
</tr>
<tr>
<td>Break Total</td>
<td>Opens the Insert Break Total dialog box.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Focus On Items</td>
<td>Updates the report to include only the selected data.</td>
</tr>
<tr>
<td>Hide Item</td>
<td>Hides the selected item from view.</td>
</tr>
<tr>
<td>Show Hidden Items</td>
<td>Restores the selected hidden item.</td>
</tr>
<tr>
<td>Show All Items</td>
<td>Updates the report to include all items removed by focusing.</td>
</tr>
<tr>
<td>Insert Table</td>
<td>Inserts a blank table in the report.</td>
</tr>
<tr>
<td>Insert Field</td>
<td>Inserts a blank field in the report.</td>
</tr>
<tr>
<td>Insert Predefined Field</td>
<td>Inserts a predefined field in the report.</td>
</tr>
<tr>
<td>Insert Graphic</td>
<td>Inserts a vector graphic in the report.</td>
</tr>
<tr>
<td>Picture</td>
<td>Inserts a picture in the report.</td>
</tr>
<tr>
<td>Remove Selected Items</td>
<td>Removes the selected item from the report.</td>
</tr>
<tr>
<td>Headers and Footer</td>
<td>Toggles the display of report and page headers and footers.</td>
</tr>
</tbody>
</table>
About Filters

When building a query, you usually do not want to see information on every product you sell. Instead, you want to see information that relates only to a specific product or product line. Similarly, you probably do not want to see this information for every year the product has been available, but only for recent periods.

When you set a filter in the Query section, data is returned from the database only if it meets the specified conditions. For instance, suppose you only want to see customers who spend more than $400,000 per year, or who buy gardening products in the Midwest. By applying a filter you are instructing the database to “give me only the data which satisfies the following conditions” (sales > $400,000; or, state is in Midwest Region and Product Line = Garden).

For example, a filter placed on Item Type (which includes an “=” (equal) operator and value “Keyboard”) returns only records associated with keyboard sales. Records associated with all other products are excluded from the Results set. The data set could be expanded to include modem sales records by adding the value “Modem” to the filter expression.

Similarly, the filter “> 5000” applied to the Amount Sold item filters out all sales transactions less than or equal to $5,000. Alternately, the expression “between 5000, 10000” would exclude transactions above $10,000 and eliminate any below or equal to 5,000.
Server versus Local Filter Processing

You apply filters in the Query or Results sections in basically the same way. The difference is in whether the filter is applied at the server or on your desktop.

- Server Filters in the Query Section
- Local Filters in the Results Section

Server Filters in the Query Section

A filter applied in the Query section instructs the database server to filter unwanted information from your request. Only the filtered data is returned to your desktop.

If your query is potentially large, and you are more or less certain of both the information you need and how to define the correct filters, it is best to apply filters in the Query section. By using the database server to filter the data, you return only the data you need across the network and onto your computer.

Another advantage of applying filters in the Query section is that you can apply a filter to any topic item in the data model, even if the item is not on the Request line. For example, if you request State, Year and Units Sold, you can filter any of these items. But you can also place a filter on Operating System (OS) if it is displayed in one of the topics in the work area. If you filter OS to the UNIX only, the server retrieves only sales information related to the UNIX operating system. You need not place OS on the Request line.

You can also filter computed items.

Local Filters in the Results Section

You can apply filters to columns to locally filter the data set in the Results section. Since the other reporting sections reference the results set, these local filters are disseminated to these sections.

Local filters are useful for managing your data set. If you do not need all information retrieved by your query, you can use a local filter to exclude data from view. Local filters are useful for temporary or hypothetical situations. You can always suspend or delete the filter to view the data and make it available for reporting.

Note: You can only apply local filters to items on the Request line.

Filter Line

Using filters involves dragging an item from the Content pane to the filter line and then setting filters in Filter.

The Filter line is a drag-and-drop command line similar to the Request and Sort lines. Filtered items are displayed on the Filter line. You can move, size, dock, and hide the Filter line.

Filter indicators on the Filter line and next to the item in the topic. Filter line expressions available in the Query section Filter line:
Filter Line Functionality

In the Query section, the Filter line includes special functionality:

- The Filter line is an interface for building compound filter, which are multiple filters linked together to form complex filter equations. See Setting Compound Filters.
- The Filter line enables you to convert a filter to a variable filter, which prompts the user of the Interactive Reporting document file to select filters as the query is processed. See Setting Variable Filters.
- When placing a filter on a request item computed with a data function, a divider is displayed on the Filter line, and the Filter icon is placed to the right of the divider. The divider indicates that the filter is applied in the SQL Having clause.
- Subqueries and correlated subqueries which use the result from an inner query as the value of a filter in an outer query.

Filter Line Syntax

Syntax rules that apply to all Filter line expressions:

- By default, all equations are solved from left to right, with enclosed sub-operations evaluated first.
- AND is evaluated before OR.
- The AND operator retrieves data that meets both conditions. For example, if you query customers, and filter State to “Florida” AND Item Type to “Modem,” the data retrieved would apply only to customers buying modems in Florida, not to modems bought in Minnesota or keyboards bought in Florida.
- The OR operator retrieves data that satisfies either of two conditions. For example, if you filter State to “Florida” OR Item Type to “Modem,” the data retrieved would include Florida customers and any customers purchasing modems. It would not include customers purchasing keyboards (unless they lived in Florida), or customers in Minnesota (unless they bought modems).
- Suboperations allow you to override the default evaluation order, and may be required for certain operations involving both AND and OR operators.

Filter Controls

When applying a filter, you supply (or select) data values associated with a data item and use mathematical logic to apply the values as constraints.

- ()—Encloses suboperations.
- Var—Indicates a variable.
- AND—Retrieves data that meets both condition.
- OR—Retrieves data that satisfies either of two conditions.
Name—Descriptive name for the filter.

Include Nulls—Toggles the inclusion/exclusion of null values.

Not—Reverses the effect of an operator (for example, ‘Not >=’ is equivalent to <).

Operators—Comparison operators for the filter expression. Values that pass the comparison test will be included.

Edit field—Enter a value (or multiple values separated by commas), and click the check mark to add them to the custom values list to complete the filter definition. Click the “x” to erase the contents of the Edit field.

Note: The Edit field is displayed only if you are entering a custom value.

Show Values—Shows all potential values from the database that are associated with the item.

Note: Show values cannot be used in the Query section for filters on computed or aggregate values.

For more information, see:

- Defining Custom Values
- Multiple Filters and the Meta Topic

Custom Values—Lists potential values saved with the filter or read from a file. This feature enables you to select values from a predefined pool. You can create and save a custom list with each filter.

Custom SQL—Displays custom SQL for coding Filters directly in SQL.

Note: The Custom SQL button is displayed only if you apply a filter in the Query section.

Select All—Selects all values displayed in the list of values.

Transfer—Adds selected values to the custom list.

Ignore—Temporarily suspends a filter without deleting it.

Fractional Digits—Specify the number of digits that you want to place to the right of the decimal point in an SQL numeric literal statement. The SQL statement uses this setting to determine which values to recognize. For example, if you type the filter value, 0.12345678, the SQL statement writes “where column = 0.12346” (the rounded value), but if you change the spinner to 8, then the SQL writes “where column = 0.12345678”.

Advanced—Displays Loaded Values settings and subquery options.

Loaded Values Settings—Toggles a custom values list to be read from a file or from the database. Change File allows you to specify the file name. If you read values from a text file, vertical tabs or paragraph markers must defilter each value. Use Show Values to display the file contents.
Create Subquery—Creates a subquery.

Note: If you access Filter from the Results section, the Advanced button is displayed as Options and the Create Subquery option is not displayed.

Showing Values

The Show Values command is a powerful option for selecting values to define filters. Show Values provides a list of values actually in the database (or in the data set in Results) and allows you to choose a value based on the data available.

This feature makes it possible to set filters accurately without being familiar with the contents of the database. Show Values is also advantageous when values change frequently and custom lists quickly become outdated.

To show values and define a filter:

1. Select Show Values on the Filter dialog box to retrieve the item’s values from the database.
2. Select values in the Values pane and click OK.
3. Click Select All or select values individually.

To create a snapshot of the values, click Select All and then click Transfer to move the variables to Custom Values.

Tip: Because Show Values retrieves every value available, it is best not to use it in the Query section when the data item is large, consists mostly of values, or does not change frequently (for example, telephone numbers). In these situations, custom lists are sufficient and help avoid extra calls to the database. You administrator may disable Show Values if database hits are a concern.

Note: Show Values cannot be used for filtering computed or aggregate items.

Defining Custom Values

Custom value lists are created by or supplied to Interactive Reporting. When users open the filter to choose new values, they can choose a filter from a custom list that has been supplied.

One reason to use custom lists with distributed Interactive Reporting document files is that many data items change rarely, if at all. For example, a Gender item has three consistent values (male, female, and unknown). A Product line item has many more values, but may only change every year or so. Data that changes rarely is said to have low cardinality. Under such circumstances, it makes sense for users to select from a custom values list, rather than continuously querying to show database values.

Custom lists are also useful when Setting Variable Filters. When an Interactive Reporting document file creator sets a variable filter, a custom list can be prepared and added to it. The subsequent users who sets the filter can use the custom list or read more custom values in from an external file.
To create a custom values list, use any method:

- Click **Show Values** in Filter to retrieve the item’s values from the database, then select the values to be included in the custom list and click **Transfer**.
- Click **Custom Values** and enter values individually in the edit field, then click the check mark to add each to the custom list.
- Click **Advanced** and select **Load From File** to read values from a separate file. Use the **Change File** button to choose a file from the Select Filter File window. Click **Show Values** to load the values from the file to the list of values. Click **Transfer** to add selected values to the custom list.

Values added to the custom list can be selected to determine the filter. If the filter is converted to a variable filter, the custom list is stored in the Interactive Reporting document file and the saved values are available to users who resolve the filter.

**Tip:** Imported values must be vertical-tab or paragraph delimited.

### Using Operators

Logical operators for defining filters support standard SQL wildcards, including “%” and “_”.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equal (=)</td>
<td>Retrieves records where the filtered item equals the specified value(s).</td>
</tr>
<tr>
<td>Not Equal (&lt;&gt; )</td>
<td>Retrieves records where the filtered item does not equal the specified value(s).</td>
</tr>
<tr>
<td>Less Than (&lt;)</td>
<td>Retrieves records where the filtered item is less than the specified value(s).</td>
</tr>
<tr>
<td>Less or Equal (&lt;=)</td>
<td>Retrieves records where the filtered item is no greater than the specified value(s)</td>
</tr>
<tr>
<td>Greater Than (&gt;);</td>
<td>Retrieves records where the filtered item is greater than the specified value(s).</td>
</tr>
<tr>
<td>Greater or Equal (&gt;=)</td>
<td>Retrieves records where the filtered item is no less than the specified value(s)</td>
</tr>
<tr>
<td>Begins With</td>
<td>Retrieves records where the filtered item begins with the specified value(s) up to and including the end value.</td>
</tr>
<tr>
<td>Contains</td>
<td>Retrieves records where the filtered item contains the specified value(s) regardless of location.</td>
</tr>
<tr>
<td>Ends With</td>
<td>Retrieves records where the filtered item ends with the specified value(s).</td>
</tr>
<tr>
<td>Like (with wildcards)</td>
<td>Retrieves records where a text string is displayed and reflects the placement of the specified value(s). For example, Name Likes _ would retrieve records for all employees whose names have the letters Ze followed by one character at the end</td>
</tr>
<tr>
<td>Is Null</td>
<td>Retrieves records where the filtered item has no value; for example a field in which no data has been entered.</td>
</tr>
<tr>
<td>Between</td>
<td>Retrieves records where the value of the filtered item lies between (and does not equal) the specified values.</td>
</tr>
<tr>
<td>Not (with operator)</td>
<td>Negates the operator it precedes, reversing the results of the equation</td>
</tr>
</tbody>
</table>
Note: Text strings are evaluated differently among database systems, and are often case-sensitive (that is, “RED”, “Red” and “red” may not be equivalent in comparison). Text characters are generally valued in ascending order (0 1... 9 A B C... Z a b c... Z.), so that “A” is less than “a” in string comparisons, and strings beginning with numbers are considered “less” than strings beginning with alphabetic characters. See your administrator for specific information.

Setting Simple Filters

Simple filters are applied in basically the same manner in the Query and Results sections. If you are working with a distributed Interactive Reporting document file, some of the buttons in the Filter dialog box may not be available.

- Filtering Queries
- Filtering Results

Filtering Queries

The simplest way to filter a query is to apply filters individually to topic items.

To filter a query:

1. Select an item in the data model and choose Query, then Add Filter(s).
   The Filter dialog box is displayed.
2. Select an arithmetic or logical operator from the list.
3. Define the potential filter values:
   - Show Values—Shows database values associated with the item.
   - Custom Values—Supplies an empty field for inputting custom values. Click the check mark to add a value to the list of values. You can also display values from a previously saved custom list or values loaded from a file.
   - Custom SQL—Supplies an empty box for typing an SQL clause to be included in the query statement. Choose View, then Custom SQL to check the complete statement logic.
4. Select the values to include in the filter definition in the Values list.
   Individually select values or click Select all and deselect the values you do not want to include. Only selected items are applied to the filter definition. To create a snapshot of the values, click Select All and then click Transfer to move the variables to the Custom Values.
5. When the correct values are highlighted in the Values list, click OK.
   The filter is applied to the topic item and an icon is added to the Filter line.
   In the Query section, the filter is applied when you process the Query.
Note: When placing a filter on an aggregate item (computed with a data function) in the Query section, a filter divider is displayed and the filter icon is placed to the right. The divider indicates the filter is in the SQL Having clause. You must drag the aggregate item from the Request line to the Filter line to create a filter on an aggregate item.

Removing a Query Filter

➢ To remove a filter on a query, select an item on the Filter line and select **Remove** on the shortcut menu.

Filtering Results

➢ To filter the display of data in the Results section:

1. **Select a column** (click the column heading) and choose **Results**, then **Filter**.
   The Filter dialog is displayed.

2. **Select an arithmetic or logical operator from the list**.

3. **Define the potential filter values by clicking one of the following options**:
   - **Show Values**—Shows database values associated with the item.
   - **Custom Values**—Supplies an empty field for inputting custom values. Click the check mark to add them to the list of values. You can also display values from a previously saved custom list or values loaded from a file.

4. **Select the values to include in the filter definition in the Values list**.
   Individually select values or click **Select all** and deselect the values you don’t want to include. Only selected items are applied to the filter definition.

5. **When the correct values are highlighted in the Values list, click OK**.
   An indicator is added to the Filter line and the filter is immediately applied to the data set.

Removing a Results Filter

➢ To remove a local filter, choose an item on the Filter line and select **Results**, then **Remove**.

Setting Compound Filters

In some cases, you may want to set two distinct filters on the same Request item or create compound constraints using more than one item.

Use the Filter line to build compound filter expressions. The Filter line enables you to apply more than one filter to one item, or create compound conditions dependent on more than one constraint.
Drag an item to the Filter line more than once and apply different logical operators to create a complex constraint. For example, to retrieve dollar values greater than $100 or less than or equal to $10, drag the Amount item to the Filter line twice, and set two separate filters. An AND operator is placed between the icons on the Filter line and the data set is constrained by both conditions.

Similar logic can be applied using two filters and substituting the OR operator. For example, to retrieve customers in the Cleveland area and those who have purchased more than $100,000 worth of goods, you can place a filters such that \( \text{City} = \text{Cleveland} \) and \( \text{SUM}(\text{Sale Amount}) > $100,000 \) and join them with the OR operator.

**Note:** The second instance of an item on the Filter line displays a “_2” next to the item name.

To create compound Filter line expressions:

1. **Add two or more items to the Filter line and apply individual filters using the Filter dialog box.**
   
   An AND operator is displayed between each item on the Filter line.

2. **In the Query section only, click the small arrow at the left edge of the Filter line.**
   
   The Filter line is adjusted to display the Filter line control buttons.

3. **On the Filter line, select filter controls to complete the equation.**
   
   - To switch Boolean operators AND and OR, double-click the operator to change.
   - To enclose suboperations, select the items you want to enclose and click the parentheses button. To remove parentheses, select a parenthesis and click **Remove** on the shortcut menu.
   - With an item selected on the Filter line, click **Var** to make the filter variable.

   When using compound filters, verify that the expression delivers the correct results.

**Tip:** The following syntax rules apply to all Filter line expressions. When creating a compound filter, be sure to verify that the expression is delivering the correct results.

- The AND operator retrieves data that meets both conditions. To retrieve data which satisfies either of two conditions, use the OR operator.
- By default, equations are solved from left to right, with enclosed sub-operations evaluated first. AND is evaluated before OR.
- Sub-operations allow you to override the default evaluation order, and may be required for certain operations involving both AND and OR operators.

### Setting Variable Filters

A variable filter is a filter in the Query section that is resolved only when a query is processed. At that time the user is prompted to select or enter filter values and complete the constraint.
You can use variable filters in standardized Interactive Reporting document files and distribute them to users to supply different filter values for each process.

Variable filters work particularly well with custom lists. If a custom list has been saved with a variable filter, the user can respond to the prompt by selecting a value from the custom list.

For example, you may use an Interactive Reporting document file monthly to monitor inventory levels. Each time you use the Interactive Reporting document file, you run it separately for each product line you carry. You can accelerate the process by making the filter variable on the product line item, and create a custom values list. Each time you process the Interactive Reporting document file, you can select a new product line without redefining filters.

To set a variable filter:

1. If necessary, add a topic item to the Filter line and define a filter.
   
   If you are providing a custom list, make the values available to the list of values.
   
   Note: You must select at least one value in the list to save the filter, even if no custom list is provided and the user clicks Show Values to choose from database values. This selection does not influence the values available to the variable filter, which offers all values in the database or in the custom list.

2. Select a topic item on the Filter line.

3. Choose Query, then Variable Filter.

The Filter item is displayed with a V(1) beside the item name to indicate it is a variable filter.

If other items are set to variable filters, they are displayed with V(2), V(3), and so on, to indicate the order in which the user is prompted to respond to Variable Filter when the query is processed.

**Customizing Filters Options**

In some cases, it may be preferable to disable or even remove such features as Include Null or the Custom Values buttons.

Note: Customizing affects one filter. Setting global filter preferences that restrict the options available throughout a distributed data model is an advanced feature not covered in this book.

To customize filter options:

1. Select an item on the Filter line and choose Query, then Customize Filter.

   Customize Filter is displayed.

2. Customize the filter as follow and click OK.

   - Title—Add a title or text to the filter dialog to instruct users on how to set the filter or on what the values mean.
Filtering Table Report Data

You can set filters on columns in a table report to filter the data displayed. Table section filters are applied in addition to the filters set in the Results section. Filters set in the Table section are automatically propagated to the other reports that inherit their data set from the Table. You can always suspend or delete filters to return data to the display and make it available for reporting. The Status bar displays the total number of rows in the table report. You can apply one filter only per column.

To filter data in a Table report:

1. Double-click the column in which to place a filter.
   The Filter dialog box is displayed.
2. Select a logical operator from the shortcut list.
3. Complete a filter definition by supplying constraining values.
   - Create a Customs Values list by supplying a value (or values, separated by commas) in the Edit field, and clicking the check mark.
   - You can also click Show Values to display column values and select one or more values depending on the comparison operator.
4. When the values are highlighted in the values pane, click OK.
   The filter is applied to the column and the column name is added to the Filter line.

To remove a filter in a Table report, select the filter item that you want to remove, and choose Remove on the shortcut menu.

To remove all filters in a Table report, select Filter on the Filter line and choose Remove on the shortcut menu.

Multiple Filters and the Meta Topic

Filters are a very important part of the querying process, helping to refine requests, filter unneeded information, and reduce system resource consumption to manageable levels. One way to further increase the reach of filters in a managed query environment is to think of them in terms of both the query and the underlying Data Model, and of ways in which fixed and variable filters may be used together.
Fixed filter values are defined with a query, and remain static until changed manually. Variable filters prompt for new values each time a query is processed. In certain instances you may want to apply both types of filter to one item in order to restrict the available variable filter values and simplify users' choices, or to grant users the latitude of variable filters only within a prescribed range.

For example, consider users of a large data warehouse or legacy system containing sales data going back for many years. For practical or security reasons the administrator might choose to confine some users' access to information dating after 1990. However, it is still important for these users to query the database on-the-fly, and retrieve a variety of data for analysis.

This situation requires two filters, one fixed and one variable, on one topic item. The fixed filter qualifies the values set available to the user in the variable filter. The fixed filter on a date column item instructs the server not to return any values < 1/1/90. The variable filter on the same date column subsequently furnishes a list of server values > 12/31/89 for the user to choose when the Showing Values command is invoked in Filter Controls.

The Two-Tier Strategy

Double-filters can be implemented by dragging an item to the Filter line twice. However, one deficiency to this method is that when applying multiple filters on one item, browse level settings are not effective; that is, the server values retrieved are not constrained by another filter on the same item, because both filters are defined using the same pool of server values. Nevertheless, an elegant solution to this issue is available through the use of the meta topic, which enables the administrator to establish filters at different logical levels.

Meta topics are logical views which overlay the original tables-and-joins model of a database. Items that display in meta topic view inherit any modifications made to the original items in the underlying physical topic view. In this instance, a fixed filter can be applied to the date item in the original database topic. Once a meta topic, which includes the date item, is created to serve as the user view of the Data Model, a variable filter is applied to the date item in the meta topic.

Since the fixed filter is applied to the original date item, the meta topic date item incorporates this filter transparently. The values excluded by the fixed filter are not associated within the new meta topic, and are not returned by the Show Value feature for fixed or variable filters applied to the new meta topic item. Note also that Filter Browse Level settings need not be modified under this scenario because the filters in question are applied to different logical layers.

Filtering Computed Items

In the Query section, you can set a filter on a computed item that resides on the Request Line and invokes an SQL HAVING clause when you process the query.

When you create a computed item, its only reference is through the Request Line. In the same way that you can add an item to Request Line by drag and drop, you can drag and drop a Computed Item onto the Filter Line to prompt for Filter. If the item uses an aggregate function, it displays to the right of all other filters, indicating its use in the HAVING clause of the SQL statement.
To create and set a filter on a computed item in the Query section

1. Click an item on the Request line and select Query, then Add Computed Item.
   Computed Item is displayed.

2. Enter the information in Computed Item and click OK.
   The computed item displays on the Request line under a new name.

3. If the Filter line is not already displayed, click the Filter button on the Section Titlebar.

4. Drag the computed item from the Request line and drop it in the Filter line.
   Filter is displayed.

5. Optional: Enter a descriptive name for the filter in the Name field.
   The name is displayed on the filter line icon and can help differentiate multiple filters set on the same item. The name is also used when scheduling an Interactive Reporting document file.

6. Select an arithmetic or logical operator from the list to begin defining a filter expression.

7. Define a pool of potential filter values by entering a filter value in the edit field and clicking the check mark to add each to the values list pane.
   - The Show Values command is not available because Interactive Reporting cannot retrieve a values list for a computed or aggregated request item from the database.
   - Click Custom Values to supply the values in the edit field and click the check mark to add them to the values pane. You can also display values from a custom list previously saved with the Interactive Reporting document file or loaded from a file.
   - If you are familiar with SQL, click Custom SQL and type a SQL Where clause to include in the query statement. You can open the Custom SQL dialog box to check the complete statement logic.

8. Select the values to include in the filter definition in the values pane.
   Select values individually or click Select all and then deselect those you do not want to include. Only selected items are applied to the filter definition.

9. When the correct values for the filter expression are highlighted in the values pane, click OK.
   An indicator is added to the Filter line. In the Results section, the filter is applied to the data set immediately. In the Query section, the filter is applied when you process the Query.

   When placing a filter on a Request item computed with a data function in the Query section, a divider is displayed on the Filter line, and the filter icon is placed to the right of the divider. The divider indicates the filter is applied in the SQL Having clause.

10. Click Process to process the final results set.

**Customize Filter**

Use Customize Filter to control access to Filter Controls features. This can be especially useful when you distribute Interactive Reporting document files to end users. For example, if an
Interactive Reporting document file is to be distributed to novice users who must set a simple variable filter, it may be preferable to disable or even remove such features as Include Null or the Custom Values buttons.

Table 87  Customize Filter

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>Add a title or text to the Filter to instruct users on setting the filter, or describe the value.</td>
</tr>
<tr>
<td>Prompt</td>
<td>Supply explanatory comments or instructions.</td>
</tr>
<tr>
<td>Values</td>
<td>Disable the Show Values option to reduce database hits, or remove the Custom SQL or Custom Values options.</td>
</tr>
<tr>
<td>Option</td>
<td>Disable elements and lock in operators.</td>
</tr>
</tbody>
</table>
About Computed Items

When looking for answers to basic business questions, raw data cannot always help. For example, while a database might track dollar sales figures, sometimes this information is much more valuable with reference to more complicated calculations such as cost of sales or profit margin.

Administrators sometimes anticipate these requirements, designing the internal database structure so that it makes such calculations as the raw data is collected. But at times you might need information that no one has yet anticipated.

Computed data items are a means of generating new information, usually from data that is already stored in the database. If your database doesn’t offer particular information, use Interactive Reporting to compute it for yourself.

Computing New Data

Computations can be performed by your database server or by Interactive Reporting on your desktop. A computation does not add data to the database; instead, new data items are added or data items are recalculated in your data set.

Interactive Reporting provides different tools for computing data items. These tools allow you to:

- Build arithmetic expressions \( \text{Units} \times \text{Amount} = \text{Revenue} \)
- Build logical expressions \( \text{If} \ \text{Score} \geq 50 \ \text{Then} \ \text{"Pass" Else "Fail"} \)
- Build mixed expressions \( \text{Tax} = \text{Revenue} \times 0.35 \)
- Apply functions \( \text{(abs (Amount_Sold))} \)
Examples of computed items include:

- A Full_Name item that concatenates the values in the First_Name and Last_Name items.
- A Profit item derived by subtracting the Cost of Goods column from the Gross Revenues column.
- A Grade item that uses `if...else` logic to assign letter grades derived from test scores.
- A Sine item that computes an item for the sine of an angle.
- A Moving Average item that smooths a set of data points, i.e., removes irregular fluctuations in the general trend of data.

**Computed Items in Sections**

You can add computed items in the Query, OLAPQuery, Results, Table, Chart, and Pivot sections. This feature works similarly in all sections with a few differences.

In all sections, you can use computed items to create a completely new data item on the Request line from an already existing data item. Each value in the original data item is computed to produce a new value in the newly generated data item. A one-to-one correspondence exists between the original values and the derived computed values.

**Computed Items in the Query Section**

In the Query section, a computed item is a set of instructions to the database server. Interactive Reporting uses the computing power of the database server to perform calculations as it retrieves data from the database.

For this reason, the Query section allows you to use computed items in a way that is not possible in the other sections. Instead of creating a new data item, the new values simply replace the original values in the data item as they are retrieved from the database.

Additionally, you can compute items using any topic item in the data model and any scalar functions provided by your RDBMS.

**Adjusting Data Types**

Since computed items are new data items, you may want to confirm or change the new item’s data type to preserve the precision of a mixed-data type computations, or to change the way a data item is handled (for example, interpreting number as strings). This ensures the correct handling of data in server computations.

Attention to data types is most important when computing items in the Query section. Here the computation is performed on the database server, and the computed item may be handled with an unanticipated data type.

Local calculations (Results, Pivot) are handled internally, and adjustment between 16- and 32-bit integers, for example, can be handled safely using the automatic or number data type specification. For a list of supported data types, see below:
<table>
<thead>
<tr>
<th>Data type</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatic</td>
<td>A data type is determined automatically given the data type of the reference items and the computations performed.</td>
</tr>
<tr>
<td>Byte</td>
<td>Variable data type of length determined by a single byte of computer storage. Bytes can store numeric values from 0 to 255, or a single text character.</td>
</tr>
<tr>
<td>Date</td>
<td>Calendar date in server default format (typically mm/dd/yy).</td>
</tr>
<tr>
<td>Integer (16-bit)</td>
<td>Retains a 16-bit value (2 bytes). A 16-bit integer stores integer values from 0 to 16,777,216, and signed integers between +8,388,608 and -8,388,608.</td>
</tr>
<tr>
<td>Integer (32-bit)</td>
<td>Retains a 32-bit value (4 bytes). A 32-bit integer has a range of 0 to 4,294,967,296 if unsigned. If signed, -2,147,483,648 to 2,147,483,647.</td>
</tr>
<tr>
<td>Packed Real</td>
<td>Real numbers packed for use with EDA middleware. The results in Interactive Reporting are the same as real numbers.</td>
</tr>
<tr>
<td>Real</td>
<td>Decimal numbers up to 5 positions right of the decimal.</td>
</tr>
<tr>
<td>String</td>
<td>Text strings to a maximum length of 256 characters.</td>
</tr>
<tr>
<td>Time</td>
<td>Time in format set by user preference.</td>
</tr>
<tr>
<td>TimeStamp</td>
<td>Date/time combination in format set by user preference.</td>
</tr>
</tbody>
</table>

**Computed Items in the Results and Reporting Sections**

In the Results and reporting sections, Interactive Reporting performs computations on the desktop. The computations involve only the data in your Results set or on the surface of a reporting section.

In these sections you can only create new computed items: you cannot modify original data items retrieved directly from the database. For the same reason, scalar functions used to compute items in these sections are provided by the Interactive Reporting, rather than the RDBMS.

Computed items in the Results and reporting sections differ in two respects:

- In the Results and Table sections, reference items are filtered to the items that are displayed on the Request line.

- In the remaining reporting sections (excluding the Report Designer section), reference items are limited to the items placed in the Fact pane in the data layout. Computations in these sections work on the aggregated cell values that make up the core of the report. To perform computations on data before it is aggregated, compute the new item in Results.

- In the Report section, the break totals of a table can be calculated

**Computed Items in the Report Section**

In the Report section, both report totals and break totals can be calculated based on the column’s data function applied to surface values, or the underlying formula of the column.
Underlying values refer to values from the original results section. Surface values refer to values in the actual report section. The two approaches produce values that may be displayed incongruous with the values in the report.

To understand this difference between underlying and surface values, consider a simple table with two values of 20 and 30. Each of these is already a total of underlying values (20 = 8 + 12 and 30 = 10 + 20). An average of the underlying value yields the result of 12.5 = (8 + 12 + 10 + 20) / 4). An average of the surface values yields the result 25 = (20 + 30) / 2).

In the following example, both the Computed and Computed2 columns have the Average data function applied. The Computed column is calculated using the data function applied to the surface values. The Computed2 column is calculated using the underlying values:

<table>
<thead>
<tr>
<th>Store</th>
<th>Amount</th>
<th>Computed</th>
<th>Computed2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer Central</td>
<td>1,982.66667</td>
<td>1,982.66667</td>
<td>1,982.66667</td>
</tr>
<tr>
<td><strong>Total Computer Cen</strong></td>
<td><strong>1,982.66667</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Computer City</td>
<td>1,883.53333</td>
<td>1,883.53333</td>
<td>1,883.53333</td>
</tr>
<tr>
<td><strong>Total Computer City</strong></td>
<td><strong>1,883.53333</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Computer Super Sbo</td>
<td>1,784.4</td>
<td>1,784.4</td>
<td>1,784.4</td>
</tr>
<tr>
<td><strong>Total Computer Sup</strong></td>
<td><strong>1,784.4</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Computer Town</td>
<td>892.2</td>
<td>892.2</td>
<td>892.2</td>
</tr>
<tr>
<td><strong>Total Computer Tow</strong></td>
<td><strong>892.2</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Computer World</td>
<td>1,784.4</td>
<td>1,784.4</td>
<td>1,784.4</td>
</tr>
<tr>
<td><strong>Total Computer Woi</strong></td>
<td><strong>1,784.4</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Discount Electronics</td>
<td>892.2</td>
<td>892.2</td>
<td>892.2</td>
</tr>
<tr>
<td><strong>Total Discount Elect</strong></td>
<td><strong>892.2</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Grapevine Peripherals</td>
<td>1,189.6</td>
<td>1,189.6</td>
<td>1,189.6</td>
</tr>
<tr>
<td><strong>Total Grapevine Per</strong></td>
<td><strong>1,189.6</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hi-Tech Mart</td>
<td>1,784.4</td>
<td>1,784.4</td>
<td>1,784.4</td>
</tr>
<tr>
<td><strong>Total Hi-Tech Mart</strong></td>
<td><strong>1,784.4</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Power Computers</td>
<td>1,487</td>
<td>1,487</td>
<td>1,487</td>
</tr>
<tr>
<td><strong>Total Power Compu</strong></td>
<td><strong>1,487</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wolfe’s Disc. Electric</td>
<td>2,200.76</td>
<td>2,200.76</td>
<td>2,200.76</td>
</tr>
<tr>
<td><strong>Total Wolfe’s Disc.</strong></td>
<td><strong>2,200.76</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>15,881.16</td>
<td>15,881.16</td>
<td>1,625,78667</td>
</tr>
</tbody>
</table>

If you are calculating totals and break totals for an Interactive Reporting document file (BQY) created in Release 8.3.2 and early (legacy computed items) you can still calculate totals based on the column’s data function applied to surface values, or the underlying formula of the column.

For legacy computed item columns, the “True Total Computation” is changed to “Legacy Total Computation”.

If the user selects “Legacy Total Computation” from the speed menu (essentially trying to turn it off), the user will get an alert dialog indicating that the change cannot be undone (“This action cannot be undone. Are you sure you want to use True Total Computations for Total and Break Totals?” … Continue/Cancel). If the user chooses “Continue”, then “True Totals” will be enabled for both Totals and Break Totals of this column. The next time they display the context menu it will show “True Total Computation” option in place of “Legacy Total Computation”. The user may no longer apply the legacy behavior to this column.
Computed Items in the Pivot Section

When you analyze and interpret Pivot section computed item break totals, two methods are available for you to present the data. These methods include:

- Break total cell results derived from the computed item formula applied to the detail cell
- True Computed Item totals

When determining which method to use, look at how the two types of totals differ from each other. The former method is derived from the formula used to calculate the detail cell.

True computed item totals use aggregation according to the specified data function and they never rely on the computed item total formula. Depending on what you are trying to achieve, each method will show different totals.

You can see how the two types of totaling computed items differ in the example below. The first part of the example shows what happens when the totals are derived from the computed item formula used in the detail cells.

Detail cell data is shown at the City level and break totals are shown when the State changes.

<table>
<thead>
<tr>
<th></th>
<th>Units</th>
<th>Computed</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA</td>
<td>Burbank</td>
<td>2,395</td>
</tr>
<tr>
<td></td>
<td>Glendale</td>
<td>990</td>
</tr>
<tr>
<td></td>
<td>Oakland</td>
<td>910</td>
</tr>
<tr>
<td></td>
<td>San Diego</td>
<td>7,400</td>
</tr>
<tr>
<td></td>
<td>Westwood</td>
<td>695</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>12,390</td>
</tr>
<tr>
<td>NY</td>
<td>New York</td>
<td>4,640</td>
</tr>
<tr>
<td></td>
<td>Rochester</td>
<td>1,815</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>6,256</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>18,645</td>
</tr>
</tbody>
</table>

In this example, the _Sum_ data function has been applied. The “Computed” column is defined by the following formula:

\[(\text{Units} \mod 50) + 1\]

where “\(\mod\)” represents modulo (remainder) operator. In other words, the formula is defined as:

Integer remainder of \(\left(\text{“Unit column cell value”} / 50\right) + 1\)

For the Unit column values for each city within a state, the formula works as expected. For example, in the Oakland, CA cell, the formula is:

Units 910
Modulo (remainder) of 910 / 50 = 10
Add 1 to assign a value of 11 (shown above).

For the California “Total” row, the value shown is 41, which is the result of the following formula:

Total “Units” for California = 12390
Modulo of 123900/50 = 40
Add 1 to assign a value of 41
The Modulo of 41 is not the sum of the displayed cell values for all cities in California, instead it is the modulo formula applied only to the cell containing the “Unit” column city total for California.

If you expected to see a break cell total value of 145, you need to use the True Computed Item feature total, which would reference the displayed values in the detail cells (this example assumes a *Sum* data function):

\[
46 + 1 + 11 + 1 + 46 \ (= 145)
\]

**Using Surface Values**

Interactive Reporting enables you to use underlying or surface values when working with totals. Underlying values refer to values from the original results section. Surface values refer to values in the actual report section. The two approaches produce values that may be displayed incongruous with the values in the report.

To understand this difference between underlying and surface values, consider a simple pivot table with two values of 20 and 30. Each of these is already a total of underlying values (20 = 8 + 12 and 30 = 10 + 20). An average of the underlying value yields the result of 12.5 = (8 + 12 + 10 + 20) / 4). An average of the surface values yields the results 25 = (20 + 30) / 2).

**Note:** Average and Count aggregation Data Functions are not evaluated in True Total mode unless the Use Surface Values property is also enabled (see Use Surface Value above). If Use Surface Values is not enabled, the Average and Count aggregation are calculated using the count of the underlying Table/Result section data values instead of the displayed Pivot values.

**Note:** By default, the Surface Values feature is deactivated. To activate surface values, choose Pivot, then Use Surface Values.

**Computed Items and Data Functions**

Computed items and data functions are fundamentally different, and the functions available in the Computed Item dialog box do not calculate data in the same way as data functions.

- Computed items calculate a fresh value for each original value, based on the computation (for example, *Revenue* calculated from *Price* and *Units Sold*). The new values are part of a new data item or replace the original values. Computed items *never* reduce the original number of records.

- Data functions, by contrast, summarize groups of database records and replace the original values with new summary data. Because data functions summarize values, the number of records are frequently reduced.
Pivot Options

The Pivot Options tab allows you to enable and disable surface values, how to handle true computed item totals, and how to treat null fact values.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Surface Values</td>
<td>Recalculates the values in the visible cells or surface of the pivot rather than the values in the Results section.</td>
<td></td>
</tr>
<tr>
<td>True Computed Item Totals</td>
<td>Recalculates break totals so that each total value is equal to the sum of their displayed detail cells. True computed item totals use aggregation according to the specified data function and does not rely on the computed item total formula. If you disable this option, the break total cell values are derived from the computed item formula applied to the detail cell. Average and Count aggregation Data Functions are not evaluated in True Total mode unless you enable Use Surface Values. If you do not enable Use Surface Values, the Average and Count aggregation are calculated using the count of the underlying Table/Result section data values instead of the displayed Pivot values.</td>
<td></td>
</tr>
<tr>
<td>Enable Null Facts In Computed Items</td>
<td>Evaluates a null fact value (an empty cell value) as a zero fact value for non-Moving functions. For Moving Function calculations, where the presence of all displayed Fact cell must be considered in calculations, the following behavior occurs when null fact values are considered as zero. The default option to evaluate and show null values is disabled.</td>
<td></td>
</tr>
</tbody>
</table>

Adding Computed Items

In the Query section, a computed item is a new data item derived from calculations. In the Results, OLAPQuery, Pivot, Chart and Report sections, you add computed items by building equations to compute data items, or by applying functions to existing data items.

Computed items are like normal data items, and can be included in reports or reused to compute other data.

For example, you can modify the Amount Sold item by building an equation around it, multiplying by the Unit Price item and renaming the resulting item ‘Revenue’. You can also apply a scalar function such as Cume to Amount Sold and return each individual value as a cumulative running total, or simply multiply Amount Sold by the local tax rate to find the tax owed on each sale.

To create (or modify) a computed item:

1. Select Add Computed Item from a Section menu (for example, Query, Results, and so on).
   The Computed Item or Modify Item dialog box is displayed.

2. In the Name field, type a name that describes the computation.
   The default name is Computed, which is numbered sequentially if there is more than one. If you assign a name to a computed item that is identical to an existing scalar function name, the Interactive Reporting numbers the name starting with the number 2.

3. Define the new data item by building an expression in the Definition text box.
   - Use the operator buttons to insert arithmetic and logical operators at the insertion point.
Click **Functions** to apply scalar functions using the Functions dialog box.

Click **Reference** to display the Reference dialog box, and select items to place in the equation.

You also can type any portion of the equation or the entire equation directly into the Definition text box using JavaScript. The names are case sensitive, and you must replace spaces in item names with underscores ("_").

4 If necessary, click the **Options** button to set a new data type for the item.

5 When the equation is complete, click **OK**.

The computed item is listed in the Results data layout and is displayed as a column in the results set.

In the Query section, the computed item is displayed on the Request line with its new name. In the Results section, computed items are displayed in the data layout in blue type.

### Operators

The following sections describe the operators available for creating computed items. Use the following guidelines as you add operators to your computation. Click the Operator buttons in the Computed Item dialog box to add arithmetic or logical operators to a computation in the Definition text box. Operators are added at the insertion point. You can use any of the following types of operators:

- **Arithmetic Operators**
- **Comparison Operators**
- **Statements**
- **Logical Operators**

**Tip:** Use the following guidelines as you add operators to your computation:

- Type the word null (no quotes) into the Expression text box to represent null values.
- Enclose all text string constant values and date constant values entered in expressions in single quotes. (Numbers can be entered without quotes.)
- To join items with a space or other character, reference or type items and strings into the Expression text box and join them with the + operator (for example, `City + ',' + State`). To join without additional characters, use the **Concat** function.
- In division operations, the divisor may not be null or equal to zero. If a data item serves as the divisor in an expression (for example, `5000 / Units_Sold`) and includes null or zero values, first create a computed item using if/else logic to remove null and zero values, and then compute the item containing the division operation.
- Two date items can be subtracted, but not added. The Add Month function adds an integer value to a date.
- You cannot nest functions inside the **Sum**, **Cume**, **Chr**, and **Breaksum** functions.
Note: For a detailed description of JavaScript operators, see the Hyperion JavaScript Reference.

Arithmetic Operators

Arithmetic operators take numerical values (either logical or variables) as their operands and return a single numerical value.

Table 89 Arithmetic Operators

<table>
<thead>
<tr>
<th>Operator</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>Add</td>
<td>Used at both the server level and the local metatopic level for all sections.</td>
</tr>
<tr>
<td>-</td>
<td>Subtract</td>
<td>Used at both the server level and the local metatopic level for all sections.</td>
</tr>
<tr>
<td>*</td>
<td>Multiply</td>
<td>Used at both the server level and the local metatopic level for all sections.</td>
</tr>
<tr>
<td>/</td>
<td>Divide</td>
<td>Used at both the server level and the local metatopic level for all sections.</td>
</tr>
<tr>
<td>(</td>
<td>Begin suboperations</td>
<td>Used at both the server level and the local metatopic level for all sections.</td>
</tr>
<tr>
<td>)</td>
<td>End suboperations</td>
<td>Used at both the server level and the local metatopic level for all sections.</td>
</tr>
<tr>
<td>++</td>
<td>Increment</td>
<td>Used at both the server level and the local metatopic level for all sections except the Query section.</td>
</tr>
<tr>
<td>--</td>
<td>Decrement</td>
<td>Used at both the server level and the local metatopic level for all sections except the Query section.</td>
</tr>
<tr>
<td>Mod (%)</td>
<td>Modulus</td>
<td>The modulus operator returns the remainder of dividing var1 by var2.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For example, 5% 4 returns 1.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Used only at the local metatopic level.</td>
</tr>
</tbody>
</table>

Tip: If a computed item is displayed on a Request Line, and that item’s definition uses subtraction, such as "Mytable.Column1-5", a SQL error can occur. The exact error depends on the database, but the most common error would indicate an undefined name was used. Because databases allow hyphenated names, Interactive Reporting attempts to deal with such names intuitively. Thus, an item definition like "Mytable.Column1-5" is interpreted as a name. In order to ensure it is treated as subtraction, include a space on either side of the hyphen/subtraction operator. For example, entering the computed item definition as Mytable.Column1 - 5" ensure that correct SQL is generated.

Comparison Operators

A comparison operator compares its operands and returns a logical value based on whether the comparison is true. The operands can be numerical or string values. When used on string values, the comparisons are based on the standard lexicographical ordering. The comparison operators
in the following table are only available at the local metatopic level. For the examples in the table, \textit{var1} has been assigned the value 3, and \textit{var2} has been assigned the value 4.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>==</td>
<td>Returns true if the operands are equal, for example, \texttt{3 == var1}</td>
</tr>
<tr>
<td>!=</td>
<td>Returns true if the operands are not equal, for example, \texttt{var1 != 4}</td>
</tr>
<tr>
<td>&lt;</td>
<td>Returns true if the left operand is less than the right operand, for example, \texttt{var1 &lt; var2}</td>
</tr>
<tr>
<td>&lt;=</td>
<td>Returns true if the left operand is less than or equal to the right operand, for example, \texttt{var &lt;= var2}, \texttt{var2 &lt;= 5}</td>
</tr>
<tr>
<td>&gt;</td>
<td>Returns true if the left operand is greater than the right operand, for example, \texttt{var2 &gt; var1}</td>
</tr>
<tr>
<td>&gt;=</td>
<td>Returns true if the left operand is greater than or equal to the right operand, for example, \texttt{var2 &gt;=}, \texttt{var1 &gt;= 2}</td>
</tr>
</tbody>
</table>

**Statements**

Executes a set of statements if a specified condition is true. If the condition is false, another set of statements can be executed.

**Note:** Statements are only available at the local metatopic level.

**Logical Operators**

Logical operators take Boolean (logical) values as operands and return a Boolean value. The logical operators below are only available at the local metatopic level.
Table 92  Logical Operators (Local Metatopic Level)

<table>
<thead>
<tr>
<th>Logical Operator</th>
<th>Description</th>
</tr>
</thead>
</table>
| **AND (&&)**     | Connects two conditional expressions and retrieves records only if each expression is true. Computed items will not be retrieved if any condition belonging to a conditional expression is false. The AND logical operator is usually nested within another conditional expression, for example, expressions which use if and else statements. For example:  
  
  ```java
  if ((OS == 'Windows') && (Item_Type == 'Modem')) {'Windows'} else {'other'}
  ```
| **OR (||)**      | Specifies a combination of expressions and retrieves records that include at least one of the expressions. For example, if one of the words is Washington or Oregon, every record with the expression “Washington” and every record with the word “Oregon” is included. Typically the OR (||) is nested within other conditional expressions, for example, expressions which use if and else logical operators. For example to assign Washington and Oregon to the "Northwestern Region" and all other states to "Other Regions", you would enter:  
  
  ```java
  :if ((State = = 'Washington') || (State == 'Oregon')) {'Northwestern Region'} else {'Other Regions'}
  ```
| **NOT (!)**      | Computes and shows items more accurately stated in a negative way. In effect, all records are retrieved except those that fulfill the conditional expression. You enter the conditional expression with the NOT (!) logical operator preceding the conditional expression. The conditional expression can be a simple value or nested within other conditional expressions, for example, expressions using AND and OR. A combined condition expression that uses NOT is true if the conditional expression following NOT is false. A combined conditional expression is false if the conditional expression following NOT is true. For example, suppose you are looking to list all states that are not in the Northwestern region. In this case, you would enter the conditional expression:  
  
  ```java
  if ( ! (State = = 'Northwestern Region')) {'Other Regions'}
  ```

### Referencing Topics in Computed Items

Use the Reference dialog box to select Request line items to place in the equation, or use as function arguments.

- To show topic items associated with a topic, select the topic in the left pane. Topic items are displayed in the right pane.

### Using Functions

These topics lists the functions available in Interactive Reporting:

- “Scalar Functions” on page 334
- “Data Functions” on page 342
- Teradata Standard Functions
Scalar Functions

The available scalar functions include:

You can use a variety of scalar functions for adding computed data items in the Pivot, Chart, Tables and Results sections.

Scalar functions (in contrast to Data Functions) do not aggregate data or compute aggregate statistics. Instead, scalar functions compute and substitute a new data value for each value associated with a data item.

The following scalar functions are available:

“Conditional Scalar Functions” on page 334
“Date Scalar Functions” on page 335
“Math Scalar Functions” on page 336
“Numeric Scalar Functions” on page 339

Note: In the tables for Conditional, Date, Math, String and Trend functions below, the variables n, s, d, and exp (and val) represent data items and columns (State, Amount Sold) or actual values (’NY’, 6000) as arguments to scalar functions, and indicate number, string, date, or variable types, respectively.

Note: The variable c indicates that only a data item reference may be used, and not a constant value. If constant values are substituted for data items, dates and text strings must be enclosed in single quotes. Examples in the tables that follow use a mixture of constants and data items, which are generally interchangeable.

Conditional Scalar Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Decode</strong> (c,exp,val, exp,val...def)</td>
<td>Compares value of item c to one or more expressions exp, and returns the value val matched to each expression, or a default def.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>decode (region, ’South America’, ’Priority 1’, ’Asia’, ’Priority 2’, ’Europe’, ’Priority 3’, null)</td>
<td>substitutes priority values for the specified regions and leaves other regions null.</td>
</tr>
</tbody>
</table>
Function | Description
--- | ---
Nvl (exp1,exp2) | Returns exp2 if null, and exp1 otherwise.

Example:

```
nvl (Phone_No,'Phone_No','Not Recorded')
```

returns "Not Recorded" when no telephone number is on record for a customer. If the function is included in a computed item expression, only two parameters are allowed: exp1 and exp2.

### Date Scalar Functions

Table 94  Date Scalar Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddMonths (d,n)</td>
<td>Adds n months to date d.</td>
</tr>
<tr>
<td>Example:</td>
<td>AddMonths ('5/13/99',4) = 9/13/99</td>
</tr>
<tr>
<td>DayOfMonth (d)</td>
<td>Returns the day of month for date d.</td>
</tr>
<tr>
<td>Example:</td>
<td>DayOfMonth ('11/02/99') = 2</td>
</tr>
<tr>
<td>LastDay (d)</td>
<td>Returns date of the last day of the month containing date d.</td>
</tr>
<tr>
<td>Example:</td>
<td>LastDay ('12/6/9') = 12/31/99</td>
</tr>
<tr>
<td>MonthsBetween (d1,d2)</td>
<td>Returns the number of months between dates d1 and d2 as a real number (fractional value).</td>
</tr>
<tr>
<td>Example:</td>
<td>MonthsBetween ('12/5/99','5/6/99') = 6.9677</td>
</tr>
<tr>
<td>NextDay (d,s)</td>
<td>Returns the date of the first weekday s after date d. If s is omitted, add one day to d.</td>
</tr>
<tr>
<td>Example:</td>
<td>NextDay ('12/16/99','Monday') = 12/22/99</td>
</tr>
<tr>
<td>Sysdate ()</td>
<td>Returns the current system date and time for each record in item c.</td>
</tr>
<tr>
<td>Example:</td>
<td>Sysdate() = 2/11/96 19:54:36</td>
</tr>
</tbody>
</table>
### ToChar (d/n, 'f' or "f")
Converts the date or number d/n into a string in the specified format. This function does not change the data, but rather the item data type. The results cannot be computed mathematically.

**Example:**
ToChar (‘05/08/06’, "mmm-yy") = May-06

If you are referencing a Date or Number column for the first argument (Date field), single quotes are not required. If you are passing a data value, single quotes are required.

A comma must immediately follow the first argument. In the second argument (Format field), single quotes or double quotes must enclose values.

### ToDate (s)
Returns date type in place of date-string s. This function does not change the data, but rather the item data type. The results can be computed mathematically.

**Example:**
ToDate (‘10/12/96’) = 10/12/96

**Note:** See Functions for Returning the Day of the Week for information on how to return the day of the week on which a given date falls.

### ToMonth (d)
Returns the referenced date and replaces the day and year part of the date with 15 and 1904 respectively. You can change the value to display as a month string by adding and applying a mmm date format or a month number by applying the date format of mm.

**Example:**
ToMonth (‘11/2/09 09:46:00 AM’) = 11/2/09 ~ Nov

### ToQtr (d)
Returns a string quarter value for each value of d.

**Example:**
ToQtr (‘11/02/06 09:46:00 AM’) = Q3

### ToYear (d)
Returns the integer year for each value of d. You can convert the year to display without commas by applying the 0 numeric format.

**Example:**
ToYear (‘11/02/06 09:46:00 AM’) = 2,006 ~ 2006

---

### Math Scalar Functions

**Table 95**  Math Scalar Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Abs (n)</strong></td>
<td>Returns the absolute value of number n.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Abs (-3) = 3</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>Atn (n)</td>
<td>Returns arc tangent of number $n$ radians.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Atn (1)</td>
<td>$0.7854$</td>
</tr>
<tr>
<td>Ceil (n)</td>
<td>Returns the smallest integer value greater than or equal to number $n$.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Ceil (5.6)</td>
<td>$6$</td>
</tr>
<tr>
<td>Cos (n)</td>
<td>Returns cosine of number $n$ radians.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Cos (0.5)</td>
<td>$0.8776$</td>
</tr>
<tr>
<td>Cosh (n)</td>
<td>Returns hyperbolic cosine of number $n$ radians.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Cosh (0.5)</td>
<td>$1.1276$</td>
</tr>
<tr>
<td>Count (c)</td>
<td>Returns the number of row values in $c$ (including nulls).</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Count (units)</td>
<td>tally of rows in units</td>
</tr>
<tr>
<td>Exp (n)</td>
<td>Returns $e$ ($2.718$) raised to exponential power $n$.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Exp (4)</td>
<td>$54.598$</td>
</tr>
<tr>
<td>Max (a,b)</td>
<td>Returns the larger of items $a$ and $b$ for each new value.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Max (7, 10)</td>
<td>$10$</td>
</tr>
<tr>
<td>Min (a,b)</td>
<td>Returns the smaller of items $a$ and $b$ for each new value.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Min (7, 10)</td>
<td>$7$</td>
</tr>
<tr>
<td>Mod (n,m)</td>
<td>Returns the integer remainder of number $n$ divided by number $m$. If $m$ is larger, the default value is $n$.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Mod (6,2)</td>
<td>$0$</td>
</tr>
<tr>
<td>Power (n,m)</td>
<td>Returns number $n$ raised to exponential power $m$.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Power (10,5)</td>
<td>$100,000$</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| **Round** \((n,m)\) | Returns \(n\) rounded to \(m\) decimal places. If \(m\) is omitted, round to 0 decimal places.  
**Example:**  
\[
\text{Round} \ (5.6178,2) = 5.62
\] |
| **Sign** \((n)\) | Returns indicator of -1, 0, or 1 if number \(n\) is variously negative, 0, or positive.  
**Example:**  
\[
\text{Sign} \ (-4) = -1
\] |
| **Sin** \((n)\) | Returns sine of number \(n\) radians.  
**Example:**  
\[
\text{Sin} \ (86) = -0.924
\] |
| **Sinh** \((n)\) | Returns hyperbolic sine of number \(n\) radians.  
**Example:**  
\[
\text{Sinh} \ (.5) = .5211
\] |
| **Sqrt** \((n)\) | Returns square root of number \(n\).  
**Example:**  
\[
\text{Sqrt} \ (81) = 9
\] |
| **Tan** \((n)\) | Returns tangent of number \(n\) radians.  
**Example:**  
\[
\text{Tan} \ (30) = -6.405
\] |
| **Tanh** \((n)\) | Returns hyperbolic tangent of number \(n\) radians.  
**Example:**  
\[
\text{Tanh} \ (45) = 1
\] |
| **Trunc** \((n,m)\) | Returns number \(n\) truncated to number \(m\) decimal places. The default value for \(m\) is 0.  
**Example:**  
\[
\text{Trunc} \ (56.0379,2) = 56.03
\] |
# Numeric Scalar Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Avg</strong> (numbers, break_col, break_value)</td>
<td>Returns the average (arithmetic mean) of values in a number column. The average includes null values when calculating the arithmetic mean.</td>
</tr>
<tr>
<td><strong>AvgNonNull</strong> (numbers, break_col, break_value)</td>
<td>Returns the average (arithmetic mean) of values in a number column, excluding null values.</td>
</tr>
<tr>
<td><strong>chr</strong> (n)</td>
<td>Returns string converted from ASCII numeric code $n$.</td>
</tr>
<tr>
<td><strong>ColMax</strong> (numbers, break_col, break_value)</td>
<td>Returns the largest value in a column of numbers.</td>
</tr>
<tr>
<td><strong>ColMin</strong> (numbers, break_col, break_value)</td>
<td>Returns the smallest value in a column of number.</td>
</tr>
<tr>
<td><strong>Count</strong> (numbers, break_col, break_value)</td>
<td>Counts and returns the number of rows in a column.</td>
</tr>
<tr>
<td><strong>CountDistinct</strong> (numbers, break_col, break_value)</td>
<td>Counts and returns the number of values in a column.</td>
</tr>
<tr>
<td><strong>CountNonNull</strong> (numbers, break_col, break_value)</td>
<td>Counts the number of rows in a column.</td>
</tr>
<tr>
<td><strong>CountNull</strong> (numbers, break_col, break_value)</td>
<td>Counts the number of rows in a column that contains null values.</td>
</tr>
<tr>
<td><strong>Cume</strong> (numbers, break_col)</td>
<td>Returns a cumulative running total for each value in a column of numbers.</td>
</tr>
<tr>
<td><strong>Next</strong> (c)</td>
<td>Returns the next row value of the referenced item $c$.</td>
</tr>
<tr>
<td><strong>Prior</strong> (c)</td>
<td>Returns the prior row value of the referenced item $c$.</td>
</tr>
<tr>
<td><strong>Sum</strong> (numbers, break_col, break_value)</td>
<td>Returns the total of a column of numbers.</td>
</tr>
</tbody>
</table>
## Picture Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Picture** (Image path, Tooltip (optional)) | Returns a BLOB image file (.JPEG, .GIF, .BMP, and .PNG). Image path refers to folder where the image is store. Tooltip is any descriptive text to associate with the picture image.  
**Example:**
```java
if (Amount_Sales >= 10000) {Picture("C:\graphics\smile.gif")}
else{Picture("C:\graphics\crying.gif")}
```

## Statistical Scalar Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Median** (numbers, break) | Returns the median of a column of numbers.  
**Median example**

| **Mode** (numbers, break_col) | Returns the most frequently occurring value in a column of numbers.  
**Mode example**

| **Percentile** (numbers, n, break_col) | Returns the \(n^{th}\) percentile of values in a column of numbers in ascending order.  
**Percentile example**

| **Rank** (numbers, break_col) | Returns the rank of a number in a column of numbers.  
**Rank example**
There is a restriction on setting a filter on an aggregate column (which “Rank” is considered). To set a filter on the column, insert a new table into the Interactive Reporting document file and drag the columns into the table data layout. This creates a “copy” of the columns which can have a filter set on it.

| **RankAsc** (numbers, break_col) | Returns the rank of a number in a column of numbers in ascending order.  
**RankAsc example**

| **StdDev** (numbers, break_col) | Estimates standard deviation based on a sample. The standard deviation is a measure of how widely values are dispersed from the average value (the mean). If your data represents the entire population, then compute the standard deviation using the StdDevp function.  
**StdDev example**

| **StdDevp** (numbers, break_col) | Calculates standard deviation based on the entire population given as arguments. The standard deviation is a measure of how widely values are dispersed from the average value (the mean). If your data represents a sample of the population, then compute the standard deviation using the StdDev function.  
**StdDevp example**

340  Computed Items
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Var** (numbers, break_col) | Estimates variance based on a sample. The Var function assumes that its arguments are a sample of the population. If your data represents the entire population, then compute the variance using the Varp function.  
Var example                                                                 |
| **Varp** (numbers, break_col)  | Estimates variance based on the entire population. The Varp function assumes that its arguments are the entire population. If your data represents a sample of the population, then compute the variance using the Var function.  
Varp example                                                                 |

### String Scalar Functions

#### Table 99 String Scalar Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Ascii</strong> (s)</td>
<td>Returns an ASCII code of the first symbol in the string. s.</td>
</tr>
<tr>
<td>Example:</td>
<td>Ascii (&quot;AZ&quot;) = 65</td>
</tr>
<tr>
<td><strong>Concat</strong> (s1, s2)</td>
<td>Returns text strings s1 and s2 concatenated.</td>
</tr>
<tr>
<td>Example:</td>
<td>Concat (&quot;interactive&quot;,&quot;reporting&quot;) = interactivereporting</td>
</tr>
<tr>
<td><strong>Initcap</strong> (s)</td>
<td>Returns string s with the first letter of each word capitalized, and remaining characters in lower case.</td>
</tr>
<tr>
<td>Example:</td>
<td>Initcap (&quot;santa fe&quot;) = Santa Fe</td>
</tr>
<tr>
<td><strong>Instr</strong> (s1, s2, n, m)</td>
<td>Returns position of mth occurrence of string s2 in string s1, beginning at position number n. If n is negative, the count is made backwards from the end of s1. If no values are found, 0 is returned.</td>
</tr>
<tr>
<td>Examples:</td>
<td>Instr (&quot;Mississippi&quot;,&quot;s&quot;,5,2) = 7</td>
</tr>
<tr>
<td></td>
<td>Instr ( City, 'a', -2, 1 )</td>
</tr>
<tr>
<td><strong>Length</strong> (s)</td>
<td>Returns character count of string s.</td>
</tr>
<tr>
<td>Example:</td>
<td>Length (&quot;Pittsburgh&quot;) = 10</td>
</tr>
<tr>
<td><strong>Lower</strong> (s)</td>
<td>Returns string s in lower case.</td>
</tr>
<tr>
<td>Example:</td>
<td>Lower (&quot;CD-Rom&quot;) = cd-rom</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Ltrim (s1,s2)</td>
<td>Trims string s1 from the left, up to the first character not included in string s2.</td>
</tr>
<tr>
<td>Example:</td>
<td>Ltrim (&quot;Mr. Jones&quot;, &quot;Mr. &quot;) = Jones</td>
</tr>
<tr>
<td>Replace (s1,s2,s3)</td>
<td>Returns string item s1 with all occurrences of string s2 replaced by string s3. The default for s3 deletes each occurrence of s2.</td>
</tr>
<tr>
<td>Example:</td>
<td>Replace (customer,&quot;Mrs.&quot;,&quot;Ms.&quot;) = replaces Mrs. with Ms. for all values of customer containing ‘Mrs.’</td>
</tr>
<tr>
<td>Rtrim (s1,s2)</td>
<td>Trims column string s1 from the right, up to the first character not included in string s2.</td>
</tr>
<tr>
<td>Example:</td>
<td>Rtrim (&quot;Columbus, OH&quot;,&quot;&quot;, OH&quot;) = Columbus</td>
</tr>
<tr>
<td>Substr (s,n,m)</td>
<td>Returns a portion of string s, m characters long, beginning at numeric position n. The default action for m includes all remaining characters.</td>
</tr>
<tr>
<td>Example:</td>
<td>Substr (&quot;312/989-9989&quot;,&quot;1&quot;,&quot;3&quot;) = <em>312</em></td>
</tr>
<tr>
<td>Translate (s1,s2,s3)</td>
<td>Returns string s1, with each character contained in string s2 replaced by the corresponding characters in string s3.</td>
</tr>
<tr>
<td>Example:</td>
<td>Translate (&quot;12/5/97 3:48:22&quot;,&quot;/&quot;,&quot;-</td>
</tr>
<tr>
<td>Upper (s)</td>
<td>Returns string s in upper case.</td>
</tr>
<tr>
<td>Example:</td>
<td>Example: Upper (&quot;st.&quot;) = ST.</td>
</tr>
</tbody>
</table>

**Data Functions**

Data functions compute aggregate values, including averages, maximums, counts and other statistics. These functions summarize groupings of data. You can use data functions to aggregate and compute data from the server before it reaches the Results section, or compute different statistics for aggregated Results totals and report items.

The effects of data functions are most dramatic in the Query section. For example, Dollars is an item of sales transaction records for your stores in London and Madrid. You can apply a data function to this item, consolidate the data, and calculate sum totals, average sale values, number counts of individual sales records, or minimum sale values with respect to each city, once the data is retrieved to Results.

Note: Although data functions are relatively standard, there may be additional data functions available to you in the Query section, depending on your RDBMS.
To apply a data function, select the item or column, and choose **Data Function** on the Query, Pivot, Chart, Results or Table menu.

You can also select the item or column, activate the shortcut menu, point to Data Function, and select (function).

**Note:** In report sections, you can apply data functions only if you select a single fact column. You can also change the way data functions are computed.

**Tip:** You cannot use Data Functions in the Table section.

The following table lists the available data functions.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>Returns unaggregated values as stored in the database. This is the default in Query.</td>
<td>Query</td>
</tr>
<tr>
<td>Sum</td>
<td>Returns sum of underlying values. This is the default in Results and report sections.</td>
<td>All</td>
</tr>
<tr>
<td>Average</td>
<td>Returns average of underlying values.</td>
<td>All</td>
</tr>
<tr>
<td>Non-Null Average</td>
<td>Returns average of underlying values; null values excluded.</td>
<td>Pivot, Chart, Report</td>
</tr>
<tr>
<td>Minimum</td>
<td>Returns lowest of underlying values.</td>
<td>All</td>
</tr>
<tr>
<td>Maximum</td>
<td>Returns highest of underlying values</td>
<td>All</td>
</tr>
<tr>
<td>Count</td>
<td>Returns number of underlying values.</td>
<td>All</td>
</tr>
<tr>
<td>CountDistinct</td>
<td>Returns the number of distinct values in a column. This function is not supported by all database servers.</td>
<td>Query</td>
</tr>
<tr>
<td>Null Count</td>
<td>Returns number of nulls among underlying values.</td>
<td>Pivot, Chart, Report</td>
</tr>
<tr>
<td>Non-Null Count</td>
<td>Returns number of underlying values; null values excluded.</td>
<td>Pivot, Chart, Report</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>Returns standard deviation of values. This function is not supported by all database servers.</td>
<td>Query</td>
</tr>
<tr>
<td>Variance</td>
<td>Returns variance of values. This function is available through Oracle servers only.</td>
<td>Query</td>
</tr>
<tr>
<td>Weight</td>
<td>Use for computing weighted items in Pivot reports.</td>
<td>Query</td>
</tr>
<tr>
<td>% of Column</td>
<td>Returns sum of all underlying values as a percentage of their respective surface column.</td>
<td>Pivot</td>
</tr>
<tr>
<td>% of Row</td>
<td>Returns sum of underlying values as a percentage of their respective surface row.</td>
<td>Pivot</td>
</tr>
<tr>
<td>% of Grand</td>
<td>Returns sum of underlying values as a percentage of all surface values in the report.</td>
<td>Pivot, Chart</td>
</tr>
<tr>
<td>% of Category</td>
<td>Returns group total percentage of the selected value.</td>
<td>Report</td>
</tr>
<tr>
<td>Increase</td>
<td>Calculates the increase between the previous two rows or columns.</td>
<td>Pivot</td>
</tr>
<tr>
<td>%Increase</td>
<td>Calculates the percentage increase between the previous two rows or columns.</td>
<td>Pivot</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
<td>Availability</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Title</td>
<td>Returns column names</td>
<td>Report</td>
</tr>
</tbody>
</table>

**Tip:** Null values are empty values for which no data has been specified; null values are not equal to zero.

## Teradata Standard Functions

Interactive Reporting supports the following standard Teradata functions:

For a complete listing of Teradata functions, see the *Teradata Data SQL Reference, Functions and Operators*.

- "Teradata Conditional Functions" on page 344
- "Teradata Date Functions" on page 344
- "Teradata Miscellaneous Functions" on page 345
- "Teradata Numeric Functions" on page 345
- "Teradata String Functions" on page 345
- "Teradata System Functions" on page 346
- "Teradata Trigonometric Functions" on page 346

### Teradata Conditional Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Return Type</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>NullIf</td>
<td>NULLIF returns NULL if its arguments are equal. Otherwise, it returns its first argument, scalar_expression_1. NULLIF is a shorthand expression for the following full CASE expression: [\text{CASE WHEN scalar_expression_1=scalar_expression_2 THEN NULL ELSE scalar_expression_1 END}]</td>
<td>String</td>
<td>String, comparison string</td>
</tr>
<tr>
<td>Coalesce</td>
<td>Returns the second argument if the first argument is null.</td>
<td>String</td>
<td>String 1, String 2</td>
</tr>
</tbody>
</table>

### Teradata Date Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Return Type</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extract</td>
<td>Return part from the given date</td>
<td>Long</td>
<td>Date part (Year, Month, Day, Hour, Second) Date value</td>
</tr>
<tr>
<td>Add Months</td>
<td>Returns the date with the months added</td>
<td>Date</td>
<td>Date, months to add</td>
</tr>
</tbody>
</table>
### Teradata Miscellaneous Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Return Type</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>BYTE</td>
<td>Returns the number of bytes contained in the specified byte string.</td>
<td>Long</td>
<td>Byte, varbyte or BLOB expression.</td>
</tr>
<tr>
<td>FORMAT</td>
<td>Returns the declared format for the named expression</td>
<td>String</td>
<td>Column expression</td>
</tr>
<tr>
<td>Title</td>
<td>Returns the title of an expression as it would appear in the heading for displayed or printed results.</td>
<td>Strong</td>
<td>Column expression</td>
</tr>
<tr>
<td>Type</td>
<td>Returns the data type defined for an expression.</td>
<td>String</td>
<td>Column expression</td>
</tr>
</tbody>
</table>

### Teradata Numeric Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Return Type</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logarithmic</td>
<td>Returns base 10 logarithm of given value.</td>
<td>Real</td>
<td>Numeric</td>
</tr>
<tr>
<td>Natural Logarithm</td>
<td>Returns a natural logarithm.</td>
<td>Real</td>
<td>Numeric</td>
</tr>
<tr>
<td>Null If Zero</td>
<td>Returns “null” if the argument is zero, otherwise, returns the argument.</td>
<td>Real</td>
<td>Numeric</td>
</tr>
<tr>
<td>Random</td>
<td>Returns random integer</td>
<td>Long</td>
<td>Lower limit, upper limit</td>
</tr>
<tr>
<td>Zero If Null</td>
<td>Returns “zero” if the argument is null, otherwise, returns the argument.</td>
<td>Real</td>
<td>Numeric</td>
</tr>
</tbody>
</table>

### Teradata String Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Return Type</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concat</td>
<td>Returns concatenated string.</td>
<td>String</td>
<td>String 1, String 2</td>
</tr>
<tr>
<td>CHAR2HEXINT</td>
<td>Returns the hexadecimal representation for a character string.</td>
<td>String</td>
<td>String expression</td>
</tr>
<tr>
<td>Index</td>
<td>Returns the starting position of string 2 inside string 1. Returns 0 if string 2 is not found.</td>
<td>String</td>
<td>String, Search string</td>
</tr>
<tr>
<td>Soundex</td>
<td>Returns a character string that represents the Soundex code for a string_expression.</td>
<td>String</td>
<td>String expression</td>
</tr>
<tr>
<td>Substring</td>
<td>Returns the substring of the string provided.</td>
<td>String</td>
<td>String, start position, length</td>
</tr>
<tr>
<td>Trim</td>
<td>Trims blanks from a given string.</td>
<td>String</td>
<td>Leading/Trailing/Both String to trim</td>
</tr>
<tr>
<td>VARGRAPHIC</td>
<td>Returns the VARGRAPHIC representation of the character data in a character_string_expression.</td>
<td>String</td>
<td>String expression</td>
</tr>
</tbody>
</table>
Teradata System Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Return Type</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date</td>
<td>Returns current date.</td>
<td>Date</td>
<td></td>
</tr>
<tr>
<td>Time</td>
<td>Returns current time.</td>
<td>Time</td>
<td></td>
</tr>
<tr>
<td>User</td>
<td>Returns current user name.</td>
<td>String</td>
<td></td>
</tr>
</tbody>
</table>

Teradata Trigonometric Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Return Type</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACOS</td>
<td>Returns angle of given cosine value.</td>
<td>Real</td>
<td>Cosine value</td>
</tr>
<tr>
<td>ASIN</td>
<td>Returns the angle whose sine is the argument.</td>
<td>Real</td>
<td>Sine value</td>
</tr>
<tr>
<td>ATAN</td>
<td>Returns the angle whose tangent is the argument.</td>
<td>Real</td>
<td>Tangent value</td>
</tr>
<tr>
<td>ATAN2</td>
<td>Returns the arctangent of the given point.</td>
<td>Real</td>
<td>X and y coordinates</td>
</tr>
<tr>
<td>COSH</td>
<td>Returns the hyperbolic cosine of an argument.</td>
<td>Real</td>
<td>Angle</td>
</tr>
<tr>
<td>SINH</td>
<td>Returns the hyperbolic sine of an argument.</td>
<td>Real</td>
<td>Angle</td>
</tr>
<tr>
<td>TANH</td>
<td>Returns the hyperbolic tangent of an argument.</td>
<td>Real</td>
<td>Angle</td>
</tr>
<tr>
<td>ACOSH</td>
<td>Returns the inverse hyperbolic cosine of an argument.</td>
<td>Real</td>
<td>Angle</td>
</tr>
<tr>
<td>ASINH</td>
<td>Returns the inverse hyperbolic sine of an argument.</td>
<td>Real</td>
<td>Angle</td>
</tr>
<tr>
<td>ATANH</td>
<td>Returns the inverse hyperbolic tangent of an argument</td>
<td>Real</td>
<td>Angle</td>
</tr>
</tbody>
</table>

Teradata OLAP Functions

Interactive Reporting supports a number of Teradata version 3 OLAP and system functions, which dramatically reduce query time.

- CSum (Cumulative Sum) Function
- MAvg (Moving Average) Function
- MDiff (Moving Difference) Function
- MSum (Moving Sum) Function
- MLingreg (Multiple Linear Regression) Function
- Quantile Function
- Rank Function
- Current_Timestamp Function
- Qualify Function
Sample Function

SampleID Function

CSum (Cumulative Sum) Function

Table 100  CSum (Cumulative Sum) Function

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Accumulates a sum over an ordered set of rows, providing the current values</td>
</tr>
<tr>
<td></td>
<td>of the SUM on each row</td>
</tr>
<tr>
<td>Syntax:</td>
<td>CSum(value_expression, sort_expression_list)</td>
</tr>
<tr>
<td>value_expression</td>
<td>A value_expression is a scalar numeric column expression for which a running</td>
</tr>
<tr>
<td></td>
<td>sum is to be computed.</td>
</tr>
<tr>
<td>sort_expression_list</td>
<td>The sort_expression_list is a list of expressions (with optional sort</td>
</tr>
<tr>
<td></td>
<td>direction specifications) separated by commas. That is, it specifies the</td>
</tr>
<tr>
<td></td>
<td>column references used to sort the values. The default sort direction is</td>
</tr>
<tr>
<td></td>
<td>ascending (ASC).</td>
</tr>
</tbody>
</table>

MAvg (Moving Average) Function

Table 101  MAvg (Moving Average) Function

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Computes the moving average of a column using the current row and the</td>
</tr>
<tr>
<td></td>
<td>preceding width-1 rows.</td>
</tr>
<tr>
<td>Syntax:</td>
<td>MAvg(value_expression, width, sort_expression_list)</td>
</tr>
<tr>
<td>value_expression</td>
<td>The value expression represents a scalar numeric column expression for which</td>
</tr>
<tr>
<td></td>
<td>a moving average is to be computed. The expression cannot contain any OLAP</td>
</tr>
<tr>
<td></td>
<td>or aggregate functions.</td>
</tr>
<tr>
<td>width</td>
<td>The width represents the number of previous rows to be used in computing the</td>
</tr>
<tr>
<td></td>
<td>moving average. The width value is always a positive integer constant. The</td>
</tr>
<tr>
<td></td>
<td>maximum width is 4096.</td>
</tr>
<tr>
<td>sort_expression_list</td>
<td>The sort_expression_list is a list of expressions (with optional sort</td>
</tr>
<tr>
<td></td>
<td>direction specifications) separated by commas. That is, it specifies the</td>
</tr>
<tr>
<td></td>
<td>column references used to sort the values. The default sort direction is</td>
</tr>
<tr>
<td></td>
<td>ascending (ASC).</td>
</tr>
</tbody>
</table>

MDiff (Moving Difference) Function

Table 102  MDiff (Moving Difference) Function

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Returns the moving difference between the current row-column value and the</td>
</tr>
<tr>
<td></td>
<td>preceding nth value.</td>
</tr>
<tr>
<td></td>
<td>The moving difference is a common business metric used to compare activity</td>
</tr>
<tr>
<td></td>
<td>for some variable in a current time period to the activity for the same</td>
</tr>
<tr>
<td></td>
<td>variable in another time period at a fixed distance in the past.</td>
</tr>
<tr>
<td>Syntax:</td>
<td>MDiff(value_expression, width, sort_expression_list)</td>
</tr>
<tr>
<td>value_expression</td>
<td>The value expression represents a scalar numeric column expression for which</td>
</tr>
<tr>
<td></td>
<td>a moving average is to be computed. The expression cannot contain any OLAP</td>
</tr>
<tr>
<td></td>
<td>or aggregate functions.</td>
</tr>
</tbody>
</table>
### Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>width</td>
<td>The width represents the number of previous rows to be used in computing the moving average. The width value is always a positive integer constant. The maximum width is 4096.</td>
</tr>
<tr>
<td>sort_expression_list</td>
<td>The sort_expression_list is a list of expressions (with optional sort direction specifications) separated by commas. That is, it specifies the column references used to sort the values. The default sort direction is ascending (ASC).</td>
</tr>
</tbody>
</table>

#### MSum (Moving Sum) Function

**Table 103** MSum (Moving Sum) Function

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Computes the moving sum of a column using the current row and the preceding ( n-1 ) row.</td>
</tr>
<tr>
<td>Syntax:</td>
<td>MSum(value_expression, width, sort_expression_list)</td>
</tr>
<tr>
<td>value_expression</td>
<td>The value expression represents a scalar numeric column expression for which a moving average is to be computed. The expression cannot contain any OLAP or aggregate functions.</td>
</tr>
<tr>
<td>width</td>
<td>The width represents the number of previous rows to be used in computing the moving average. The width value is always a positive integer constant. The maximum width is 4096.</td>
</tr>
<tr>
<td>sort_expression_list</td>
<td>The sort_expression_list is a list of expressions (with optional sort direction specifications) separated by commas. That is, it specifies the column references used to sort the values. The default sort direction is ascending (ASC).</td>
</tr>
</tbody>
</table>

#### MLingreg (Multiple Linear Regression) Function

**Table 104** MLingreg (Multiple Linear Regression) Function

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Returns a predicted value for a column expression based on a least squares multiple linear regression of the previous ( width-1 ) based on the sort_expression column values. When there are fewer than ( width-1 ) preceding rows, the MLingreg function computes the regression using all the preceding rows.</td>
</tr>
<tr>
<td>Syntax:</td>
<td>MLingreg(value_expression, width, sort_expression_list)</td>
</tr>
<tr>
<td>value_expression</td>
<td>The value expression represents a scalar numeric column expression for which a moving average is to be computed. The expression cannot contain any OLAP or aggregate functions.</td>
</tr>
<tr>
<td>width</td>
<td>The width represents the number of previous rows to be used in computing the moving average. The width-1 previous rows are used to compute the linear regression and the row value itself is used to calculate the predicted value. The width value is always a positive integer constant greater than 1. The maximum width is 4096.</td>
</tr>
<tr>
<td>sort_expression_list</td>
<td>The sort_expression_list is a column reference used to sort the values and to define the dependent variable for calculating the linear regression. The sort_expression_list is an expression with optional sort direction specification. The default sort direction is ascending (ASC). Only one sort_expression is allowed with this function.</td>
</tr>
</tbody>
</table>
## Quantile Function

### Table 105  Quantile Function

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Computes the quantile scores for the values in a column. A quantile is a generic interval of user-defined width.</td>
</tr>
<tr>
<td>Syntax:</td>
<td>Quantile(quantile_constant, sort_expression_list)</td>
</tr>
<tr>
<td>quantile_constant</td>
<td>A positive integer constant used to define the number of quantile partitions to be used.</td>
</tr>
<tr>
<td>sort_expression_list</td>
<td>List of expressions (with optional sort direction specifications) separated by commas. That is, it specifies the column references used to sort the values. The default sort direction is ascending (ASC).</td>
</tr>
<tr>
<td>Quantile Value Range</td>
<td>0 through (Q-1) where Q is the number of quantile partitions specified by the quantile constant.</td>
</tr>
</tbody>
</table>

## Rank Function

### Table 106  Rank Function

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Returns the rank (1…n) of all the rows in the group by the value of sort_expression_list, with the same sort_expression values receiving the same rank.</td>
</tr>
<tr>
<td>Syntax:</td>
<td>Rank(sort_expression_list)</td>
</tr>
<tr>
<td>sort_expression_list</td>
<td>List of expressions (with optional sort direction specifications) separated by commas. That is, it specifies the column references used to sort the values. The default sort direction is ascending (ASC).</td>
</tr>
</tbody>
</table>

## Current_Timestamp Function

### Table 107  Current Timestamp Function

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Returns the current system timestamp and current session Time Zone displacement.</td>
</tr>
<tr>
<td>Syntax:</td>
<td>Current_Timestamp(fractional_precision)</td>
</tr>
<tr>
<td>fractional_precision</td>
<td>An option precision range for the returned timestamp value. The value range is 0 through 6, inclusive. The default is 6.</td>
</tr>
<tr>
<td>Properties</td>
<td>Data type: TIMESTAMP WITH TIME ZONE Length: 12 Not nullable</td>
</tr>
<tr>
<td>Fields</td>
<td>YEAR, MONTH, DAY, HOUR, MINUTE, SECOND, TIMEZONE_HOUR, TIMEZONE_MINUTE</td>
</tr>
</tbody>
</table>
## Qualify Function

### Table 108  Qualify Function

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Filters results of a previously computed OLAP function according to user-specified conditions.</td>
</tr>
<tr>
<td>Syntax:</td>
<td>Qualify search_condition</td>
</tr>
<tr>
<td>Qualify</td>
<td>Represents a conditional clause in the SELECT statement.</td>
</tr>
<tr>
<td>search_condition</td>
<td>One or more conditional expressions that must be satisfied by the result rows. Aggregate operators with a Qualify clause can be used.</td>
</tr>
</tbody>
</table>

**Usage Notes**

When you specify a QUALIFY clause in a query, you must also specify a statistical function in one of the following locations within the query.

- **select_list** of the SELECT clause
- **grouping_key** of the GROUP BY clause
- **search_condition** of the QUALIFY clause

When the WHERE, GROUP BY, and QUALIFY clauses are used together in a SELECT statement, the order of evaluation is:

WHERE GROUP BY QUALIFY

## Sample Function

### Table 109  Sample Function

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Reduces the number of rows to be considered for further processing by returning mutually exclusive samples of rows specified either as a list of fractions of the total number of rows or as a list of numbers of rows from the SELECT query.</td>
</tr>
<tr>
<td>Syntax:</td>
<td>Sample(fractional_description [count_description])</td>
</tr>
<tr>
<td>fractional_</td>
<td>Represents any number of floating point constants in the closed interval (0, 1) and separated by a comma. This is a list of fractions, the sum of which must not exceed 1.</td>
</tr>
<tr>
<td>description</td>
<td></td>
</tr>
<tr>
<td>count_</td>
<td>Represents a positive integer constant list of row counts. A warning is returned if there are not enough rows in the result to satisfy the sampling request completely.</td>
</tr>
<tr>
<td>description</td>
<td></td>
</tr>
</tbody>
</table>

**Usage Notes**

No more than 16 samples can be requested per SELECT statement.

SAMPLE operates on the evaluated output of the table expression, which can include a WHERE clause and GROUP BY, HAVING, or QUALIFY clauses, sampling the result according to use specification.

A sampling request cannot be repeated. The identical sampling query run twice against the same data will report different rows in the result.

Sampling can be used in a derived table, view, or INSERT-SELECT to reduce the number of rows to be considered for further computation.

You cannot use SAMPLE in a subquery.

If a fraction_description results in no rows being returned, a warning is generated.

If a count_description cannot be completely satisfied, a warning is generated and the sample size is reduced to the number of remaining rows.
SampleID Function

Table 110  SampleID Function

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Identifies the sample to which a row belongs, distinguishing rows belonging to different samples specified in the SAMPLE clause of a SELECT statement.</td>
</tr>
<tr>
<td>Syntax:</td>
<td>SAMPLEID</td>
</tr>
<tr>
<td>Definition</td>
<td>The sample ID identifies the sample to which a row belongs in the left-to-right order of the SAMPLE clause specification, from 1 through $n$ (where $n$ is the number of samples requested in the SAMPLE clause).</td>
</tr>
<tr>
<td>Where to Specify SAMPLEID</td>
<td>SAMPLEID can only be specified with a SAMPLE clause and can be displayed either as part of a SELECT clause or in an ORDER BY clause.</td>
</tr>
</tbody>
</table>

ANSI SQL Functions

Interactive Reporting supports ANSI SQL functions, and windowing clauses for the data. Windowing or partitions refers to a fixed window or "sub-range" over which the aggregate function is computed for each row in the partition.

The windowing clauses include:

- **Partition by**—Data to be analyzed in subsets.
- **Order by**—Sort parameter in ascending or descending order.
- **Rows**—Defines the rows over which the aggregate functions is computed for each row in the partition.

For example, if you want to compute the three-month moving average sales for each store in the sales_fact table, then partition by StoreID, order by Month, and perform the computation over the current row and the two preceding rows as shown below:

```sql
SELECT StoreID, Sales_Month, ProductID, Sales, AVG(Sales) OVER
(PARTITION BY StoreID ORDER BY Sales_Month ROWS BETWEEN 2 PRECEDING AND CURRENT ROW)
```

To apply an ANSI SQL function:

1. Drag an topic from the Elements Table Catalog to the Content pane.
2. Drag a topic to the Request line.
3. Select the topic item on the Request line and on the shortcut menu, select Add Computed Item.
   
The Modify Item dialog box is displayed.
4. Select Functions.
   
The Functions dialog box is displayed.
Note: If a function does not support windowing clauses, the Function dialog box shows only a Parameters tab. If windowing clauses are supported by the currently selected function, an Advanced tab is available for you to build windowing clauses.

5 Select a function from the Function Categories list.

A list of supported ANSI functions is shown in the Function list. Database specific ANSI functions are also shown.

6 In the Partition by field, select a topic item by which to partition data.

To review and select from a list of topic items, click Reference and select an item from the Reference dialog box.

7 In the Ordered by field, select the topic item by which to sort.

To review and select from a list of topic items, click Reference and select an item from the Reference dialog box.

8 Select to sort in Ascending or Descending order.

9 In the Rows fields, specify the number of rows over which the aggregate function is computed for each row in the partition.

To select all values, leave the default value as “Unbounded”.

10 Select the partition parameters for the Rows values. That is, select the capture area for the partition. Valid options are

- Preceding
- Current Row

11 To select values between the rows selected in the Row field, specify which rows and indicate the partition parameter. Valid options are:

- Current Row
- Following

12 Select OK.

Table 111 ANSI SQL Functions

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Return Type</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORR</td>
<td>Returns the Pearson product moment correlation coefficient of its argument fall all non-null data point pairs.</td>
<td>Real</td>
<td>numeric column expressions x and y</td>
</tr>
<tr>
<td>COVAR_POP</td>
<td>Returns the population covariance of its arguments for all non-null data point pairs.</td>
<td>Real</td>
<td>numeric column expressions x and y</td>
</tr>
<tr>
<td>COVAR_SAMP</td>
<td>Returns the sample covariance of its arguments for all non-null pairs.</td>
<td>Real</td>
<td>numeric column expression x and y</td>
</tr>
<tr>
<td>GROUPING</td>
<td>Returns a value that indicates whether a specified column in the result row was exclude</td>
<td>Integer</td>
<td>an item of a GROUP BY clause</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Return Type</td>
<td>Parameters</td>
</tr>
<tr>
<td>--------------</td>
<td>------------------------------------------------------------------------------</td>
<td>-------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>KURTOSIS</td>
<td>Returns the distribution kurtosis of the column expression.</td>
<td>Real</td>
<td>numeric column expression</td>
</tr>
<tr>
<td>REGR_AVGX</td>
<td>Returns the mean of the independent_column_expression for all non-null data pairs of the dependent and independent variable arguments.</td>
<td>Real</td>
<td>dependent and independent column expressions</td>
</tr>
<tr>
<td>REGR_AVGY</td>
<td>Returns the mean of the dependent_column_expression for all non-null data pairs of the dependent and independent variable arguments.</td>
<td>Real</td>
<td>dependent and independent column expressions</td>
</tr>
<tr>
<td>REGR_COUNT</td>
<td>Returns the count of all non-null data pairs of the dependent and independent variable arguments.</td>
<td>Real</td>
<td>dependent and independent column expressions</td>
</tr>
<tr>
<td>REGR_INTERCEPT</td>
<td>Returns the intercept of the univariate linear regression line through all non-null data pairs of the dependent and independent variable arguments.</td>
<td>Real</td>
<td>dependent and independent column expressions</td>
</tr>
<tr>
<td>REGR_R2</td>
<td>Returns the coefficient of determination for all non-null data pairs of the dependent and independent variable arguments.</td>
<td>Real</td>
<td>dependent and independent column expressions</td>
</tr>
<tr>
<td>REGR_SLOPE</td>
<td>Returns the slope of the univariate linear regression line through all non-null data pairs of the dependent and independent variable arguments.</td>
<td>Real</td>
<td>dependent and independent column expressions</td>
</tr>
<tr>
<td>REGR_SXX</td>
<td>Returns the sum of the squares of the independent_column_expression for all non-null data pairs of the dependent and independent variable arguments.</td>
<td>Real</td>
<td>dependent and independent column expressions</td>
</tr>
<tr>
<td>REGR_SXY</td>
<td>Returns the sum of the products of the independent_column_expression and the dependent_column_expression for all non-null data pairs of the dependent and independent variable arguments.</td>
<td>Real</td>
<td>dependent and independent column expressions</td>
</tr>
<tr>
<td>REGR_SYY</td>
<td>Returns the sum of the squares of the dependent_column_expression for all non-null data pairs of the dependent and independent variable arguments.</td>
<td>Real</td>
<td>dependent and independent column expressions</td>
</tr>
<tr>
<td>SKEW</td>
<td>Returns the skewness of the distribution of value_expression.</td>
<td>Real</td>
<td>numeric column expression</td>
</tr>
<tr>
<td>STDDEV_SAMP</td>
<td>Returns the sample standard deviation for the non-null data points in value_expression.</td>
<td>Real</td>
<td>numeric column expression</td>
</tr>
<tr>
<td>VAR_POP</td>
<td>Returns the population variance for the data points in value_expression.</td>
<td>Real</td>
<td>numeric column expression</td>
</tr>
<tr>
<td>VAR_SAMP</td>
<td>Returns the sample variance for the data points in value_expression.</td>
<td>Real</td>
<td>numeric column expression</td>
</tr>
<tr>
<td>WIDTH_BUCKET</td>
<td>Returns the number of the partition to which value_expression is assigned.</td>
<td>Integer</td>
<td>value expression, lower bound, upper bound, partition count</td>
</tr>
<tr>
<td>PERCENT_RANK</td>
<td>Returns the relative rank of rows for a value_expression.</td>
<td>Real</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Return Type</td>
<td>Parameters</td>
</tr>
<tr>
<td>----------------</td>
<td>------------------------------------------------------------------------------</td>
<td>-------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td><strong>ROW_NUMBER</strong></td>
<td>Returns the sequential row number, where the first row is number one, of the row within its window partition according to the window ordering of the window.</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td><strong>POSITION</strong></td>
<td>Returns the position in string_expression_2 where string_expression_1 starts.</td>
<td>Integer</td>
<td>String expression 1, string expression 2</td>
</tr>
<tr>
<td><strong>TRANSLATE</strong></td>
<td>Converts a character string or character string expression from one server character set to another server character set.</td>
<td>String</td>
<td>String expression, conversion</td>
</tr>
<tr>
<td><strong>CHARACTER_LENGTH</strong></td>
<td>Returns the length of a string either in logical characters or in bytes.</td>
<td>Integer</td>
<td>String expression</td>
</tr>
<tr>
<td><strong>OCTET_LENGTH</strong></td>
<td>Returns the length of string_expression in octets.</td>
<td>Integer</td>
<td>String expression</td>
</tr>
<tr>
<td><strong>CURRENT_DATE</strong></td>
<td>Returns the current system date.</td>
<td>Date</td>
<td></td>
</tr>
<tr>
<td><strong>CURRENT_TIME</strong></td>
<td>Returns the current system time.</td>
<td>Time</td>
<td></td>
</tr>
<tr>
<td><strong>CURRENT_TIMESTAMP</strong></td>
<td>Returns the current system timestamp.</td>
<td>Timestamp</td>
<td></td>
</tr>
</tbody>
</table>

An Oracle database loads these functions:

| Function | | | | |
|----------|---------------------------------------------------------------------------------------------------|
| Sum      | Abs, Sin, Rpad                                                                                     |
| Min      | Floor, Cos, Substr                                                                                |
| Max      | Power, Tan, To_char                                                                               |
| Avg      | Round, Decode, Last_day                                                                        |
| Count    | Sign, Nvl, Sysdate                                                                              |
| Add Months | Sqrt, Ceil, Months_between                                                                       |
| Concat   | Lower, Stddev, Next_day                                                                       |
| Length   | Upper, Variance                                                                                 |
| Trunc    | Ascii, Initcap                                                                                  |
| User     | Exp, Lpad                                                                                       |

A DB2 database loads these functions:

| Function | | | | | | |
|----------|---------------------------------------------------------------------------------------------------|
| Sum      | Lower, Coalesce, Microsecond, Decimal, Concat, Right                                            |
| Min      | Upper, Current Date, Minute, Digits, Hex, Soundex                                                |
### Functions for Returning the Day of the Week

If you need to return the day of the week on which a given date falls, some database systems enable this through functions that can be applied in the SELECT statement. Examples are the `datepart()` function in Microsoft and Sybase SQL Servers (which requests the ‘weekday’ part of the date), and Oracle’s `to_char()` function, which specifies a format of `D` (for day of week number 1 through 7) or `DAY` to get the name of the day.

If your database does not support this function, you can add a computed item to the Results section to derive it.
Formatting Day of Week Data

If you simply need to format the day of the week for displaying or printing out the date, then no special computation or statement is needed.

➤ To format the day of the week:

1. In the Results section, select the date item and choose Format, then Number.
   The Number page of the Properties dialog box is displayed.

2. Select Custom from the Category list.

3. In the Format field, type ddd to display a three-letter day abbreviation or dddd to display the full name, and click OK.

Analyzing Data Based on Day of Week Data

If you add a format such as dddd mm/dd/yyyy, it displays a date as Tuesday 05/19/2000. This may not be sufficient if you need to perform analysis on data based on the day of the week. Despite the display format, the data in the field is still a date. That is, if you need to compare sales for Mondays versus other days of the week over a given time period, changing the display format does not address your original question. In this case, you need to group all Mondays to do that sort of analysis.

➤ To analyze data based on the day of the week:

1. In the Results section, select the item and choose Results, then Add Computed Item.

2. In the Name field, assign a new name to the column.

3. In the Definition field, type:

   
   to_char(<MyDate>', 'ddd')

Replace <MyDate> with the name of the column for which you need the day of the week information. This creates a string from the date column with the desired format, as discussed earlier. You can also add the following:

   
   : decode((NEXT_DAY (<MyDate>', 'Sunday')) -<MyDate>', 7,'Sunday', 6,'Monday', 5,'Tuesday', 4,'Wednesday', 3,'Thursday', 2,'Friday', 1,'Saturday', 'Error!')

Common Computed Item Examples

The examples that follow show you how to apply some of the mathematical, numerical, and statistical calculations available in the Pivot and Chart sections using computed items.

- Math Functions
- Central Tendency
- Calculated Averages
- Percentile
- Rank
Math Functions

A mathematical equation consists of the argument and a simple or complex arithmetic operator that is applied to the argument. In this example, the sales tax is calculated as a percentage of the revenue (3.5%). The “Net Amount” column is calculated by subtracting the tax dollars from the revenue.

<table>
<thead>
<tr>
<th></th>
<th>Amount Sales</th>
<th>Tax</th>
<th>Net Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>1999</td>
<td>$114,266,340.20</td>
<td>$3,999,321.91</td>
<td>$110,267,018.295</td>
</tr>
<tr>
<td>2000</td>
<td>$18,815,338.30</td>
<td>$651,536.84</td>
<td>$17,963,801.464</td>
</tr>
</tbody>
</table>

Central Tendency

If you need to distinguish patterns within a given set of data, you can begin looking for the center of distribution where statistics tend to reside. This form of measurement involves finding the “average” in the data set and is typically referred to as the Measure of Central Tendency technique.

Three types of measurements are associated with this technique:

- **Mean**—The value is equal to the sum of the measures divided by the number of measures.
- **Median**—The value is representative of the positional middle measure.
- **Mode**—The value that occurs with the greatest frequency.

In this example, Unit Sales represent the total number of product units purchased. *Mean* of Unit Sales represents the average purchase size. *Median* of Unit Sales represent the number of product units that scores exactly in the middle of all purchase quantities. The *Mode* of Unit Sales presents the number of product units most commonly purchased at one time.

<table>
<thead>
<tr>
<th></th>
<th>Unit Sales</th>
<th>Mean Value</th>
<th>Median Value</th>
<th>Mode Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1999</td>
<td>5272112</td>
<td>411</td>
<td>276</td>
<td>370</td>
</tr>
<tr>
<td>2000</td>
<td>884533</td>
<td>621</td>
<td>404</td>
<td>441</td>
</tr>
</tbody>
</table>

Calculated Averages

Calculated averages can be applied to break columns and break values. In this example, a variety of columns are created and display different average calculations based on the Amount_Sales column.
The following table shows how averages have been calculated in the figure above.

Table 112  Calculated Averages

<table>
<thead>
<tr>
<th>Column</th>
<th>Numeric Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amount Sales</td>
<td>sum of Amount Sales by quarter and Product Line</td>
</tr>
<tr>
<td></td>
<td>= Sum (Amount_Sales)</td>
</tr>
<tr>
<td>Entire Year</td>
<td>average purchase amount (Amount Sales) across all quarters and Product Lines</td>
</tr>
<tr>
<td></td>
<td>= Avg (Amount_Sales)</td>
</tr>
<tr>
<td>By Quarter</td>
<td>average purchase amount (Amount Sales) in a specific quarter</td>
</tr>
<tr>
<td></td>
<td>= Avg (Amount_Sales, Quarter)</td>
</tr>
<tr>
<td>For Quarter and Product Line</td>
<td>average purchase amount (Amount Sales) in a specific quarter for a Product line</td>
</tr>
<tr>
<td></td>
<td>= Avg data function applied to Amount_Sales column</td>
</tr>
<tr>
<td>For Q1</td>
<td>average purchase size (Amount Sales) across all Product Lines for Q1 specifically</td>
</tr>
<tr>
<td></td>
<td>= Avg (Amount_Sales, Quarter, 'Q1')</td>
</tr>
<tr>
<td>For Books</td>
<td>average purchase size (Amount Sales) across all Quarters for Books</td>
</tr>
<tr>
<td></td>
<td>= Avg (Amount_Sales, Product_Line, 'Books')</td>
</tr>
</tbody>
</table>

**Percentile**

Suppose Sales Managers qualify for a special bonus if they are within the 80th percentile (Qualify column). You can define an 80th percentile value for Amount Sales.
The second example identifies countries that make sales transactions under $10,000.

<table>
<thead>
<tr>
<th>Rank</th>
<th>Amount Sales</th>
<th>Eightieth Percentile</th>
<th>Qualify</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$351,481</td>
<td>7,344</td>
<td>$7,344</td>
</tr>
<tr>
<td>2</td>
<td>$153,290</td>
<td>7,290</td>
<td>$7,290</td>
</tr>
<tr>
<td>3</td>
<td>$234,890</td>
<td>7,234</td>
<td>$7,234</td>
</tr>
<tr>
<td>4</td>
<td>$98,790</td>
<td>7,987</td>
<td>$7,987</td>
</tr>
<tr>
<td>5</td>
<td>$453,900</td>
<td>7,453</td>
<td>$7,453</td>
</tr>
<tr>
<td>6</td>
<td>$123,900</td>
<td>7,123</td>
<td>$7,123</td>
</tr>
<tr>
<td>7</td>
<td>$324,900</td>
<td>7,324</td>
<td>$7,324</td>
</tr>
<tr>
<td>8</td>
<td>$198,790</td>
<td>7,198</td>
<td>$7,198</td>
</tr>
<tr>
<td>9</td>
<td>$98,790</td>
<td>7,987</td>
<td>$7,987</td>
</tr>
<tr>
<td>10</td>
<td>$453,900</td>
<td>7,453</td>
<td>$7,453</td>
</tr>
</tbody>
</table>

**Rank**

You can return the rank of a number in a column of numbers. The *Rank* function works as if you were to sort the list in descending order. In this example, Amount Sales values are ranked for each Country.

**Note:** The *Rank* function assigns duplicate numbers the same rank, which affects the ranks of subsequent numbers.
Project Sales

This example shows a calculation for a 20% increase in sales projections for each quarter, based on Amount Sales for 1999.

Scalar Function Examples

Following are examples of some common scalar functions. Each example shows the syntax used and the result of applying the function. The examples that follow do not include all the scalar functions. The functions described include:
Avg

The `Avg` function returns the average (arithmetic mean) of values in a number column.

\[ \text{Avg} \left( \text{numbers}, \text{break}_\text{col}, \text{break}_\text{value} \right) \]

where:

- `numbers` references the column that contains the numbers on which the average is calculated.
- `break_col` is an optional parameter that references a break column.
- `break_value` is an optional parameter that returns the average of numbers column where value in `break_col` equals `break_value`.

**Note:** If constant values in the `break_value` column are substituted for data items, dates and text strings must be enclosed in single quotes.

The following examples show the results of inserting the `Avg` function in three separate tables.
Example 1

In this example, the `Avg` function is used on the numeric column. The results are shown in the Computed column.

\[ \text{Avg (Amount)} \]

<table>
<thead>
<tr>
<th>State</th>
<th>City</th>
<th>Amount</th>
<th>Computed</th>
</tr>
</thead>
<tbody>
<tr>
<td>AZ</td>
<td>Tucson</td>
<td>112</td>
<td>183.67</td>
</tr>
<tr>
<td>CA</td>
<td>Burbank</td>
<td>240</td>
<td>183.67</td>
</tr>
<tr>
<td>CA</td>
<td>Glendale</td>
<td>97</td>
<td>183.67</td>
</tr>
<tr>
<td>FL</td>
<td>Palmetto</td>
<td>70</td>
<td>183.67</td>
</tr>
<tr>
<td>MD</td>
<td>Laurel</td>
<td>97</td>
<td>183.67</td>
</tr>
<tr>
<td>MI</td>
<td>Detroit</td>
<td>208</td>
<td>183.67</td>
</tr>
<tr>
<td>MN</td>
<td>Eagan</td>
<td>159</td>
<td>183.67</td>
</tr>
<tr>
<td>NY</td>
<td>New York</td>
<td>490</td>
<td>183.67</td>
</tr>
<tr>
<td>NY</td>
<td>Rochester</td>
<td>180</td>
<td>183.67</td>
</tr>
</tbody>
</table>

Example 2

In this example, the `Avg` function is used on the numeric column and break_column. The results are shown in the Computed column.

\[ \text{Avg (Amount, State)} \]

<table>
<thead>
<tr>
<th>State</th>
<th>City</th>
<th>Amount</th>
<th>Computed</th>
</tr>
</thead>
<tbody>
<tr>
<td>AZ</td>
<td>Tucson</td>
<td>112</td>
<td>112</td>
</tr>
<tr>
<td>CA</td>
<td>Burbank</td>
<td>240</td>
<td>168.5</td>
</tr>
<tr>
<td>CA</td>
<td>Glendale</td>
<td>97</td>
<td>168.5</td>
</tr>
<tr>
<td>FL</td>
<td>Palmetto</td>
<td>70</td>
<td>70</td>
</tr>
<tr>
<td>MD</td>
<td>Laurel</td>
<td>97</td>
<td>97</td>
</tr>
<tr>
<td>MI</td>
<td>Detroit</td>
<td>208</td>
<td>208</td>
</tr>
<tr>
<td>MN</td>
<td>Eagan</td>
<td>159</td>
<td>159</td>
</tr>
<tr>
<td>NY</td>
<td>New York</td>
<td>490</td>
<td>335</td>
</tr>
<tr>
<td>NY</td>
<td>Rochester</td>
<td>180</td>
<td>335</td>
</tr>
</tbody>
</table>
**AvgNonNull**

The `AvgNonNull` function returns the average (arithmetic mean) of values in a number column, excluding null values.

\[
\text{Avg (numbers, break\_col, break\_value)}
\]

where:

* `numbers` references the column that contains the numbers on which the average is calculated.
* `break\_col` is an optional parameter that references a break column.
* `break\_value` is an optional parameter that returns the average of non-null numbers column where value in `break\_col` equals `break\_value`.

**Note:** If constant values in the `break\_value` column are substituted for data items, dates and text strings must be enclosed in single quotes.

The following examples show the results of inserting the `AvgNonNull` function in three separate tables.

**Example 1**

In this example, the `AvgNonNull` function is used on the numeric column. The results are shown in the Computed column

\[
\text{AvgNonNull (Amount)}
\]

<table>
<thead>
<tr>
<th>State</th>
<th>City</th>
<th>Amount</th>
<th>Computed</th>
</tr>
</thead>
<tbody>
<tr>
<td>AZ</td>
<td>Tucson</td>
<td>112</td>
<td>194.5</td>
</tr>
<tr>
<td>CA</td>
<td>Burbank</td>
<td>240</td>
<td>194.5</td>
</tr>
<tr>
<td>CA</td>
<td>Glendale</td>
<td>NULL</td>
<td>194.5</td>
</tr>
<tr>
<td>FL</td>
<td>Palmetto</td>
<td>70</td>
<td>194.5</td>
</tr>
<tr>
<td>MD</td>
<td>Laurel</td>
<td>97</td>
<td>194.5</td>
</tr>
<tr>
<td>MI</td>
<td>Detroit</td>
<td>208</td>
<td>194.5</td>
</tr>
<tr>
<td>MN</td>
<td>Eagan</td>
<td>159</td>
<td>194.5</td>
</tr>
<tr>
<td>NY</td>
<td>New York</td>
<td>490</td>
<td>194.5</td>
</tr>
<tr>
<td>NY</td>
<td>Rochester</td>
<td>180</td>
<td>194.5</td>
</tr>
</tbody>
</table>

**Example 2**

In this example, the `AvgNonNull` function is used on the numeric column and `break\_column`. The results are shown in the Computed column

\[
\text{AvgNonNull (Amount, State)}
\]
### Table 115  AvgNonNull Example 2

<table>
<thead>
<tr>
<th>State</th>
<th>City</th>
<th>Amount</th>
<th>Computed</th>
</tr>
</thead>
<tbody>
<tr>
<td>AZ</td>
<td>Tucson</td>
<td>112</td>
<td>112</td>
</tr>
<tr>
<td>CA</td>
<td>Burbank</td>
<td>240</td>
<td>240</td>
</tr>
<tr>
<td>CA</td>
<td>Glendale</td>
<td>NULL</td>
<td>240</td>
</tr>
<tr>
<td>FL</td>
<td>Palmetto</td>
<td>70</td>
<td>70</td>
</tr>
<tr>
<td>MD</td>
<td>Laurel</td>
<td>97</td>
<td>97</td>
</tr>
<tr>
<td>MI</td>
<td>Detroit</td>
<td>208</td>
<td>208</td>
</tr>
<tr>
<td>MN</td>
<td>Eagan</td>
<td>159</td>
<td>159</td>
</tr>
<tr>
<td>NY</td>
<td>New York</td>
<td>490</td>
<td>335</td>
</tr>
<tr>
<td>NY</td>
<td>Rochester</td>
<td>180</td>
<td>335</td>
</tr>
</tbody>
</table>

### Example 3

In this example, the `AvgNonNull` function is used on the numeric column, `break_column`, and `break_value`. The results are shown in the `Computed` column.

```
AvgNonNull (Amount, State, ‘CA’)
```

### Table 116  AvgNonNull Example 3

<table>
<thead>
<tr>
<th>State</th>
<th>City</th>
<th>Amount</th>
<th>Computed</th>
</tr>
</thead>
<tbody>
<tr>
<td>AZ</td>
<td>Tucson</td>
<td>112</td>
<td>240</td>
</tr>
<tr>
<td>CA</td>
<td>Burbank</td>
<td>240</td>
<td>240</td>
</tr>
<tr>
<td>CA</td>
<td>Glendale</td>
<td>NULL</td>
<td>240</td>
</tr>
<tr>
<td>FL</td>
<td>Palmetto</td>
<td>70</td>
<td>240</td>
</tr>
<tr>
<td>MD</td>
<td>Laurel</td>
<td>97</td>
<td>240</td>
</tr>
<tr>
<td>MI</td>
<td>Detroit</td>
<td>208</td>
<td>240</td>
</tr>
<tr>
<td>MN</td>
<td>Eagan</td>
<td>159</td>
<td>240</td>
</tr>
<tr>
<td>NY</td>
<td>New York</td>
<td>490</td>
<td>240</td>
</tr>
<tr>
<td>NY</td>
<td>Rochester</td>
<td>180</td>
<td>240</td>
</tr>
</tbody>
</table>

### ColMax

The `ColMax` function returns the largest value in a column of numbers.

```
ColMax (numbers, break_col, break_value)
```
where:

- `numbers` references the column that contains the number on which the maximum column value is calculated.
- `break_col` is an optional parameter that references a break column.
- `break_value` is an optional parameter that returns the maximum value of numbers column where value in `break_col` equals `break_value`.

**Note:** If constant values in the `break_value` column are substituted for data items, dates and text strings must be enclosed in single quotes.

The following examples show the results of inserting the `ColMax` function in three separate tables.

**Example 1**

In this example, the `ColMax` function is used on the numeric column. The results are shown in the Computed column.

`ColMax(Amount)`

<table>
<thead>
<tr>
<th>State</th>
<th>City</th>
<th>Amount</th>
<th>Computed</th>
</tr>
</thead>
<tbody>
<tr>
<td>AZ</td>
<td>Tucson</td>
<td>112</td>
<td>490</td>
</tr>
<tr>
<td>CA</td>
<td>Burbank</td>
<td>240</td>
<td>490</td>
</tr>
<tr>
<td>CA</td>
<td>Glendale</td>
<td>NULL</td>
<td>490</td>
</tr>
<tr>
<td>FL</td>
<td>Palmetto</td>
<td>70</td>
<td>490</td>
</tr>
<tr>
<td>MD</td>
<td>Laurel</td>
<td>97</td>
<td>490</td>
</tr>
<tr>
<td>MI</td>
<td>Detroit</td>
<td>208</td>
<td>490</td>
</tr>
<tr>
<td>MN</td>
<td>Eagan</td>
<td>159</td>
<td>490</td>
</tr>
<tr>
<td>NY</td>
<td>New York</td>
<td>490</td>
<td>490</td>
</tr>
<tr>
<td>NY</td>
<td>Rochester</td>
<td>180</td>
<td>490</td>
</tr>
</tbody>
</table>

**ColMin**

The `ColMin` function returns the smallest value in a column of numbers.

`ColMin (numbers, break_col, break_value)`

where:
*numbers* references the column that contains the numbers on which the count of minimum column value is calculated.

*break_col* is an optional parameter that references a break column.

*break_value* is an optional parameter that returns the minimum value of numbers column where value in *break_col* equals *break_value*.

**Note:** If constant values in the *break_value* column are substituted for data items, dates and text strings must be enclosed in single quotes.

The following examples show the results of inserting the *ColMin* function in three separate tables.

**Example 1**

In this example, the *ColMin* function is used on the numeric column. The results are shown in the Computed column.

*ColMin* (*Amount*)

<table>
<thead>
<tr>
<th>State</th>
<th>City</th>
<th>Amount</th>
<th>Computed</th>
</tr>
</thead>
<tbody>
<tr>
<td>AZ</td>
<td>Tucson</td>
<td>112</td>
<td>70</td>
</tr>
<tr>
<td>CA</td>
<td>Burbank</td>
<td>240</td>
<td>70</td>
</tr>
<tr>
<td>CA</td>
<td>Glendale</td>
<td>NULL</td>
<td>70</td>
</tr>
<tr>
<td>FL</td>
<td>Palmetto</td>
<td>70</td>
<td>70</td>
</tr>
<tr>
<td>MD</td>
<td>Laurel</td>
<td>97</td>
<td>70</td>
</tr>
<tr>
<td>MI</td>
<td>Detroit</td>
<td>208</td>
<td>70</td>
</tr>
<tr>
<td>MN</td>
<td>Eagan</td>
<td>159</td>
<td>70</td>
</tr>
<tr>
<td>NY</td>
<td>New York</td>
<td>490</td>
<td>70</td>
</tr>
<tr>
<td>NY</td>
<td>Rochester</td>
<td>180</td>
<td>70</td>
</tr>
</tbody>
</table>

**Example 2**

In this example, the *ColMin* function is used on the numeric column and *break_column*. The results are shown in the Computed column.

*ColMin* (*Amount*, *State*)

<table>
<thead>
<tr>
<th>State</th>
<th>City</th>
<th>Amount</th>
<th>Computed</th>
</tr>
</thead>
<tbody>
<tr>
<td>AZ</td>
<td>Tucson</td>
<td>112</td>
<td>112</td>
</tr>
<tr>
<td>State</td>
<td>City</td>
<td>Amount</td>
<td>Computed</td>
</tr>
<tr>
<td>-------</td>
<td>----------</td>
<td>--------</td>
<td>----------</td>
</tr>
<tr>
<td>CA</td>
<td>Burbank</td>
<td>240</td>
<td>240</td>
</tr>
<tr>
<td>CA</td>
<td>Glendale</td>
<td>Null</td>
<td>240</td>
</tr>
<tr>
<td>FL</td>
<td>Palmetto</td>
<td>70</td>
<td>70</td>
</tr>
<tr>
<td>MD</td>
<td>Laurel</td>
<td>97</td>
<td>97</td>
</tr>
<tr>
<td>MI</td>
<td>Detroit</td>
<td>208</td>
<td>208</td>
</tr>
<tr>
<td>MN</td>
<td>Eagan</td>
<td>159</td>
<td>159</td>
</tr>
<tr>
<td>NY</td>
<td>New York</td>
<td>490</td>
<td>180</td>
</tr>
<tr>
<td>NY</td>
<td>Rochester</td>
<td>180</td>
<td>180</td>
</tr>
</tbody>
</table>

**Count**

The Count function counts the number of rows in a column.

Count \((\text{numbers, break\_col, break\_value})\)

where:

- **numbers** references the column that contains the numbers on which the count is calculated.
- **break\_col** is an optional parameter that references a break column.
- **break\_value** is an optional parameter that returns the count of numbers column where value in break\_col equals break\_value.

**Note:** If constant values in the break\_value column are substituted for data items, dates and text strings must be enclosed in single quotes.

The following examples show the results of inserting the Count function in three separate tables.

**Example 1**

In this example, the Count function is used on the numeric column. The results are shown in the Computed column.

\[
\text{Count (Amount)}
\]

**Table 120**  
Count Example

<table>
<thead>
<tr>
<th>State</th>
<th>City</th>
<th>Amount</th>
<th>Computed</th>
</tr>
</thead>
<tbody>
<tr>
<td>AZ</td>
<td>Tucson</td>
<td>112</td>
<td>9</td>
</tr>
<tr>
<td>CA</td>
<td>Burbank</td>
<td>240</td>
<td>9</td>
</tr>
<tr>
<td>CA</td>
<td>Glendale</td>
<td>97</td>
<td>9</td>
</tr>
</tbody>
</table>
### CountDistinct

The `CountDistinct` function counts the number of values in a column.

`CountDistinct(numbers, break_col, break_values)`

where:

- `numbers` references the column that contains the numbers on which the count of distinct (unique) values is calculated.
- `break_col` is an optional parameter that references a break column.
- `break_value` is an optional parameter that returns a distinct (unique) count of numbers column where value in `break_col` equals `break_value`.

**Note:** The `CountDistinct` function differentiates actual values and not the rows. The Count function counts only the actual rows in a column. For example, if a column named “OS Operating Systems” has one hundred rows and shows data by Windows and UNIX systems, the `CountDistinct` function counts only the three operating systems and not the number of rows.

The following examples show the results of inserting the `CountDistinct` function in three separate tables.

**Example 1**

In this example, the `CountDistinct` function is used on the numeric column. The results are shown in the Computed column.

`CountDistinct(Amount)`

<table>
<thead>
<tr>
<th>State</th>
<th>City</th>
<th>Amount</th>
<th>Computed</th>
</tr>
</thead>
<tbody>
<tr>
<td>AZ</td>
<td>Tucson</td>
<td>112</td>
<td>8</td>
</tr>
<tr>
<td>CA</td>
<td>Burbank</td>
<td>240</td>
<td>8</td>
</tr>
</tbody>
</table>

---

**Table 121**  CountDistinct Example 1

---

*368  Computed Items*
<table>
<thead>
<tr>
<th>State</th>
<th>City</th>
<th>Amount</th>
<th>Computed</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA</td>
<td>Glendale</td>
<td>240</td>
<td>8</td>
</tr>
<tr>
<td>FL</td>
<td>Palmetto</td>
<td>70</td>
<td>8</td>
</tr>
<tr>
<td>MD</td>
<td>Laurel</td>
<td>97</td>
<td>8</td>
</tr>
<tr>
<td>MI</td>
<td>Detroit</td>
<td>208</td>
<td>8</td>
</tr>
<tr>
<td>MN</td>
<td>Eagan</td>
<td>159</td>
<td>8</td>
</tr>
<tr>
<td>NY</td>
<td>New York</td>
<td>490</td>
<td>8</td>
</tr>
<tr>
<td>NY</td>
<td>Rochester</td>
<td>180</td>
<td>8</td>
</tr>
</tbody>
</table>

**Example 2**

In this example, the `CountDistinct` function is used on the numeric column and break_column. The results are shown in the Computed column.

`CountDistinct (Amount, State)`

<table>
<thead>
<tr>
<th>State</th>
<th>City</th>
<th>Amount</th>
<th>Computed</th>
</tr>
</thead>
<tbody>
<tr>
<td>AZ</td>
<td>Tucson</td>
<td>112</td>
<td>1</td>
</tr>
<tr>
<td>CA</td>
<td>Burbank</td>
<td>240</td>
<td>1</td>
</tr>
<tr>
<td>CA</td>
<td>Glendale</td>
<td>240</td>
<td>1</td>
</tr>
<tr>
<td>FL</td>
<td>Palmetto</td>
<td>70</td>
<td>1</td>
</tr>
<tr>
<td>MD</td>
<td>Laurel</td>
<td>97</td>
<td>1</td>
</tr>
<tr>
<td>MI</td>
<td>Detroit</td>
<td>208</td>
<td>1</td>
</tr>
<tr>
<td>MN</td>
<td>Eagan</td>
<td>159</td>
<td>1</td>
</tr>
<tr>
<td>NY</td>
<td>New York</td>
<td>490</td>
<td>1</td>
</tr>
<tr>
<td>NY</td>
<td>Rochester</td>
<td>180</td>
<td>1</td>
</tr>
</tbody>
</table>

**CountNull**

The `CountNull` function counts the number of rows in a column that contains null values.

`CountNull (numbers, break_col, break_value)`

where:

`numbers` references the column that contains the numbers on which the count of null values is calculated.

`break_col` is an optional parameter that references a break column.
break_value is an optional parameter that returns the count of null numbers column where value in break_col equals break_value.

**Note:** If constant values in the break_value column are substituted for data items, dates and text strings must be enclosed in single quotes.

The following examples show the results of inserting the CountNull function in three separate tables.

**Example**

In this example, the CountNull function is used on the numeric column. The results are shown in the Computed column.

CountNull(Amount)

<table>
<thead>
<tr>
<th>State</th>
<th>City</th>
<th>Amount</th>
<th>Computed</th>
</tr>
</thead>
<tbody>
<tr>
<td>AZ</td>
<td>Tucson</td>
<td>112</td>
<td>1</td>
</tr>
<tr>
<td>CA</td>
<td>Burbank</td>
<td>240</td>
<td>1</td>
</tr>
<tr>
<td>CA</td>
<td>Glendale</td>
<td>NULL</td>
<td>1</td>
</tr>
<tr>
<td>FL</td>
<td>Palmetto</td>
<td>70</td>
<td>1</td>
</tr>
<tr>
<td>MD</td>
<td>Laurel</td>
<td>97</td>
<td>1</td>
</tr>
<tr>
<td>MI</td>
<td>Detroit</td>
<td>208</td>
<td>1</td>
</tr>
<tr>
<td>MN</td>
<td>Eagan</td>
<td>159</td>
<td>1</td>
</tr>
<tr>
<td>NY</td>
<td>New York</td>
<td>490</td>
<td>1</td>
</tr>
<tr>
<td>NY</td>
<td>Rochester</td>
<td>180</td>
<td>1</td>
</tr>
</tbody>
</table>

**CountNonNull**

The CountNonNull function counts the number of rows in a column that do not contain null values.

CountNonNull (numbers, break_col, break_value)

where:

- **numbers** references the column that contains the numbers on which the count of non-null values is calculated.
- **break_col** is an optional parameter that references a break column.
- **break_value** is an optional parameter that returns the count of non-null numbers column where value in break_col equals break_value.
Note: If constant values in the break_value column are substituted for data items, dates and text strings must be enclosed in single quotes.

The following examples show the results of inserting the CountNonNull function in three separate tables.

Example 1
In this example, the CountNonNull function is used on the numeric column. The results are shown in the Computed column.

\[ \text{CountNonNull (Amount)} \]

<table>
<thead>
<tr>
<th>State</th>
<th>City</th>
<th>Amount</th>
<th>Computed</th>
</tr>
</thead>
<tbody>
<tr>
<td>AZ</td>
<td>Tucson</td>
<td>112</td>
<td>8</td>
</tr>
<tr>
<td>CA</td>
<td>Burbank</td>
<td>240</td>
<td>8</td>
</tr>
<tr>
<td>CA</td>
<td>Glendale</td>
<td>NULL</td>
<td>8</td>
</tr>
<tr>
<td>FL</td>
<td>Palmetto</td>
<td>70</td>
<td>8</td>
</tr>
<tr>
<td>MD</td>
<td>Laurel</td>
<td>97</td>
<td>8</td>
</tr>
<tr>
<td>MI</td>
<td>Detroit</td>
<td>208</td>
<td>8</td>
</tr>
<tr>
<td>MN</td>
<td>Eagan</td>
<td>159</td>
<td>8</td>
</tr>
<tr>
<td>NY</td>
<td>New York</td>
<td>490</td>
<td>8</td>
</tr>
<tr>
<td>NY</td>
<td>Rochester</td>
<td>180</td>
<td>8</td>
</tr>
</tbody>
</table>

Example 2
In this example, the CountNonNull function is used on the numeric column and break_column. The results are shown in the Computed column.

\[ \text{CountNonNull (Amount, State)} \]

<table>
<thead>
<tr>
<th>State</th>
<th>City</th>
<th>Amount</th>
<th>Computed</th>
</tr>
</thead>
<tbody>
<tr>
<td>AZ</td>
<td>Tucson</td>
<td>112</td>
<td>1</td>
</tr>
<tr>
<td>CA</td>
<td>Burbank</td>
<td>240</td>
<td>1</td>
</tr>
<tr>
<td>CA</td>
<td>Glendale</td>
<td>NULL</td>
<td>1</td>
</tr>
<tr>
<td>FL</td>
<td>Palmetto</td>
<td>70</td>
<td>1</td>
</tr>
<tr>
<td>MD</td>
<td>Laurel</td>
<td>97</td>
<td>1</td>
</tr>
</tbody>
</table>
Example 3

In this example, the `CountNonNull` function is used on the numeric column, `break_column`, and `break_value`. The results are shown in the `Computed` column.

```
CountNonNull (Amount, State, 'CA')
```

<table>
<thead>
<tr>
<th>State</th>
<th>City</th>
<th>Amount</th>
<th>Computed</th>
</tr>
</thead>
<tbody>
<tr>
<td>MI</td>
<td>Detroit</td>
<td>208</td>
<td>1</td>
</tr>
<tr>
<td>MN</td>
<td>Eagan</td>
<td>159</td>
<td>1</td>
</tr>
<tr>
<td>NY</td>
<td>New York</td>
<td>490</td>
<td>2</td>
</tr>
<tr>
<td>NY</td>
<td>Rochester</td>
<td>180</td>
<td>2</td>
</tr>
</tbody>
</table>

**Cume**

The `Cume` function returns a cumulative running total for each value in a column of numbers.

```
Cume (numbers, break_col)
```

where:

- `numbers` references the column that contains the numbers on which the cume is calculated.
- `break_col` is an optional parameter that references a break column.

**Note:** If constant values in the `break_value` column are substituted for data items, dates and text strings must be enclosed in single quotes.

The following examples show the results of inserting the `Cume` function in two separate tables.
Example 1
In this example, the Cume function is used on the numeric column. The results are shown in the Computed column.

\texttt{Cume (Amount)}

Table 127  Cume Example 1

<table>
<thead>
<tr>
<th>State</th>
<th>City</th>
<th>Amount</th>
<th>Computed</th>
</tr>
</thead>
<tbody>
<tr>
<td>AZ</td>
<td>Tucson</td>
<td>112</td>
<td>112</td>
</tr>
<tr>
<td>CA</td>
<td>Burbank</td>
<td>240</td>
<td>352</td>
</tr>
<tr>
<td>CA</td>
<td>Glendale</td>
<td>97</td>
<td>449</td>
</tr>
<tr>
<td>FL</td>
<td>Palmetto</td>
<td>70</td>
<td>519</td>
</tr>
<tr>
<td>MD</td>
<td>Laurel</td>
<td>97</td>
<td>646</td>
</tr>
<tr>
<td>MI</td>
<td>Detroit</td>
<td>208</td>
<td>824</td>
</tr>
<tr>
<td>MN</td>
<td>Eagan</td>
<td>159</td>
<td>983</td>
</tr>
<tr>
<td>NY</td>
<td>New York</td>
<td>490</td>
<td>1473</td>
</tr>
<tr>
<td>NY</td>
<td>Rochester</td>
<td>180</td>
<td>1653</td>
</tr>
</tbody>
</table>

Example 2
In this example, the Cume function is used on the numeric column and break_column. The results are shown in the Computed column.

\texttt{Cume (Amount, State)}

Table 128  Cume Example 2

<table>
<thead>
<tr>
<th>State</th>
<th>City</th>
<th>Amount</th>
<th>Computed</th>
</tr>
</thead>
<tbody>
<tr>
<td>AZ</td>
<td>Tucson</td>
<td>112</td>
<td>112</td>
</tr>
<tr>
<td>CA</td>
<td>Burbank</td>
<td>240</td>
<td>240</td>
</tr>
<tr>
<td>CA</td>
<td>Glendale</td>
<td>97</td>
<td>337</td>
</tr>
<tr>
<td>FL</td>
<td>Palmetto</td>
<td>70</td>
<td>70</td>
</tr>
<tr>
<td>MD</td>
<td>Laurel</td>
<td>97</td>
<td>97</td>
</tr>
<tr>
<td>MI</td>
<td>Detroit</td>
<td>208</td>
<td>208</td>
</tr>
<tr>
<td>MN</td>
<td>Eagan</td>
<td>159</td>
<td>159</td>
</tr>
<tr>
<td>NY</td>
<td>New York</td>
<td>490</td>
<td>490</td>
</tr>
<tr>
<td>NY</td>
<td>Rochester</td>
<td>180</td>
<td>670</td>
</tr>
</tbody>
</table>
Median

The Median function returns the median of a column of numbers. The median is the middle value or number in the middle of a set of numbers (and not the average).

\[
\text{Median} \ (\text{numbers, break\_col})
\]

*numbers* references the column that contains the numbers on which the median is calculated. *break\_col* is an optional parameter that references a break column.

The following examples show the results of inserting the \text{Median} function in two separate tables.

Example 1

In this example, the \text{Median} function is used on a numeric column that has an odd number of rows:

\[
\text{Median} \ (\text{Amount})
\]

The \text{Median} function returns the number in the middle, which in this example is 30.

<table>
<thead>
<tr>
<th>State</th>
<th>Amount</th>
<th>Computed</th>
</tr>
</thead>
<tbody>
<tr>
<td>AZ</td>
<td>10</td>
<td>30</td>
</tr>
<tr>
<td>CA</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>FL</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>MD</td>
<td>40</td>
<td>30</td>
</tr>
<tr>
<td>MI</td>
<td>50</td>
<td>30</td>
</tr>
</tbody>
</table>

If the numbers column has an even number of rows, the \text{Median} function calculates the average of the two numbers in the middle.

<table>
<thead>
<tr>
<th>State</th>
<th>Amount</th>
<th>Computed</th>
</tr>
</thead>
<tbody>
<tr>
<td>AZ</td>
<td>10</td>
<td>25</td>
</tr>
<tr>
<td>CA</td>
<td>20</td>
<td>25</td>
</tr>
<tr>
<td>FL</td>
<td>30</td>
<td>25</td>
</tr>
<tr>
<td>MD</td>
<td>40</td>
<td>25</td>
</tr>
</tbody>
</table>

Mode

The Mode function returns the most frequently occurring value in columns of numbers.

\[
\text{Mode} \ (\text{numbers, break\_col})
\]
where:

- `numbers` references the column that contains the numbers on which the mode is calculated.
- `break_col` is an optional parameter that references a break column.

**Note:** Null values in the numbers column are ignored. Zeroes (0) are included. If the numbers column has no duplicate data values, the `Mode` function returns the value of the first cell in the numbers column.

The following examples show the results of inserting the `Mode` function in two separate tables.

**Example**

In this example, the `Mode` function is used on the numeric column. The results are shown in the `Computed` column.

\[
\text{Mode(Amount)}
\]

<table>
<thead>
<tr>
<th>State</th>
<th>Amount</th>
<th>Computed</th>
</tr>
</thead>
<tbody>
<tr>
<td>AZ</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>CA</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>FL</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>MD</td>
<td>40</td>
<td>10</td>
</tr>
<tr>
<td>MI</td>
<td>50</td>
<td>10</td>
</tr>
</tbody>
</table>

**Percentile**

The `Percentile` function returns the \( n \)th percentile of values in a column of numbers.

\[
\text{Percentile(numbers, n, break\_col)}
\]

where:

- `numbers` references the column that contains the numbers on which the percentile is calculated.
- \( n \) is the percentile value 0 to 1 inclusive.
- `break_col` is an optional parameter that references a break column.

**Note:** `Percentile` can also be used to return quartile values by setting the \( n \)th percentile to the following: 0.25 for first quartile, 0.5 for second quartile, 0.75 for third quartile.

The following examples show the results of inserting the `Percentile` function in three separate tables.
Example 1

In this example, two computed value columns have been calculated. In the first computed column, the \texttt{Percentile} function is used on the numeric column and the $n^{th}$ percentile of values (.80 in this case).

\texttt{Percentile (Units, .80)}

In the second computed column, the \texttt{Percentile} function is used on the numeric column, the $n^{th}$ percentile of values (.80 in this case), and the \texttt{break\_column}.

\texttt{Percentile (Units, .80, State)}

<table>
<thead>
<tr>
<th>State</th>
<th>Units</th>
<th>Computed 1</th>
<th>Computed 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>WI</td>
<td>50</td>
<td>191.8</td>
<td>50</td>
</tr>
<tr>
<td>AZ</td>
<td>70</td>
<td>191.8</td>
<td>70</td>
</tr>
<tr>
<td>CA</td>
<td>96</td>
<td>191.8</td>
<td>99.8</td>
</tr>
<tr>
<td>CA</td>
<td>98</td>
<td>191.8</td>
<td>99.8</td>
</tr>
<tr>
<td>CA</td>
<td>101</td>
<td>191.8</td>
<td>99.8</td>
</tr>
<tr>
<td>FL</td>
<td>112</td>
<td>191.8</td>
<td>112</td>
</tr>
<tr>
<td>MD</td>
<td>159</td>
<td>191.8</td>
<td>159</td>
</tr>
<tr>
<td>NY</td>
<td>241</td>
<td>191.8</td>
<td>440.2</td>
</tr>
<tr>
<td>NY</td>
<td>490</td>
<td>191.8</td>
<td>440.2</td>
</tr>
</tbody>
</table>

Example 2

In this example, two computed value columns have been calculated. In the first computed column, the \texttt{Percentile} function is used on the numeric column and the $n^{th}$ percentile of values (0 in this case).

\texttt{Percentile (Units, 0)}

In the second computed column, the \texttt{Percentile} function is used on the numeric column, and the $n^{th}$ percentile of values (.25 in this case).

\texttt{Percentile (Units, .25)}

<table>
<thead>
<tr>
<th>State</th>
<th>Units</th>
<th>Computed 1</th>
<th>Computed 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>WI</td>
<td>50</td>
<td>50</td>
<td>96</td>
</tr>
<tr>
<td>AZ</td>
<td>70</td>
<td>50</td>
<td>96</td>
</tr>
<tr>
<td>CA</td>
<td>96</td>
<td>50</td>
<td>96</td>
</tr>
</tbody>
</table>
State Units Computed 1 Computed 2
CA 98 50 96
CA 101 50 96
FL 112 50 96
MD 159 50 96
NY 241 50 96
NY 490 50 96

**Rank**

The Rank function returns the rank of a number in a column of numbers. It works as if you were to sort the list in descending order.

Rank(numbers, break_col)

where:

- **numbers** references the column that contains the numbers on which the rank is calculated.
- **break_col** is an optional parameter that references a break column.

**Note:** The Rank function assigns duplicate numbers the same rank. The presence of duplicate numbers affects the ranks of subsequent numbers.

The following examples show the results of inserting the Rank function in two separate tables.

**Example 1**

In this example, the Rank function is used for the numeric column (the column that contains the numbers to rank). The results are shown in the Computed column.

Rank(Amount)

<table>
<thead>
<tr>
<th>State</th>
<th>City</th>
<th>Amount</th>
<th>Computed</th>
</tr>
</thead>
<tbody>
<tr>
<td>AZ</td>
<td>Tucson</td>
<td>112</td>
<td>6</td>
</tr>
<tr>
<td>CA</td>
<td>Burbank</td>
<td>241</td>
<td>2</td>
</tr>
<tr>
<td>CA</td>
<td>Glendale</td>
<td>97</td>
<td>7</td>
</tr>
<tr>
<td>FL</td>
<td>Palmetto</td>
<td>70</td>
<td>9</td>
</tr>
<tr>
<td>MD</td>
<td>Laurel</td>
<td>97</td>
<td>7</td>
</tr>
<tr>
<td>MI</td>
<td>Detroit</td>
<td>208</td>
<td>3</td>
</tr>
</tbody>
</table>
**RankAsc**

The `RankAsc` function returns the rank of a number in a column of numbers. It works as if you were to sort the list in ascending order. In this case, the rank of the number would be its position.

`RankAsc(numbers, break_col)`

where:

- `numbers` references the column that contains the numbers on which the rank is calculated.
- `break_col` is an optional parameter that references a break column.

**Note**: The `RankAsc` function assigns duplicate numbers the same rank. The presence of duplicate numbers affects the ranks of subsequent numbers.

The following examples show the results of inserting the `RankAsc` function in two separate tables.

**Example 1**

In this example, the `RankAsc` function is used on the numeric column (the column containing the numbers to rank). The results are shown in the Computed column.

`RankAsc(Amount)`

<table>
<thead>
<tr>
<th>State</th>
<th>City</th>
<th>Amount</th>
<th>Computed</th>
</tr>
</thead>
<tbody>
<tr>
<td>AZ</td>
<td>Tucson</td>
<td>112</td>
<td>4</td>
</tr>
<tr>
<td>CA</td>
<td>Burbank</td>
<td>240</td>
<td>8</td>
</tr>
<tr>
<td>CA</td>
<td>Glendale</td>
<td>97</td>
<td>2</td>
</tr>
<tr>
<td>FL</td>
<td>Palmetto</td>
<td>70</td>
<td>1</td>
</tr>
<tr>
<td>MD</td>
<td>Laurel</td>
<td>97</td>
<td>2</td>
</tr>
<tr>
<td>MI</td>
<td>Detroit</td>
<td>208</td>
<td>7</td>
</tr>
<tr>
<td>MN</td>
<td>Eagan</td>
<td>159</td>
<td>5</td>
</tr>
<tr>
<td>NY</td>
<td>New York</td>
<td>490</td>
<td>9</td>
</tr>
</tbody>
</table>
Example 2
In this example, the `RankAsc` function is used on the numeric column and `break_column`. The results are shown in the `Computed` column.

`RankAsc (Amount, State)`

<table>
<thead>
<tr>
<th>State</th>
<th>City</th>
<th>Amount</th>
<th>Computed</th>
</tr>
</thead>
<tbody>
<tr>
<td>NY</td>
<td>Rochester</td>
<td>180</td>
<td>6</td>
</tr>
</tbody>
</table>

Table 136  RankAsc Example 2

<table>
<thead>
<tr>
<th>State</th>
<th>City</th>
<th>Amount</th>
<th>Computed</th>
</tr>
</thead>
<tbody>
<tr>
<td>AZ</td>
<td>Tucson</td>
<td>112</td>
<td>3</td>
</tr>
<tr>
<td>CA</td>
<td>Burbank</td>
<td>240</td>
<td>6</td>
</tr>
<tr>
<td>CA</td>
<td>Glendale</td>
<td>97</td>
<td>6</td>
</tr>
<tr>
<td>FL</td>
<td>Palmetto</td>
<td>70</td>
<td>1</td>
</tr>
<tr>
<td>MD</td>
<td>Laurel</td>
<td>97</td>
<td>2</td>
</tr>
<tr>
<td>MI</td>
<td>Detroit</td>
<td>208</td>
<td>5</td>
</tr>
<tr>
<td>MN</td>
<td>Eagan</td>
<td>159</td>
<td>4</td>
</tr>
<tr>
<td>NY</td>
<td>New York</td>
<td>490</td>
<td>7</td>
</tr>
<tr>
<td>NY</td>
<td>Rochester</td>
<td>180</td>
<td>7</td>
</tr>
</tbody>
</table>

Note: The `RankAsc` is calculated on the `SUM(col_1)` and grouped by `col_2`.

**StdDev**

The `StdDev` function returns a standard deviation based on a sample. It is a measure of how widely values are dispersed from the average value (the mean).

`StdDev (numbers, break_col)`

where:

`numbers` references the column that contains the numbers on which the standard deviation is calculated.

`break_col` is an optional parameter that references a break column.

Note: `StdDev` assumes that its arguments are a sample of the population. If you data represents the entire population, then compute the standard deviation using `StdDevp`.

Note: The standard deviation is calculated using the nonbiased or n-1 method.
Note: If a result set contains one row of data or less, the **StdDev** function should return an error.

The following examples show the results of inserting the **StdDev** function in two separate tables.

**Example**

In this example, the **StdDev** function is used on the numeric column. The results are shown in the Computed column.

```
StdDev (Amount)
```

**StdDevp**

The **StdDevp** function calculates and returns a standard deviation based on the entire population given as arguments. The standard deviation is a measure of how widely values are dispersed from the average value (the mean).

```
StdDevp (numbers, break_col)
```

where:

- `numbers` references the column that contains the numbers on which the standard deviation is calculated.
- `break_col` is an optional parameter that references a break column.

Note: **StdDevp** assumes that its arguments are the entire population. If your data represents a sample of the population, then compute the standard deviation using **StdDev**.

Note: The standard deviation is calculated using the *biased* or *n* method.

The following examples show the results of inserting the **StdDevp** function in two separate tables.

**Example**

In this example, the **StdDevp** function is used for the numeric column. The results are shown in the Computed column.

```
StdDevp (Amount)
```

<table>
<thead>
<tr>
<th>State</th>
<th>City</th>
<th>Amount</th>
<th>Computed</th>
</tr>
</thead>
<tbody>
<tr>
<td>AZ</td>
<td>Tucson</td>
<td>112</td>
<td>120.79</td>
</tr>
<tr>
<td>CA</td>
<td>Burbank</td>
<td>240</td>
<td>120.79</td>
</tr>
<tr>
<td>CA</td>
<td>Glendale</td>
<td>97</td>
<td>120.79</td>
</tr>
</tbody>
</table>
The `Sum` function computes the total for a column of numbers.

\[
\text{Sum (numbers, break\_col, break\_value)}
\]

where:

- `numbers` references the column that contains the numbers on which the sum is calculated.
- `break\_col` is an optional parameter that references a break column.
- `break\_value` is an optional parameter that returns the sum of numbers column where value in `break\_col` equals `break\_value`.

**Note:** If constant values in the `break\_value` column are substituted for data items, dates and text strings must be enclosed in single quotes.

The following examples show the results of inserting the `Sum` function in two separate tables.

**Example 1**

In this example, the `Sum` function is used on the numeric column. The results are shown in the `Computed` column.

\[
\text{Sum(Amount)}
\]

<table>
<thead>
<tr>
<th>State</th>
<th>City</th>
<th>Amount</th>
<th>Computed</th>
</tr>
</thead>
<tbody>
<tr>
<td>AZ</td>
<td>Tucson</td>
<td>112</td>
<td>1556</td>
</tr>
<tr>
<td>CA</td>
<td>Burbank</td>
<td>240</td>
<td>1556</td>
</tr>
<tr>
<td>CA</td>
<td>Glendale</td>
<td>NULL</td>
<td>1556</td>
</tr>
<tr>
<td>FL</td>
<td>Palmetto</td>
<td>70</td>
<td>1556</td>
</tr>
<tr>
<td>MD</td>
<td>Laurel</td>
<td>97</td>
<td>1556</td>
</tr>
</tbody>
</table>
Example 2

In this example, the `Sum` function has been inserted for the numeric column and `break_column` using the following definition:

```plaintext
Sum (Amount, State)
```

The results are shown in the Computed column.

<table>
<thead>
<tr>
<th>State</th>
<th>City</th>
<th>Amount</th>
<th>Computed</th>
</tr>
</thead>
<tbody>
<tr>
<td>AZ</td>
<td>Tuscon</td>
<td>112</td>
<td>112</td>
</tr>
<tr>
<td>CA</td>
<td>Burbank</td>
<td>240</td>
<td>240</td>
</tr>
<tr>
<td>CA</td>
<td>Glendale</td>
<td>Null</td>
<td>240</td>
</tr>
<tr>
<td>FL</td>
<td>Palmetto</td>
<td>70</td>
<td>70</td>
</tr>
<tr>
<td>MD</td>
<td>Laurel</td>
<td>97</td>
<td>97</td>
</tr>
<tr>
<td>MI</td>
<td>Detroit</td>
<td>208</td>
<td>208</td>
</tr>
<tr>
<td>MN</td>
<td>Eagan</td>
<td>159</td>
<td>159</td>
</tr>
<tr>
<td>NY</td>
<td>New York</td>
<td>490</td>
<td>670</td>
</tr>
<tr>
<td>NY</td>
<td>Rochester</td>
<td>180</td>
<td>670</td>
</tr>
</tbody>
</table>

Var

The `Var` function estimates variance based on a sample.

```plaintext
Var (numbers, break_col)
```

where:

- `numbers` references the column that contains the numbers on which the variance is calculated.
- `break_col` is an optional parameter that references a break column.

Note: `Var` assumes that its arguments are a sample of the population. If your data represents the entire population, then compute the variance using `Varp`.
The following examples show the results of inserting the `Var` function in two separate tables.

**Example 1**

In this example, the `Var` function is used on the numeric column. The results are shown in the Computed column.

`Var(Amount)`

<table>
<thead>
<tr>
<th>State</th>
<th>City</th>
<th>Amount</th>
<th>Computed</th>
</tr>
</thead>
<tbody>
<tr>
<td>AZ</td>
<td>Tucson</td>
<td>112</td>
<td>16,413.25</td>
</tr>
<tr>
<td>CA</td>
<td>Burbank</td>
<td>240</td>
<td>16,413.25</td>
</tr>
<tr>
<td>CA</td>
<td>Glendale</td>
<td>97</td>
<td>16,413.25</td>
</tr>
<tr>
<td>FL</td>
<td>Palmetto</td>
<td>70</td>
<td>16,413.25</td>
</tr>
<tr>
<td>MD</td>
<td>Laurel</td>
<td>97</td>
<td>16,413.25</td>
</tr>
<tr>
<td>MI</td>
<td>Detroit</td>
<td>208</td>
<td>16,413.25</td>
</tr>
<tr>
<td>MN</td>
<td>Eagan</td>
<td>159</td>
<td>16,413.25</td>
</tr>
<tr>
<td>NY</td>
<td>New York</td>
<td>490</td>
<td>16,413.25</td>
</tr>
<tr>
<td>NY</td>
<td>Rochester</td>
<td>180</td>
<td>16,413.25</td>
</tr>
</tbody>
</table>

**Example 2**

In this example, the `Var` function is used on the numeric column and `break_column`. The results are shown in the Computed column.

`Var(Amount, State)`

<table>
<thead>
<tr>
<th>State</th>
<th>City</th>
<th>Amount</th>
<th>Computed</th>
</tr>
</thead>
<tbody>
<tr>
<td>AZ</td>
<td>Tucson</td>
<td>112</td>
<td>0</td>
</tr>
<tr>
<td>CA</td>
<td>Burbank</td>
<td>240</td>
<td>10,224.50</td>
</tr>
<tr>
<td>CA</td>
<td>Glendale</td>
<td>97</td>
<td>10,224.50</td>
</tr>
<tr>
<td>FL</td>
<td>Palmetto</td>
<td>70</td>
<td>0</td>
</tr>
<tr>
<td>MD</td>
<td>Laurel</td>
<td>97</td>
<td>0</td>
</tr>
<tr>
<td>MI</td>
<td>Detroit</td>
<td>208</td>
<td>0</td>
</tr>
<tr>
<td>MN</td>
<td>Eagan</td>
<td>159</td>
<td>0</td>
</tr>
<tr>
<td>NY</td>
<td>New York</td>
<td>490</td>
<td>48,050.00</td>
</tr>
</tbody>
</table>
The `Varp` function estimates variance based on the entire population.

`Varp (numbers, break_col)`

where:

`numbers` references the column that contains the numbers on which the variance is calculated.

`break_col` is an optional parameter that references a break column.

Note: `Varp` assumes that its arguments are the entire population. If your data represents a sample of the population, then compute the variance using `Var`.

The following examples show the results of inserting the `Varp` function in two separate tables.

**Example**

In this example, the `Varp` function is used on the numeric column. The results are shown in the `Computed` column.

`Varp (Amount)`

**Table 142  Varp Example**

<table>
<thead>
<tr>
<th>State</th>
<th>City</th>
<th>Amount</th>
<th>Computed</th>
</tr>
</thead>
<tbody>
<tr>
<td>AZ</td>
<td>Tucson</td>
<td>112</td>
<td>14,589.56</td>
</tr>
<tr>
<td>CA</td>
<td>Burbank</td>
<td>240</td>
<td>14,589.56</td>
</tr>
<tr>
<td>CA</td>
<td>Glendale</td>
<td>97</td>
<td>14,589.56</td>
</tr>
<tr>
<td>FL</td>
<td>Palmetto</td>
<td>70</td>
<td>14,589.56</td>
</tr>
<tr>
<td>MD</td>
<td>Laurel</td>
<td>97</td>
<td>14,589.56</td>
</tr>
<tr>
<td>MI</td>
<td>Detroit</td>
<td>208</td>
<td>14,589.56</td>
</tr>
<tr>
<td>MN</td>
<td>Eagan</td>
<td>159</td>
<td>14,589.56</td>
</tr>
<tr>
<td>NY</td>
<td>New York</td>
<td>490</td>
<td>14,589.56</td>
</tr>
<tr>
<td>NY</td>
<td>Rochester</td>
<td>180</td>
<td>14,589.56</td>
</tr>
</tbody>
</table>
Trend Functions

Both the Pivot and Results sections offer trend functions, which allow you to perform statistical analysis within a sample data set of variable size. Trend functions are useful for removing data irregularities/fluctuations, analyzing data trends, and smoothing a set of data points. They are often used to reduce noise that exists when visualizing large data sets by providing aggregation capabilities at configurable window sizes.

Review these topics for information:

- General Moving Average and Moving Function Functionality
- Simple Moving Averages
- Positioning of Moving Average Results—Trailing and Centered Averages
- “Weighted Moving Averages” on page 389
- “Exponential Moving Averages” on page 391
- “Moving Diff” on page 392
- “Moving Maximum” on page 393
- “Moving Median” on page 395
- “Moving Sum” on page 396
- “Moving Minimum” on page 397
- “Direction Of Moving Function Calculation” on page 398

General Moving Average and Moving Function Functionality

The term "Moving" when used with functions such as Moving Averages usually refers to a fixed window or "sub-range" of analysis that moves over a larger range of numeric data values. For each window, a calculation such as an average (mean) is performed. As the window moves over the larger range, one number value is dropped from the window and a new value is added.

To illustrate the effect of Moving Averages as opposed to a simple average (mean) consider the following list of arbitrary "Sales" values over a period of twelve months as shown in a Table or Results section:
The simple mean value of all of the above twelve Sales values is:

\[
\text{Sum of all twelve values divided by 12} = \frac{296}{12} = 24.67
\]

While having some virtue, this simple mean value does not offer any insight into analyzing the variation trend of the original values or to provide a more accurate mean value at varying points of the range. Moving Averages, on the other hand, do offer the ability to understand the trend of data by calculating many average (mean) values over the entire range of original data values.

There are several variations to Moving Average calculation, the most common of these are: Simple Moving Averages, Weighted Moving Averages and Exponential Moving Averages. All of these variations are described in detail in the following sections. Additionally, other "Moving Functions": Moving Maximum, Moving Minimum, Moving Median, Moving Sum and Moving Difference whose functionality closely resembles that of Moving Averages are discussed.

### Simple Moving Averages

In a Simple Moving Average calculation, the original range of data values is split into smaller "windows" and a simple mean value of each window is calculated. An example of the calculation involved for such Simple Moving Averages is shown below.

<table>
<thead>
<tr>
<th></th>
<th>Month</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Jan</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>Feb</td>
<td>15</td>
</tr>
<tr>
<td>3</td>
<td>Mar</td>
<td>17</td>
</tr>
<tr>
<td>4</td>
<td>Apr</td>
<td>20</td>
</tr>
<tr>
<td>5</td>
<td>May</td>
<td>22</td>
</tr>
<tr>
<td>6</td>
<td>Jun</td>
<td>20</td>
</tr>
<tr>
<td>7</td>
<td>Jul</td>
<td>25</td>
</tr>
<tr>
<td>8</td>
<td>Aug</td>
<td>27</td>
</tr>
<tr>
<td>9</td>
<td>Sep</td>
<td>30</td>
</tr>
<tr>
<td>10</td>
<td>Oct</td>
<td>35</td>
</tr>
<tr>
<td>11</td>
<td>Nov</td>
<td>37</td>
</tr>
<tr>
<td>12</td>
<td>Dec</td>
<td>40</td>
</tr>
</tbody>
</table>
If the first three Sales values were summed and then this sum was divided by 3 to give a mean value, this value would be:

\[
\frac{10 + 15 + 17}{3} = \frac{42}{3} = 14
\]

Next, consider the mean of the sum of the second, third and fourth original values i.e.:

\[
\frac{15 + 17 + 20}{3} = \frac{52}{3} = 17.33
\]

This pattern could be repeated to reveal the mean values of the third, fourth and fifth values; fifth, sixth and seventh values and so on until all subsequent windows of three numbers are summed and their mean values obtained.

The Simple Moving Averages of the original range of values for a window of 3 (i.e. in this case, a 3-Month Simple Moving Average) could be evaluated to be:

<table>
<thead>
<tr>
<th>Month</th>
<th>Original Sales Values</th>
<th>3-Month Simple Moving Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Feb</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>Mar</td>
<td>17</td>
<td>14</td>
</tr>
<tr>
<td>Apr</td>
<td>20</td>
<td>17.33</td>
</tr>
<tr>
<td>May</td>
<td>22</td>
<td>19.66</td>
</tr>
<tr>
<td>Jun</td>
<td>20</td>
<td>20.66</td>
</tr>
<tr>
<td>Jul</td>
<td>25</td>
<td>22.33</td>
</tr>
<tr>
<td>Aug</td>
<td>27</td>
<td>24</td>
</tr>
<tr>
<td>Sep</td>
<td>30</td>
<td>27.33</td>
</tr>
<tr>
<td>Oct</td>
<td>35</td>
<td>30.66</td>
</tr>
</tbody>
</table>
Positioning of Moving Average Results—Trailing and Centered Averages

Note that in Simple Moving Average table, the average of numbers n, n+1 and n+2 in the "Original Values" column (where "n" refers to the row position) is placed in row position n+2 of the "3-Month Simple Moving Average" column. This Moving Average display technique is known as "Trailing Averages". An alternative display technique is known as "Centered Averages" which instead positions the Moving Average in the center row of the window. The table following illustrates the difference in these display techniques using the first three values from above:

<table>
<thead>
<tr>
<th>Month</th>
<th>Centered Averages</th>
<th>Trailing Averages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feb</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>Mar</td>
<td></td>
<td>14</td>
</tr>
</tbody>
</table>

"Centered Averages" display requires further calculations when the window is an even number and it is not available for Simple Moving Averages and other Moving Functions at this time.

All "Moving Functions" in this particular implementation will display data according to the "Trailing Averages" principle.

Note also that from the above two tables, "Trailing Averages" display causes the initial n-1 (where n = window size) rows of result data to have no value (rows 1 and 2 are blank in the above examples). This is the generally accepted standard for the initial "n-1" terms and is the standard adopted for the implementation of most Moving Functions.

The following table illustrates the above monthly Sales data Simple Moving Average calculation using "Trailing Averages" display:

The Simple Moving Averages of the original range of values for a window of 3 (i.e. in this case, a 3-Month Simple Moving Average) could be evaluated to be:

<table>
<thead>
<tr>
<th>Month</th>
<th>Original Sales Values</th>
<th>3-Month Simple Moving Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Feb</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>Mar</td>
<td>17</td>
<td>14</td>
</tr>
</tbody>
</table>
Month | Original Sales Values | 3-Month Simple Moving Average
---|---|---
Apr | 20 | 17.33
May | 22 | 19.66
Jun | 20 | 20.66
Jul | 25 | 22.33
Aug | 27 | 24
Sep | 30 | 27.33
Oct | 35 | 30.66
Nov | 37 | 34
Dec | 40 | 37.33

**Weighted Moving Averages**

With Simple Moving Averages, each data value in the "window" in which the calculation is performed is given an equal significance or weight. It is often the case, especially in financial price data analysis, that more chronologically recent data should carry a greater weight. In these cases, Weighted Moving Average (or Exponential Moving Average - see the following topic) functionality is often preferred.

Consider the same table of Sales data values for twelve months:

<table>
<thead>
<tr>
<th>Month</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan</td>
<td>10</td>
</tr>
<tr>
<td>Feb</td>
<td>15</td>
</tr>
<tr>
<td>Mar</td>
<td>17</td>
</tr>
<tr>
<td>Apr</td>
<td>20</td>
</tr>
<tr>
<td>May</td>
<td>22</td>
</tr>
<tr>
<td>Jun</td>
<td>20</td>
</tr>
<tr>
<td>Jul</td>
<td>25</td>
</tr>
<tr>
<td>Aug</td>
<td>27</td>
</tr>
<tr>
<td>Sep</td>
<td>30</td>
</tr>
<tr>
<td>Oct</td>
<td>35</td>
</tr>
<tr>
<td>Nov</td>
<td>37</td>
</tr>
<tr>
<td>Dec</td>
<td>40</td>
</tr>
</tbody>
</table>

> To calculate a Weighted Moving Average:

1. Calculate how many intervals of data are participating in the Moving Average calculation (i.e. the size of the calculation "window").
If the calculation window is said to be n, then the most recent data value in the window is multiplied by n, the next most recent multiplied by n-1, the value prior to that multiplied by n-2 and so on for all values in the Window.

2 Divide the sum of all of the multiplied values by the sum of the weights to give the Weighted Moving Average over that window.

3 Place the Weighted Moving Average value in a new column according to the trailing averages positioning described above.

To illustrate these steps, consider if a 3-month Weighted Moving Average of Sales in December is required (using the above table of Sales values).

The term "3-month" implies that the calculation "window" is 3, therefore the Weighted Moving Average calculation algorithm for this case should be:

\[
\frac{((\text{Dec Sales value} \times 3) + (\text{Nov Sales value} \times 2) + (\text{Oct Sales value} \times 1))}{(3 + 2 + 1)}
\]

\[
= \frac{(40 \times 3) + (37 \times 2) + (35 \times 1)}{6}
\]

\[
= \frac{120 + 74 + 35}{6}
\]

\[
= \frac{229}{6}
\]

\[
= 38.17
\]

Or, if a 3-month Weighted Moving Average were evaluated over the entire original range of data, the results would be:

<table>
<thead>
<tr>
<th>Month</th>
<th>Original Sales Values</th>
<th>3-Month Simple Moving Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Feb</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>Mar</td>
<td>17</td>
<td>15.12</td>
</tr>
<tr>
<td>Apr</td>
<td>20</td>
<td>18.17</td>
</tr>
<tr>
<td>May</td>
<td>22</td>
<td>20.5</td>
</tr>
<tr>
<td>Jun</td>
<td>20</td>
<td>20.67</td>
</tr>
<tr>
<td>Jul</td>
<td>25</td>
<td>22.83</td>
</tr>
<tr>
<td>Aug</td>
<td>27</td>
<td>25.17</td>
</tr>
<tr>
<td>Sep</td>
<td>30</td>
<td>28.17</td>
</tr>
<tr>
<td>Oct</td>
<td>35</td>
<td>32</td>
</tr>
<tr>
<td>Nov</td>
<td>37</td>
<td>35.17</td>
</tr>
<tr>
<td>Dec</td>
<td>40</td>
<td>38.17</td>
</tr>
</tbody>
</table>
Exponential Moving Averages

Exponential Moving Averages, similar to Weighted Moving Averages, also assign a greater weight to more recent data values. Unlike Weighted Moving Averages, however, they use the previously calculated Exponential Moving Average value as a basis for calculation rather than the original (non-Averaged) data values. In this way, the calculation method used by Exponential Moving Averages is cumulative, meaning that (unlike Simple Moving Averages or Weighted Moving Averages) all previous data values have some effect on the Exponential Moving Average to be calculated, although this effect diminishes greatly with time.

Exponential Moving Averages tend to be more accurate than the other types of Moving Average when the original data values show a more rapid degree of variation over time (or other variable).

The formula for calculating an Exponential Moving Average (EMA) is:

\[ X = (K \times (C - P)) + P \]

Where:

\( X \) = Current EMA (i.e. EMA to be calculated)

\( C \) = Current original data value

\( K \) = Smoothing Constant

\( P \) = Previous EMA

(The first EMA in the range to be calculated is arbitrary and can be the corresponding original data value or, often, a Simple Moving Average value.

Where:

\( K = \text{Smoothing Constant} = \frac{2}{1 + n} \)

\( n \) = number of periods for EMA i.e. the Window to calculate.

This rather complex calculation is, perhaps, best illustrated by example.

Consider the table of monthly Sales values as shown previously:

<table>
<thead>
<tr>
<th>Month</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan 1</td>
<td>10</td>
</tr>
<tr>
<td>Feb 2</td>
<td>15</td>
</tr>
<tr>
<td>Mar 3</td>
<td>17</td>
</tr>
<tr>
<td>Apr 4</td>
<td>20</td>
</tr>
<tr>
<td>May 5</td>
<td>22</td>
</tr>
<tr>
<td>Jun 6</td>
<td>20</td>
</tr>
<tr>
<td>Jul 7</td>
<td>25</td>
</tr>
<tr>
<td>Aug 8</td>
<td>27</td>
</tr>
<tr>
<td>Sep 9</td>
<td>30</td>
</tr>
<tr>
<td>Oct 10</td>
<td>35</td>
</tr>
<tr>
<td>Nov 11</td>
<td>37</td>
</tr>
<tr>
<td>Dec 12</td>
<td>40</td>
</tr>
</tbody>
</table>
If we calculated the Exponential Moving Average in a similar fashion to the 3-Month Simple Moving Average, we would perform the following steps:

1. **Calculate the Smoothing Constant according to the \( \frac{2}{1 + n} \) formula.**
   
   \( N \) = window of values = 3, therefore the Smoothing Constant is:
   
   \[ \frac{2}{1 + 3} = 0.5 \]

2. **For the first Exponential Moving Average, use the first original data value (in this case, that for the Month of "Jan").**

3. **For subsequent values, follow the calculation according to the above formula.**
   
   \[ X = (K \times (C - P)) + P \] as indicated below:

<table>
<thead>
<tr>
<th>Month</th>
<th>Sales</th>
<th>Calculation ( (K \times (C - P)) + P )</th>
<th>Exponential Moving Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan</td>
<td>10</td>
<td>Original value</td>
<td>10</td>
</tr>
<tr>
<td>Feb</td>
<td>15</td>
<td>( 0.5 \times (15 - 10) + 10 )</td>
<td>12.5</td>
</tr>
<tr>
<td>Mar</td>
<td>17</td>
<td>( 0.5 \times (17 - 12.5) + 12.5 )</td>
<td>14.75</td>
</tr>
<tr>
<td>Apr</td>
<td>20</td>
<td>( 0.5 \times (20 - 14.75) + 14.75 )</td>
<td>17.375</td>
</tr>
<tr>
<td>May</td>
<td>22</td>
<td>( 0.5 \times (22 - 17.375) + 17.375 )</td>
<td>19.688</td>
</tr>
<tr>
<td>Jun</td>
<td>20</td>
<td>( 0.5 \times (20 - 19.6875) + 19.6875 )</td>
<td>19.844</td>
</tr>
<tr>
<td>Jul</td>
<td>25</td>
<td>( 0.5 \times (25 - 19.844) + 19.844 )</td>
<td>22.422</td>
</tr>
<tr>
<td>Aug</td>
<td>27</td>
<td>( 0.5 \times (27 - 22.422) + 22.422 )</td>
<td>24.711</td>
</tr>
<tr>
<td>Sep</td>
<td>30</td>
<td>( 0.5 \times (30 - 24.711) + 24.711 )</td>
<td>27.355</td>
</tr>
<tr>
<td>Oct</td>
<td>35</td>
<td>( 0.5 \times (35 - 27.355) + 27.355 )</td>
<td>31.178</td>
</tr>
<tr>
<td>Nov</td>
<td>37</td>
<td>( 0.5 \times (37 - 31.178) + 31.178 )</td>
<td>34.089</td>
</tr>
<tr>
<td>Dec</td>
<td>40</td>
<td>( 0.5 \times (40 - 34.089) + 34.089 )</td>
<td>37.044</td>
</tr>
</tbody>
</table>

### Moving Diff

The MovingDiff function operates over a moving window of values, and returns the difference between the current value and the value at the beginning of the window. This function is similar to the MovingMax function. No “Weighted” or “Exponential” can be calculated for this function.

The MovingDiff function takes the following arguments:

MovingDiff (column, window, break_col)
where:

- **column** (required)—Specify the column that contains the numeric fact on which the Moving Function column value is calculated. If you are applying the function from the Pivot section, you can only use a column that has already been added to the Facts pane of the data layout.

- **window** (optional)—Specify a moving “window” of values from the Column on which the Moving Function gets calculated. The window value must be a positive integer of value equal to or less than the total number of rows in the Column (within any Break Column value). If the window value is an integer of greater value than the number of values in the Column, then the window value defaults to the number of rows in the Column (within any Break Column value). If no window value is specified, then the window value defaults to 3.

- **break column** (optional)—Specify the name of the column on which to perform the Moving Function calculation. The break column refers to the rows of similar valued data cell within the Pivot or Results section.

The following results would occur if you used the Moving Diff function with a Window of 3 (MovingDiff(Sales,3)).

<table>
<thead>
<tr>
<th>Month</th>
<th>Original Sales Values</th>
<th>3-Month MovingDiff Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Feb</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>Mar</td>
<td>17</td>
<td>7</td>
</tr>
<tr>
<td>Apr</td>
<td>20</td>
<td>5</td>
</tr>
<tr>
<td>May</td>
<td>22</td>
<td>5</td>
</tr>
<tr>
<td>Jun</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>Jul</td>
<td>25</td>
<td>3</td>
</tr>
<tr>
<td>Aug</td>
<td>27</td>
<td>7</td>
</tr>
<tr>
<td>Sep</td>
<td>30</td>
<td>5</td>
</tr>
<tr>
<td>Oct</td>
<td>35</td>
<td>5</td>
</tr>
<tr>
<td>Nov</td>
<td>37</td>
<td>7</td>
</tr>
<tr>
<td>Dec</td>
<td>40</td>
<td>5</td>
</tr>
</tbody>
</table>

### Moving Maximum

The MovingMax (Moving Maximum) function operates over a moving window of values. For each Window, the MovingMax returns the maximum value found in the Window. This function
The MovingMax function takes the following arguments:

`MovingMax (column, window, break_col)`

where:

- `column` (required)—Specify the column that contains the numeric fact on which the Moving Function column value is calculated. If you are applying the function from the Pivot section, you can only use a column that has already been added to the Facts pane of the data layout.

- `window` (optional)—Specify a moving “window” of values from the Column on which the Moving Function gets calculated. The window value must be a positive integer of value equal to or less than the total number of rows in the Column (within any Break Column value). If the window value is an integer of greater value than the number of values in the Column, then the window value defaults to the number of rows in the Column (within any Break Column value). If no window value is specified, then the window value defaults to 3.

- `break column` (optional)—Specify the name of the column on which to perform the Moving Function calculation. The break column refers to the rows of similar valued data cell within the Pivot or Results section.

The following results would occur if you used the MovingMax function with a Window of 3 (MovingMax(Sales,3)).

<table>
<thead>
<tr>
<th>Month</th>
<th>Original Sales Values</th>
<th>3-Month MovingMax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Feb</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>Mar</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>Apr</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>May</td>
<td>22</td>
<td>22</td>
</tr>
<tr>
<td>Jun</td>
<td>20</td>
<td>22</td>
</tr>
<tr>
<td>Jul</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td>Aug</td>
<td>27</td>
<td>27</td>
</tr>
<tr>
<td>Sep</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>Oct</td>
<td>35</td>
<td>35</td>
</tr>
<tr>
<td>Nov</td>
<td>37</td>
<td>37</td>
</tr>
<tr>
<td>Dec</td>
<td>40</td>
<td>40</td>
</tr>
</tbody>
</table>
### Moving Median

The MovingMed function operates over a moving window of values. For each Window, the MovingMed returns the middle value (in terms of rank) found in the Window. If the Window value is an even number, the simple mean value of the two middle ranking values should be used. No “Weighted” or “Exponential” can be calculated for this function.

The MovingMed function takes the following arguments:

MovingMed (column, window, break_col)

where:

- **column** (required)—Specify the column that contains the numeric fact on which the Moving Function column value is calculated. If you are applying the function from the Pivot section, you can only use a column that has already been added to the Facts pane of the data layout.

- **window** (optional)—Specify a moving “window” of values from the Column on which the Moving Function gets calculated. The window value must be a positive integer of value equal to or less than the total number of rows in the Column (within any Break Column value).

  If the window value is an integer of greater value than the number of values in the Column then the window value defaults to the number of rows in the Column (within any Break Column value). If no window value is specified, then the window value defaults to 3.

- **break column** (optional)—Specify the name of the column on which to perform the Moving Function calculation. The break column refers to the rows of similar valued data cell within the Pivot or Results section.

The following results would occur if you used the MovingMed function with a Window of 3 (MovingMed(Sales,3)).

<table>
<thead>
<tr>
<th>Month</th>
<th>Original Sales Values</th>
<th>3-Month MovingMed Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Feb</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>Mar</td>
<td>17</td>
<td>15</td>
</tr>
<tr>
<td>Apr</td>
<td>20</td>
<td>17</td>
</tr>
<tr>
<td>May</td>
<td>22</td>
<td>20</td>
</tr>
<tr>
<td>Jun</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>Jul</td>
<td>25</td>
<td>22</td>
</tr>
<tr>
<td>Aug</td>
<td>27</td>
<td>25</td>
</tr>
<tr>
<td>Sep</td>
<td>30</td>
<td>27</td>
</tr>
<tr>
<td>Oct</td>
<td>35</td>
<td>30</td>
</tr>
</tbody>
</table>
Moving Sum

MovingSum (Moving Sum) function operates over a moving window of values. For each Window, the MovingSum returns the sum of values found in the window. No “Weighted” or “Exponential” can be calculated for this function.

The MovingSum function takes the following arguments:

MovingSum (column, window, break_col)

where:

- `column` (required)—Specify the column that contains the numeric fact on which the Moving Function column value is calculated. If you are applying the function from the Pivot section, you can only use a column that has already been added to the Facts pane of the data layout.

- `window` (optional)—Specify a moving “window” of values from the Column on which the Moving Function gets calculated. The window value must be a positive integer of value equal to or less than the total number of rows in the Column (within any Break Column value). If the window value is an integer of greater value than the number of values in the Column, then the window value defaults to the number of rows in the Column (within any Break Column value). If no window value is specified, then the window value defaults to 3.

- `break_column` (optional) —Specify the name of the column on which to perform the Moving Function calculation. The break column refers to the rows of similar valued data cell within the Pivot or Results section.

The following results would occur if you used the Moving Sum function with a Window value of 3.

Table 151  Moving Sum Example

<table>
<thead>
<tr>
<th>Month</th>
<th>Original Sales Values</th>
<th>3-Month MovingSum Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Feb</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>Mar</td>
<td>17</td>
<td>42</td>
</tr>
<tr>
<td>Apr</td>
<td>20</td>
<td>52</td>
</tr>
<tr>
<td>May</td>
<td>22</td>
<td>59</td>
</tr>
<tr>
<td>Jun</td>
<td>20</td>
<td>62</td>
</tr>
<tr>
<td>Month</td>
<td>Original Sales Values</td>
<td>3-Month MovingSum Average</td>
</tr>
<tr>
<td>-------</td>
<td>-----------------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>Jul</td>
<td>25</td>
<td>72</td>
</tr>
<tr>
<td>Aug</td>
<td>27</td>
<td>72</td>
</tr>
<tr>
<td>Sep</td>
<td>30</td>
<td>82</td>
</tr>
<tr>
<td>Oct</td>
<td>35</td>
<td>92</td>
</tr>
<tr>
<td>Nov</td>
<td>37</td>
<td>102</td>
</tr>
<tr>
<td>Dec</td>
<td>40</td>
<td>112</td>
</tr>
</tbody>
</table>

**Moving Minimum**

The MovingMin function operates over a moving window of values. For each Window, the Moving Minimum returns the minimum value found in the Window. This function is similar to the Simple Moving Average. No “Weighted” or “Exponential” can be calculated for this function.

The MovingMin function takes the following arguments:

MovingMin (column, window, break_col)

where:

- **column** (required)—Specify the column that contains the numeric fact on which the Moving Function column value is calculated. If you are applying the function from the Pivot section, you can only use a column that has already been added to the Facts pane of the data layout.

- **window** (optional)—Specify a moving “window” of values from the Column on which the Moving Function gets calculated. The window value must be a positive integer of value equal to or less than the total number of rows in the Column (within any Break Column value).

  If the window value is an integer of greater value than the number of values in the Column, then the window value defaults to the number of rows in the Column (within any Break Column value). If no window value is specified, then the window value defaults to 3.

- **break column** (optional)—Specify the name of the column on which to perform the Moving Function calculation. The break column refers to the rows of similar valued data cell within the Pivot or Results section.

The following results would occur if you used the Moving Minimum function with a Window value of 3.

Similar to Simple Moving Averages above, note that the values in the "3-Month Moving Minimum" column are positioned according to the trailing averages.
Table 152  Moving Minimum Example

<table>
<thead>
<tr>
<th>Month</th>
<th>Original Sales Values</th>
<th>3-Month Moving Minimum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Feb</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>Mar</td>
<td>17</td>
<td>10</td>
</tr>
<tr>
<td>Apr</td>
<td>20</td>
<td>15</td>
</tr>
<tr>
<td>May</td>
<td>22</td>
<td>17</td>
</tr>
<tr>
<td>Jun</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>Jul</td>
<td>25</td>
<td>20</td>
</tr>
<tr>
<td>Aug</td>
<td>27</td>
<td>20</td>
</tr>
<tr>
<td>Sep</td>
<td>30</td>
<td>25</td>
</tr>
<tr>
<td>Oct</td>
<td>35</td>
<td>27</td>
</tr>
<tr>
<td>Nov</td>
<td>37</td>
<td>30</td>
</tr>
<tr>
<td>Dec</td>
<td>40</td>
<td>35</td>
</tr>
</tbody>
</table>

**Direction Of Moving Function Calculation**

In all of the Moving Function calculation examples, it is assumed that the Moving Function calculation proceeds in a downward direction for each Computed Item column. That is to say, the Moving Function calculation window progresses downward, one row at a time, for each subsequent row of displayed fact data. This downward movement is evident in the Table section data (which has been used in the examples shown above) since, in a Table section, each new instance of fact data can only be represented as a new row.

Other sections, however, such as Pivot and Chart allow the possibility for fact data to be represented in more than one directional axis. In the case of Pivot and Chart sections, therefore, you can specify the direction of Moving Function calculation. To illustrate the effect of changing the direction of Moving Function calculation, consider the previously shown Table data:

Table 153  Moving Function Example

<table>
<thead>
<tr>
<th>Month</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan</td>
<td>10</td>
</tr>
<tr>
<td>Feb</td>
<td>15</td>
</tr>
<tr>
<td>Mar</td>
<td>17</td>
</tr>
<tr>
<td>Apr</td>
<td>20</td>
</tr>
<tr>
<td>May</td>
<td>22</td>
</tr>
</tbody>
</table>
Adding a new Computed Item column to display a 3-Month Moving Difference would yield:

Table 154 3-Month Moving Difference Example

<table>
<thead>
<tr>
<th>Month</th>
<th>Sales</th>
<th>3-Month Moving Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Feb</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>Mar</td>
<td>17</td>
<td>7</td>
</tr>
<tr>
<td>Apr</td>
<td>20</td>
<td>5</td>
</tr>
<tr>
<td>May</td>
<td>22</td>
<td>5</td>
</tr>
<tr>
<td>Jun</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>Jul</td>
<td>25</td>
<td>3</td>
</tr>
<tr>
<td>Aug</td>
<td>27</td>
<td>7</td>
</tr>
<tr>
<td>Sep</td>
<td>30</td>
<td>5</td>
</tr>
<tr>
<td>Oct</td>
<td>35</td>
<td>5</td>
</tr>
<tr>
<td>Nov</td>
<td>37</td>
<td>7</td>
</tr>
</tbody>
</table>

Consider, however, if the "Month" and "Sales" data were oriented as follows (as could be done in a Pivot section):

<table>
<thead>
<tr>
<th>Month</th>
<th>Jan</th>
<th>Feb</th>
<th>Mar</th>
<th>Apr</th>
<th>May</th>
<th>Jun</th>
<th>Jul</th>
<th>Aug</th>
<th>Sep</th>
<th>Oct</th>
<th>Nov</th>
<th>Dec</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sales</td>
<td>10</td>
<td>15</td>
<td>17</td>
<td>20</td>
<td>22</td>
<td>20</td>
<td>25</td>
<td>27</td>
<td>30</td>
<td>35</td>
<td>37</td>
<td>40</td>
</tr>
</tbody>
</table>

If a new Computed Item were added, to represent a 3-Month Moving Difference, and the Moving Function calculation direction was still down each column, this would result in null values being
represented for the Computed Item (as below) since there would only be one fact value per column (this would not be enough fact occurrences to satisfy a 3-term Moving Difference).

<table>
<thead>
<tr>
<th>Month</th>
<th>Jan</th>
<th>Feb</th>
<th>Mar</th>
<th>Apr</th>
<th>May</th>
<th>Jun</th>
<th>Jul</th>
<th>Aug</th>
<th>Sep</th>
<th>Oct</th>
<th>Nov</th>
<th>Dec</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sales</td>
<td>10</td>
<td>15</td>
<td>17</td>
<td>20</td>
<td>22</td>
<td>20</td>
<td>25</td>
<td>27</td>
<td>30</td>
<td>35</td>
<td>37</td>
<td>40</td>
</tr>
<tr>
<td>3-Month Moving Difference</td>
<td>7</td>
<td>5</td>
<td>5</td>
<td>0</td>
<td>3</td>
<td>7</td>
<td>5</td>
<td>7</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

If, instead, the Moving Function calculation direction were specified to be along each row (i.e. horizontally from right to left) the Moving Difference Computed Item would yield the "expected" results since the source "Sales" Fact data cells would be examined in their "correct" left-to-right sequence. That is, the following display would result:

<table>
<thead>
<tr>
<th>Month</th>
<th>Jan</th>
<th>Feb</th>
<th>Mar</th>
<th>Apr</th>
<th>May</th>
<th>Jun</th>
<th>Jul</th>
<th>Aug</th>
<th>Sep</th>
<th>Oct</th>
<th>Nov</th>
<th>Dec</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sales</td>
<td>10</td>
<td>15</td>
<td>17</td>
<td>20</td>
<td>22</td>
<td>20</td>
<td>25</td>
<td>27</td>
<td>30</td>
<td>35</td>
<td>37</td>
<td>40</td>
</tr>
<tr>
<td>3-Month Moving Difference</td>
<td>7</td>
<td>5</td>
<td>5</td>
<td>0</td>
<td>3</td>
<td>7</td>
<td>5</td>
<td>7</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Applying Sorts

In This Chapter

- Sorting Data ................................................................. 401
- Simple Sorts ................................................................. 401
- Sort Lines ................................................................. 402
- Complex Sorting ............................................................ 402
- Nested Sorts ................................................................. 405

Sorting Data

Sorting simplifies the process of data analysis. After data is sorted, the answers to questions are often readily at your fingertips. Sorting is also very useful for ranking data to reveal business trends and margins.

<table>
<thead>
<tr>
<th>Year</th>
<th>State</th>
<th>Item Type</th>
<th>Amounts</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>1997</td>
<td>NY</td>
<td>The Standard</td>
<td>9,958.6</td>
<td>70</td>
</tr>
<tr>
<td>1996</td>
<td>NY</td>
<td>EZ Fax Modern-I</td>
<td>7,203.9</td>
<td>55</td>
</tr>
<tr>
<td>1996</td>
<td>MI</td>
<td>The Turbo</td>
<td>7,986.6</td>
<td>45</td>
</tr>
<tr>
<td>1996</td>
<td>MI</td>
<td>1024 ME Drive</td>
<td>10,438</td>
<td>40</td>
</tr>
<tr>
<td>1996</td>
<td>NY</td>
<td>The Turbo</td>
<td>11,536.2</td>
<td>65</td>
</tr>
<tr>
<td>1996</td>
<td>NY</td>
<td>1024 ME Drive</td>
<td>9,132.25</td>
<td>35</td>
</tr>
<tr>
<td>1996</td>
<td>CA</td>
<td>EZ Fax Modern-I</td>
<td>8,513.7</td>
<td>65</td>
</tr>
<tr>
<td>1995</td>
<td>CA</td>
<td>1024 ME Drive</td>
<td>10,438</td>
<td>40</td>
</tr>
</tbody>
</table>

You can perform simple and complex sorts depending on your requirements. You can sort individual data items or use the Sort line to sort items in reference to associated data values. These complex sorts can be nested, so that the sorted data reflects the hierarchical relationships between data items.

Simple Sorts

If you only want to sort a single Request item, report row, or column, you can use the Sort buttons on the toolbar to quickly order the data.

If you apply simple sort conditions in the Query section, the database server sorts the data while processing the query before it is retrieved to your desktop. Or you can sort data on your desktop.
in Results or report sections. The data associated with a selected item is sorted in ascending or descending order as you wish.

To select items to sort:

1. **Select the data item to sort.**

   In the Query section, select a topic item on the Request line. In other sections, select a corresponding report element in the Content pane.

2. **On the Standard toolbar, click the ascending or descending Sort icon.**

   The data is sorted in the Content pane.

   If the Sort line is visible, the item appears on the Sort line. The item name is followed by an up or down arrow to indicate the sort order.

### Sort Lines

Interactive Reporting sort lines have two functions:

- To maintain a record of sort conditions that you have applied to the data set.
- To enable you to specify compound and **Nested Sorts**.

Sort lines take on a different appearance in each section depending on the data presentation and the types of sorts available.

- In the **Complex Sorting in the Query, Results, and Table Sections** sections, sort lines are drag and drop command lines similar to Request and Filter lines.
- In the **Complex Sorting in Chart, Pivot, and OLAPQuery Reports** sections, sort lines are list driven and include provisions for sorting by aggregate calculations, such as averages and counts.

**Tip:** You can move, size, dock, or hide the Sort line. To toggle the Sort line, click the Sort button on the Section title bar.

### Complex Sorting

In addition to performing **Simple Sorts**, you can use Interactive Reporting to perform complex sorting. Review the following sections for information on:

- **Complex Sorting in the Query, Results, and Table Sections**
- **Complex Sorting in Chart, Pivot, and OLAPQuery Reports**

### Complex Sorting in the Query, Results, and Table Sections

**Note:** The information discussed here also applies to sorting in the Report section.
The appearance and functionality of the Sort line is nearly identical in the Query, Results, and Table sections. In each section, the Sort line uses a drag-and-drop interface similar to the Request and Filter lines.

<table>
<thead>
<tr>
<th>Year</th>
<th>State</th>
<th>Item Type</th>
<th>Amounts</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>1995</td>
<td>CA</td>
<td>1024 ME Drive</td>
<td>7,828.5</td>
<td>30</td>
</tr>
<tr>
<td>1995</td>
<td>CA</td>
<td>1024 ME Drive</td>
<td>10,438</td>
<td>40</td>
</tr>
<tr>
<td>1995</td>
<td>NY</td>
<td>1024 ME Drive</td>
<td>10,438</td>
<td>40</td>
</tr>
<tr>
<td>1996</td>
<td>CA</td>
<td>1024 ME Drive</td>
<td>9,133.25</td>
<td>35</td>
</tr>
<tr>
<td>1996</td>
<td>CA</td>
<td>560 MB Drive</td>
<td>8,428.8</td>
<td>60</td>
</tr>
<tr>
<td>1996</td>
<td>CA</td>
<td>EZ Fax Modern-I</td>
<td>7,858.8</td>
<td>60</td>
</tr>
<tr>
<td>1996</td>
<td>CA</td>
<td>EZ Fax Modern-I</td>
<td>8,513.7</td>
<td>65</td>
</tr>
</tbody>
</table>

Using this feature, you can drag items to the sort line and request them to be sorted in sequence to yield Nested Sorts results.

When you sort more than one data item at a time, the left to right order of data on the Sort line dictates the sort order and creates a nested effect. Data is sorted in the order you specify. The leftmost item on the Sort line is the primary sort. Items to the right are sorted in progression, each within the categories of the preceding item.

For example, if the first item is State, the second item City, and the third item Store, States are sorted alphabetically by name. Within each state, cities are sorted by name. Within each city, stores are sorted.

Nested effects are based solely on the placement of items on the Sort line. If the item order on the Request line differs from the order in the Content pane, the sort is still nested, but the visual impact is not as pronounced.

The key difference between the sections lies in where the sorting is performed:

- In the Query section, the database server sorts items placed on the Sort line as the query is processed. The data is returned already presorted to the Results section.
- In the Results and Table sections, items placed on the Sort line are sorted on your desktop.

The bottom line effect is the same whether you apply sort conditions locally in the Results or Table section or on the database server in the Query section. Depending on the situation and the needs of your business, one method or the other may be preferable.

To use the Sort line in the Query, Results, or Table sections:

1. If the Sort line is not already displayed, click Sort on the section titlebar.
2. Select the desired data item that on the Request line (Query) or its corresponding report element from Elements.
3. Click the ascending or descending Sort button on the Standard toolbar.
   - You can also select the sort options from the section menu or double-click items on the Sort line to toggle between ascending and descending sort order.
4. Review the sorted data.
● If you are sorting in the Query section, the data is sorted by the database server when
you process the query.

● If the Sort line is displayed in the Query section, the item appears on the Sort line. If
you sort in ascending order, an up arrow appears to the right of the item name on the
Sort line; if you sort in descending order, a down arrow appears.

● If you are sorting in the Results or Table section, click Sort Now.

● After sorting, the data in the Results or Table Contents pane is ordered according to the
sort criteria.

Complex Sorting in Chart, Pivot, and OLAPQuery Reports

Note: The information discussed here does not apply to sorting in the Report Designer section.

In Chart, Pivot, and OLAPQuery reports, you generally want to override the default sort order
and sort dimensional data with reference to other data. In these sections, you can use the Sort
line to impose a sort condition for each dimensional data item in your report, and to nest your
sort conditions at each hierarchical level of the report from the outside in.

For example, if a chart lists each type of item your company sells and the total amount sold of
each, initially the item types will be ordered alphabetically. But this data becomes more
meaningful when you instead sort the item types with reference to the total produced by each.
This approach allows you to rank each product type from the highest to lowest total sales.

The Sort line includes three drop-down menus used to define the sort conditions. The contents
of the menus vary depending on the data items in query.

Tip: Data in Chart, Pivot, and OLAPQuery sections is sorted alphabetically by default. You can
use the sort buttons on the Standard toolbar to perform simple sorts on selected report
items and reverse the sort order. (See “Simple Sorts” on page 401 for more information.)

Sort Items

The Sort drop-down menu lists the data items that can be sorted. Each dimensional item included
in the report (name and date) is listed in this menu.

Reference Items

The By drop-down menu lists items used as a basis for a complex sort condition (for example,
sorting Cities by the revenue generated in each). The drop-down menu includes each numerical
data item in the report as well as the keyword entry "label." These choices provide two ways to
sort the dimensional item specified in the Sort menu:

● Sorting by Label—By default Interactive Reporting sorts dimensional data items
   alphabetically by name when you create your report – this is equivalent to sorting by labels.
When selected, *label* indicates that the item chosen from the Sort list is sorted by label or name, rather than by reference to corresponding numeric data values in the report.

- **Sorting by Value**—Sorting by a numeric data item orders each value of the target item chosen from the Sort list by its corresponding numeric value in the Value list.

  Sorting by values produces an entirely different sort order. For example, your chart may list each state in which your company has made sales revenue and the total cost-of-sales for each. The states are initially listed in alphabetical order. When you sort by cost-of-goods, the states are ranked in order by each corresponding cost-of-sales figure.

**Functions**

The Using drop-down menu contains aggregate statistical functions that are available when you sort by values. The functions generally duplicate the data functions available in the active section.

The default function for sorting is *Sum*. When you sort by values, Oracle Hyperion Interactive Reporting sorts dimensional data by the corresponding numeric values of the referenced item (for example, sorting states by the sum total of the cost of goods sold in each state).

To specify a sort using the features of the Sort line:

1. If the Sort line is not already displayed, click **Sort** on the Section title bar.
2. Select an item to sort from the Sort drop-down list.
3. Select a **value** from the By drop-down list as a sort reference, or select **Label** to sort the item alphabetically.
4. If desired, select an aggregate function from the Using drop-down list when sorting by values.
   - The Using drop-down menu is not available when you sort by labels.
5. If desired, click **Sort** on the Sort line.
   - The Sort line stores a sort condition for each dimensional item included in the report.

**Nested Sorts**

When you sort more than one data item at a time, the left to right order of data on the Sort line dictates the sort order and creates a nested effect. The item at left on the Sort line is sorted first. Then items to the right are sorted in progression, each within the categories of the preceding item.

For example, if the first item is State, the second item City, and the third item Store, States are sorted alphabetically by name, and then within each state, cities are sorted by name, and then within each city, stores are sorted.

Nested effects are based solely on the placement of items on the Sort line. If the item order on the Request line differs from the order in the Contents pane, the sort is still nested, but the visual impact is not as pronounced.
Bidirectional text is text containing both right-to-left and left-to-right directional runs and it is now supported in Hyperion Performance Suite. Bidirectional text support is available on all platforms, in the user interface and in all exported documents:

- Export to HTML
- Office HTML
- Export to PDF.

Bidirectional text is rendered equally in all possible output modes, but with the following restrictions:

- Ligatures are not placed between Arabic letters if the text is exported to a PDF file. A ligature is a sequence of characters displayed as a single unit replacing characters that occur next to each other when common components are shared.
- In Office HTML Export, the bidirectional processing is performed by Microsoft Excel after the exported document is opened. Due to this behavior, text rendering may differ from what is presented in the Designer or Hyperion System 9 BI + Workspace.
- Release 8.5 does not change the system-defined bidirectional behavior of the following user interface controls: edit, drop-down, and list-box. This is true both for clients (Designer and Intelligence Client) and the Hyperion System 9 BI + Workspace.

Understanding Bidirectional Text

Understanding Bidirectional Text Each language has its writing system, or script. Script includes the set of characters which are used for writing. Most languages can be viewed in a left-to-right (LTR) direction of the text, which means that writing begins from left-hand side of page, and concludes at the right-hand side. However there are scripts which have a right-to-left (RTL) direction, for example, Arabic and Hebrew.

Since both RTL and LTR directional text can be displayed within the same paragraph, these paragraphs are referred to as “bidirectional.” For example in the Hebrew text below, the text is written in the RTL direction, and numbers are LTR directional.
Complex structures occur when a paragraph of one direction has a citation of different directional text. This citation can contain other citations, which can have original text direction and so on. Consequently text of different directions can be enclosed several times.

The part of the text which has the same direction is called a run.

Bidirectional text can be ordered in visual or logical order. When characters are ordered in the same way in which they are displayed - from left to right, it is a visual order. Entering or reading bidirectional text is done in logical order, which is the sequencing of characters ordered on the way in which characters should be read. Some text scripts require that characters be combined into one ligature when output.

**Bidirectional Text Configuration Settings**

Bidirectional text properties can be set in the application configuration file. On UNIX, processing of bidirectional text is enabled by default. On Windows, bidirectional text support is performed when complex script support is enabled in the system. Additionally, bidirectional support is disabled by using the setting:

```
[Regional Settings]
BidiEnabled=0
```

All paragraphs are left-to-right directional by default, but this setting can be modified by using the setting:

```
[Regional Settings]
TextDirection=LTR|RTL|Context
```

All paragraphs have a left to right reading order if the setting is LTR, or a right to left reading order if the setting is RTL. Context refers to the paragraph direction based on the first letter of paragraph.
Index

Essbase
  applying measure filters, 171
  attribute dimensions, 158
  command language, 23
  display options, 160
  options, 66
  query language, 23
  quitting, 65
  substitution variables, 173

Symbols
% Increase function, 190
% of Category function, 300
% of Column function, 189
% of Grand function, 190, 240
% of Row function, 190
(), 311
.bqtools.ini, 81
.css, 84
.csv, 43, 48
.gif, 84
.htm, 84
.jpg, 85
.sql, 63
.txt, 43, 48
.xls, 48
=, 314
\<, 314
\<=, 314
\<>, 314
\>, 314
\>=, 314

Numbers
2-D bar charts, 220
2-D bar charts, using to analyze data, 220
3-D bar charts, 222

3-D View, 222

A
Abs function, 336
Add Computed Item command, 125, 145, 155, 207, 260, 307
Add Cume command, 207
Add Date Group command, 145, 155
Add Fact/Measure command, 180
Add Filter(s) command, 125
Add Grouping Column command, 145, 155
Add Request Item(s) command, 125
Add Selected Items command, 207
Add Side Label command, 180
Add Sort(s) command, 125
Add Top Label command, 180
Add Totals command, 207
adding
  computed items, 329
  footers, 71
  headers, 71
  sections, 69
adding computed items, 150, 193, 240, 297
adding cumulative totals, 188
adding date groups, 152
adding grouping columns, 151
adding lines to labels, 216
adding other report elements, 287
adding report groups, 283
AddMonths function, 335
Administer Repository command, 83
Aggregate Shared Members command, 145
aggregation, 24, 107
analyzing data based on day of week data, 356
AND, 311
Append Query command, 125
applying data functions to tables, 153
area charts, 224, 225
ascending sorts, 155, 260, 307
Asciifunction, 341
Atn function, 337
Auto generate results when processing OLAP Query, 77
Auto Logon, 74
Auto Refresh Query, 77
AutoAdd Columns command, 145
Automatic data type, 325
automatically adding columns (Results section), 140
Average function, 153, 189, 239, 299
Avg function, 361
Avgfunction, 339
AvgNonNull function, 363
AvgNonNullfunction, 339
axes, 210
axis grid lines, showing, 245

B
bar, 211
bar charts
  2-D, 220
  3-D, 222
    stacked, 224
      three-dimensional, 222
    two-dimensional, 220
Baselines, 234
BEGIN WITH, 314
BETWEEN, 314
bqtbls5.ini, 105
bqtools.ini, 81
Break Total command, 145, 155
break total data functions, 153
break totals, 135, 154
browser restrictions and limitations, 62
Byte data type, 325

C
cascading style sheets, 84
Ceil function, 337
changing chart color schemes and fill patterns, 254
changing chart data labels, 255
changing chart legends, 247
changing color of chart elements, lines, and text, 259
changing column names, 281
changing label nesting levels, 186
chart angle and elevation, changing, 250
Chart Colors, 256
chart elements, working with, 215
Chart menu commands, 260
chart properties, changing, 253
Chart section, 210
  applying sorts in, 241
  computed items, 240
  Outliner, 213
Chart This Pivot command, 184
charting basics, 210
charts, 31. See also Data Analysis and Reporting with Interactive Reporting
  adding computed items, 240
  adding lines to labels, 216
  area, 225
  building, 213
  combination, 228
  creating, 213
  creating from pivot tables, 184
  Data Layout, 213
  inserting text, 246
  line charts, 226
  multidimensional, 221
  pie charts, 214
  pivoting, 242
  ribbon charts, 225
  sorting, 31
  stacked bar, 224
  terminology, 210
  three-dimensional, 222
  three-dimensional bar, 222
  two-dimensional, 214
  two-dimensional bar, 220
chr function, 339
clustered bar charts
  understanding, 223
ColMax function, 364
ColMaxfunction, 339
ColMin function, 365
ColMinfunction, 339
color, adding, 30
color, changing in charts, 259
Column commands
  Add Grouping, 155
  Hide, 155, 260, 308
Modify, 155
Unhide, 155, 260, 308
Column Name function, 299
column names
  changing, 281
  hiding, 282
column totals, 153
columns, 27, 141, 154, 282, 294
combination charts, 228
comma-delimited files, 43, 48
commands
  Chart menu, 260
  formatting, 141
  Pivot menu, 207, 366
  Report menu, 307
  Results menu, 145
  Table menu, 155
  Tools menu, 83
comparing related values, 238
complex sorting
  in OLAPQuery section, 404
  in Query, Results, and Table sections, 402
components, report, 281
Compress All Documents, 75
compressing documents, 37
computed fields, 289, 290
Computed Items, 323
computed items
  adding, 155, 260, 307, 329
  Chart section, 240
  compared to data functions, 328
  examples, 356
  in Interactive Reporting document sections, 324
  in OLAP queries, 166
  in Query section, 324
  in Results and report sections, 325
  Report section, 297
  Table section, 150
computing new data, 323
concatenating report element syntax, 279
Concatfunction, 341
Conditional Formatting, 87
Connection commands, 83
Connections Directory, 76
Connections Manager command, 83
CONTAINS, 314
contents, document, 21
Convert BMP to PNG, 74
correlated subqueries, 98
Cosfunction, 337
Coshfunction, 337
Count function, 153, 189, 239, 299
CountDistinctfunction, 339
Countfunction, 337, 339
CountNonNullfunction, 339
CountNullfunction, 339
Create New Documents Compressed, 75
creating
  charts, 213
  custom reports, 280
  pivot tables, 183
  pivot tables from charts, 242
  smart reports, 304
  three-dimensional bar charts, 222
creating documents, 65
CSS files, 84
CSV files, 43, 48
Cumefunction
  definition, 339
cumulative filters, 188
cumulative totals, adding, 188
custom function, 138
custom functions, 153
custom menu items, 80
custom totals, inserting, 135, 153
custom values, 313
Customize command, 83
Customize Filter command, 125
customizing chart patterns, colors, and labels, 254
customizing chart properties and labels, 253
customizing filter options, 318

D
data
  calculating, 28
  charting, 30
  exporting, 47
  filtering, 28
  hiding, 30
  importing, 43
  pivoting, 28
  refining in OLAPQuery, 164
Data Function command, 207, 260, 307
data functions
compared to computed items, 328
in charts, 239
in custom reports, 299
in OLAP queries, 177
in pivot tables, 189
Data Functions command, 125, 181
Data Layout
  Chart section, 213
  Pivot section, 184
  Table section, 147
data models, 22
data source connections, 21
data sources, working with, 41
data types, 129
data, computing, 323
data, pivoting, 184
database
  definition, 20
  displaying remarks, 106
  viewing tables, 26
Interactive Reporting database connection file, 21
Date & Time field, 290
Date & Time Now field, 290
Date command, 72
Date data type, 325
Date field, 290
Date Group command, Add, 155
date handling, 75
Date handling options, 75
Date Now field, 290
dates, breakout columns, 140, 152
day of week, returning, 355
DayOfMonth function, 335
Decode function, 334
Default Connection, 76
default formats, 72
deleting columns, 154
deleting sections, 71
descending sorts, 155
design guides, 293
Design Guides command, 307
dimensions, definition, 130
display differences, 306
display options
  Essbase, 160
displaying axis grid lines, 245
displaying rulers, 292
Interactive Reporting document
  analysis and reporting, 23
Interactive Reporting document (BQY), 65
Interactive Reporting document sections. See sections.
document sections. See sections.
Document Sort Order Language, 79
documents
  compressing, 37
  contents of, 21
  exporting as Web pages, 62
  password protecting, 38
  saving, 36
  starting, 25
Documents Directory, 76
Interactive Reporting documents, definition, 20
Download Database Totals, 77
Download To Results command, 181
Drill Anywhere command, 260
drill down, 29, 165
Drill Down command, 180, 207
drill up, 165
Drill Up command, 180, 207, 260
drilling into charts, 242
drilling through, multi-dimensional to relational, 178
Duplicate Section command, 70
duplicate values, suppressing, 282

E
  editing footers, 71
  editing headers, 71
  elements
    selecting pivot, 184
    working with chart, 215
  empty rows, suppressing, 161
  Encrypt Dashboard scripts for new document., 75
  ENDS WITH, 314
  EQUAL, 314
  Estimate Query Size command, 125
  Excel files, 48
  Expfunction, 337
  export
    documents as Web pages, 61
    HTML Wizard, 85
    query logs, 63
    script to text file, 64
    sections, 47
SQL, 63
Expression line, 278
expression syntax, 279

F
facts, 183
facts, definition, 130
features, sort line, 241
field definitions, number, 73
file locations, 76
File Name field, 290
files, exporting, 47
Filter command, 145, 155
filter dialog box, 311
filter line, 310
filter line syntax, 311
filter options, customizing, 318
filter, operators, 314
filtering data in a table, 148
filtering queries, 315
filtering results, 316
filters
applying
in OLAPQuery, 170
member selection, 171
cumulative, 188
measure
applying, 171
variable, 172
member variable, 172
removing, 149
tables, 148
variable, 171
in OLAP Query, 171
slicer, 173
filters, applying in OLAPQuery, 170
filters, cumulative, 188
focus on items, 243, 302
Focus On Items command, 207, 260, 308
font styles, changing, 305
fonts, changing, 305
fonts, default, 72
footers, 71
report, 284
footers, page in custom reports, 286
formats
default
fonts, 72
number, 73
specifying, 72
formatting commands, 141
formatting day of week data, 356
formatting report items, 304
function, custom, 138
functions
data, 177
in sorts, 242, 405
MDX, 167
weighted, 191
G
GIF files, 84
Grand Total command, 145, 155
grand total data functions, 153
grand totals, 136
grand totals, inserting, 135, 153
graphic elements, 288
GREATER OR EQUAL, 314
GREATER THAN, 314
Grid command, 307
grid lines, 211
grids, 292
group fact syntax, 279
group headers, report, 283
Group Items command, 207, 260
group label syntax, 279
grouped column, 139
grouping columns, 155
guides, design, 293
H
headers, 71
page, in custom reports, 286
report, 284
report group, 283
Headers and Footers command, 308
Hide Column command, 145, 155, 260, 308
hide items, 244, 302
Hide Items command, 207, 260, 308
hiding and focusing on charted data, 243
hiding and focusing on reported data, 302
hiding column names, 282
hiding column totals, 282
hiding Request items, 93
hiding sections, 69
horizontal lines, 289
how Dimension Root Level (ODBO Level 0/Generation 1) Data, 77
HTML
export wizard, 85
restrictions and limitations, 62
HTML file formats, 49
comparison of, 49
HTML template directory, 76

I
import
data files, 43
SQL, 44
Increase function, 190
INI file, 81
Initcap function, 341
Insert Field command, 308
Insert Graphic command, 308
Insert Predefined Field command, 308
Insert Table command, 308
inserting
additional tables in a custom report, 282
limit values, 291
page breaks, 286
page headers and footers, 286
report headers and footers, 284
text, 246
text in charts, 246
inserting sections, 66
Instr function, 341
Integer command, 73
Integer data type, 325
IS NULL, 314
items, sort, 241

J
JavaScript expressions, 53
JPEG/JPG files, 85

K
Keep Together command, 305
Keep With Next command, 305

L
label reference, in sorts, 241, 404
labels, nested, 186
Las Saved field, 290
Last Printed field, 290
LastDay function, 335
layout aids, 277
Legacy Chart Color Palette, 257
legacy chart colors, 256
legend, 211
legends, positioning and resizing, 247
Length function, 341
LESS OR EQUAL, 314
LESS THAN, 314
level rules, OLAPQuery member and, 163
LIKE, 314
limit values, in reports, 291
line charts, 225, 226
lines, 289
local filters, Results section, 310
Local Results
limitations, 111
processing order, 112
Locked command, 289
logical operators, 314
Lower function, 341
Ltrim function, 342

M
manipulating chart data, 238
manipulating table data, 148
margins, in Report section, 294
master data models, 22
Max function, 337
Maximum function, 153, 189, 239, 299
MaxL (Multidimensional Access Language), 23
MDSQL (Multidimensional Query Language), 23
MDX (Multidimensional Expression Language), 23
MDX functions, 167
measure filters, applying, 171
measure variable filters, 172
measures, 165
Median function, 340
OLAPQuery member and, 163
members, drilling down, 165
menu commands
Chart, 260
Pivot, 207, 366
Report, 307
Results, 145
Table, 155
Tools, 83
menus and menu items, customizing, 80
meta topics, 22
metadata, definition, 20
Microsoft Office Excel, 51
Minfunction, 337
Minimum function, 153, 189, 240, 300
Modefunction, 340
Modfunction, 337
Modify Column command, 145, 155
Modify command, 207
Modify Computed Item command, 260
modifying pivot tables, 184
Month (for Add Date Groups) command, 73
MonthsBetween function, 335
moving between sections, 70
moving pivot table elements, 185
MovingMax, 393
multicolumned reports, 294
Multidimensional Access Language (MaxL), 23
multidimensional charts, 221
multidimensional databases
components, 157
definition, 20
querying, 157
supported, 20
Multidimensional Expression Language (MDX), 23
Multidimensional Query Language (MDSQL), 23
multiple data sources, in reports, 303

N
nested labels, 186
nested sorts, 403
NextDay function, 335
Nextfunction, 339
NM function, 335
Non-Null Average function, 190, 240, 300
Non-Null Count function, 190, 240, 300
NOT (with operator), 314
NOT EQUAL, 314
Null command, 73
Null Count function, 190, 240, 300
number field definitions, 73
number formats, changing, 305
number formats, default, 72
Number of Pages field, 290

O
object names, 289
ODBC, stored procedures and, 112
OLAP queries
adding totals to, 177
building, 158
OLAPQuery items, formatting, 178
OLAPQuery member and level rules, 163
OLAPQuery section, 23
accessing offline, 170
diagram, 157
showing as a chart, 178
working offline, 170
OLE DB for OLAP query language, 23
Open From Repository command, 41
operators
logical, 314
options
program, 74
Options command, 83
OR, 311
ovals, 289

P
Packed Real data type, 325
page breaks, inserting, 66, 286
Page Margins command, 307
Page Number field, 290
Page Setup command, 64
Page X of Y field, 290
pages, setting up in reports, 33
password protect designer mode, 38
password protect document, 38
Path Name field, 290
Percentilefunction, 340
Picture command, 308
Picture General Properties, 124
pictures, 289
pie charts
analyzing data, 214
rotating, 216
pie slices, positioning, 215
Pivot Data Layout, 184
pivot elements, selecting, 184
Pivot menu commands , 207, 366
Pivot Options, 329
Pivot Section, 183
pivot tables
calculating totals and subtotals, 187
charting, 184
how to use, 184
modifying, 184
selecting elements, 184
subtotals, 187, 188
working with pivot items, 184
pivoting data, 184
planes, 211
plot area, 210
positioning pie slices, 215
Powerfunction, 337
Preferred Repository Connection, 76
primary sort, 403
Print command, 64
Print Preview command, 64
Priorfunction, 339
process
in OLAPQuery, 169
in Query, 94
Process Query command, 83
Process Results To Table command, 125
program options, selecting, 74
Properties command, 261
publish, 143

Q
queries
adding topics to, 27
appending, 110
building OLAP, 158
cancelling, 96
estimating size, 106@example, 26
multiple, 91
processing, 94, 169
sizing, 106
troubleshooting, 106
Query Limit field, 289
query logs, exporting, 63
Query Options command, 125
Query Processing Order, 95
query properties, 113
Query section
data functions , 107
types of, 23
Query SQL field, 290
query, definition, 22

R
RankAscfunction, 340
Rankfunction
scalar, 340
Real command, 73
Real data type, 325
rectangles, 289
Reference dialog, 333
reference items, in sorts, 241, 404
Reference Line, 229
general properties, 231
label format properties, 233
line style properties, 232
Refresh Data command, 207, 261
relational databases
definition, 20
remarks, displaying, 106
Remove command, 145, 155
Remove Selected Items command, 207, 260, 308
Remove Total command, 180
removing columns, 282
removing Request items, 93
removing Request line items, 93
renaming sections, 70
reordering columns, 282
reordering Request items, 93
reordering Request line items, 93
Replacefunction, 342
report components, 281
report element syntax, concatenating, 279
report expression syntax , 279
report group headers, 283
report headers and footers, 284
Report menu commands, 307
Report Name field, 290
report pages, 292
Report section
computed fields, 290
computed items, 297
creating smart reports, 304
data functions, 299
diagram, 275
elements, 275
Expression syntax, 279
fields, 289
formatting items, 304
graphic elements, 289
limit values, 291
toolbar, 277
using multiple data sources, 303
Report Setup command, 308
reporting, 23
reports
adding computed items, 297
creating, 33
data functions, 299
designing, 32
inserting page breaks, 286
multicolumned, 294
setting up, 33
smart, 304
repository, saving documents to, 42
Request items, hiding, 93
Reset Print Properties, 74
Resource Manager, 121
Restore Name command, 207, 261
Result Limit field, 290
results
automatically adding columns, 140
enhancing, 132
exporting, 143
filtering, 133
saving, 37
viewing, 27
working with, 141
Results menu commands, 145
Results section
automatically creating, 170
data functions, 135
diagram, 129
Retrieve Dimensions command, 180
returning day of week, 355
ribbon charts, 225
rotating and elevating charts, 250
rotating pie charts, 216
round rectangles, 289
Roundfunction, 338
rows, selecting, 141, 154
Rtrimfunction, 342
Rulers command, 307
Save Connection command, 83
Save To Repository command, 42
saving documents, 36
scalar functions, 334
examples, 360
scripts, running from menu items, 80
Section Boundaries command, 307
sections
adding, 69
deleting, 71
duplicating, 69, 70
exporting, 47
hiding, 69
inserting, 66
moving between, 70
renaming, 70
viewing, 69
Select A Formatting Locale command, 72
selecting chart elements, 245
selecting columns, 282
selecting columns and rows, 154
selecting pivot table elements, 185
server filters in the Query section, 310
server versus local filter processing, 310
setting compound filters, 316
setting simple filters, 315
setting up
page columns, 294
reports, 293
setting variable filters, 317
Show All Items command, 207, 260, 308
Show As Chart command, 181
Show Hidden Items command, 207, 260, 308
Show Remarks command, 125
show values, 313
showing column names, 282
showing column totals, 282
showing pie percentages, 215
showing positive and negative values, 215
Signfunction, 338
simple sorts, 401
Sinfunction, 338
Sinhfunction, 338
slice, 211
slicers
  setting, 164
  variable filters, 173
smart reports, 304
smart reports, creating, 33
Sort Ascending command, 145, 155, 260, 307
Sort command, 207, 307
Sort Descending command, 145, 155
sort items, 241, 404
sort line features, 241, 404
sort lines, 402
sort order, 403
sorting
  chart items, 241
  charts, 186
  pivot tables, 186
  report items, 296
  reports, 296
  tables, 149
sorting data, 401
sorting, in OLAPQuery section, 404
sorting, in Query, Results, and Table sections, 402
sorts, nested, 403
specifying
  page margins, 294
  page size, 293
SQL
  definition, 22
  exporting, 63
  importing, 44
SQL files, 63
Sqrtfunction, 338
stacked bar charts, 224
staging data in tables, 148
Standard Chart Color Palette, 258
standard query languages, 22
StdDevfunction, 340
StdDevpfunction, 340
stored procedures, 112
  opening, 113
  processing, 113
Stored Procedures command, 125
String data type, 325
Structured Query Language (SQL), 22
subqueries, building, 96
Substrfunction, 342
subtotals, 187
subtotals, calculating, 29
Sum function, 153, 189, 239, 299, 381
Sumfunction, 339
suppressing duplicate values, 282
surface values, 326, 328
surface values, in data functions, 191
syntax
  concatenating, 279
  expression, 279
syntax, concatenating, 279
Sysdate function, 335
T
tab-delimited files, 43, 48
table column formatting options, 281
Table Data Layout, 147
table dimension syntax, 279
table fact syntax, 279
Table menu commands, 155
Table section
  computed items, 150
data functions, 153
diagram, 147
tables
  adding computed items, 150
  filters, 148
in bqtbls5.inifile, 105
modifying pivot, 184
pivot, 184, 187, 188
working with, 154
tables as a data staging area, 148
Tanfunction, 338
Tanhfunction, 338
Teradata Functions, 344
  Conditional, 344
Teradata Standard Functions
  Date, 344
  Miscellaneous, 345
  Numeric, 345
  OLAP, 346
  String, 345
  System, 346
  Trigonometric, 346
text in charts, inserting, 246
text labels, 289

text, formatting, 66

text, wrapping, 282

three-dimensional bar charts, 222

Time Aware Axis, 226

Time command, 72

Time datatype, 325

Time field, 290

Time Now field, 290

Timestamp command, 72

TimeStamp data type, 325

ToCha function, 336

ToDate function, 336

ToMonth function, 336

toolbar, in the Report section, 277

Tools menu commands, 83

topics, adding to query, 27

ToQtr function, 336

Total commands, 155

totals

adding to OLAPQuery, 177

break, 154

calculating, 29

calculating pivot totals, 187

cumulative, 188, 189

inserting break totals, 154

inserting column totals, 153

inserting in Results section, 135

inserting in Tables section

break, 153

custom, 153

grand, 153

surface, 189

surface, 190

ToYear function, 336

Translate function, 342

Trend Functions, 385

Trend Line

line format properties, 237

line general properties, 236

line style properties, 237

Trend Lines, 235

True Computed Item Totals, 190

Truncfunction, 338

two-dimensional bar charts, 220

TXT files, 43, 48

U

underlying values, 326, 328

Ungroup Items command, 207, 260

Unhide Column command, 145, 155, 260, 308

Upperfunction, 342

using functions, 333

using surface values, 328

V

value reference, in sorts, 242, 405

values, 210, 238, 282

Var limit indicator, 311

Varfunction, 341

Variable Filter command, 125

variable filters, in OLAPQuery, 171

Varpfunction, 341

VBA functions, 59

vertical lines, 289

views, changing, 66

Visible command, 289

W

Web browser restrictions and limitations, 62

weighted functions, 191

wrapping text, 282

X

XLS files, 48