HYPERION® INTERACTIVE REPORTING

RELEASE 11.1.2

USER'S GUIDE

Interactive Reporting User's Guide, 11.1.2

Contents

1

Introducing Interactive Reporting

Overview

Oracle's 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's 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.

2 Managing Interactive Reporting Data

Before you start developing pivots, charts or reports themselves, consider what you are trying to accomplish. Are you trying to provide insight into what is happening within your organization? In all cases, you want your users to come to correct conclusions about what is happening so they can make quality decisions that have positive outcomes. In evaluating how to prepare data, note the following:

Preparing the Data

To produce the right charts and pivots, you need the right data to start with. Unfortunately, data sources in the real world don't always provide the data in the format it is needed. You may need to manipulate the data in some way. This manipulation can be performed in Interactive Reportingitself, but at times may best be performed in the database itself. Other times, you may have several different ways of grabbing the right data and need to analyze the costs and benefits of each approach.

Data Requirements

To produce the right charts and pivots, you need the right data to start with. Unfortunately, data sources in the real world don't always provide the data in the format it is needed. You may need to manipulate the data in some way. This manipulation can be performed in Interactive Reportingitself, but at times may best be performed in the database itself. Other times, you may have several different ways of grabbing the right data and need to analyze the costs and benefits of each approach.

One highly overlooked area is the power to manipulate the local data cache that resides in the Results section. Many difficult analysis tasks are eased when you take the raw data and craft it into a more usable form in the Results or Table section.

Creating Reusable Components

You may find yourself creating the same computed items over and over in your pivots, charts and reports. To prevent this, try creating them once in the Results or Table section. They are then available throughout the entire Interactive Reporting document file. You can do the same thing in the Query section as well. Creating computed items in the Query section adds the ability to use any of the functions available from your database.

Cross-Row Comparisons

The set orientation of SQL makes it difficult to build logic that compares information in one row with another. The Next() and Prior() functions in Interactive Reportingprovide the ability to create complex logic that depends on comparing data between rows. Combine a computed item using these functions with local sorting of the Results section to create controlled comparisons.

Grouping Data to Provide Ad-Hoc Organization

Your database may not be organized in the way you want to analyze it. You may have a large number of discreet values that you would like to break out into a set of ranges. Perhaps you only have a date field, but want to perform analysis based on years or quarters. Or perhaps you have years and quarters, but they are calendar based and you want to perform a fiscal year based analysis. All of these data organization problems can be addressed by using combinations of Computed Items, Grouping Columns and Date Groupings in the Results section.

Sometimes what you get out of your database is not in the most usable form. Data entry can be inconsistent. The data needs to be transformed into a consistent representation of values. You may need to split out what is stored as a single measure into multiple "dimensionalized facts" for the analysis you want to perform. Or you may want to transform a dimension into a measure. Or the values in a dimension may be too long to conveniently fit in a chart or pivot you plan to build. Instead, you'd like to use some kind of short-hand value computed from the actual ones. Computed Items using If-Then-Else logic can address these issues.

For example, suppose you want to create a Pivot with a count of the number of transactions for each item. You could drop the Customer_ID field into the data items area in the Pivot data layout, right-click on the column and change the data function to be Count. Additionally, you want to add the dollar amounts and then expand the use of the Customer_ID count to calculate the average per transaction. In building the formula for the average, the Pivot reverts to the default data function of summing the Customer_ID as if it were a number to be added instead of an ID to be counted. This does not yield the correct average. Back up a step to the Results section instead. Use a computed item to "translate" the Customer_ID into a true count. Simply define the formula as a 1. Then you can use this numeric field directly in the pivot to compute the correct number of transactions and as part of the formula to yield the correct average per transaction.

Consolidation

The proper use of Computed Items in the Results or Table section depends on understanding how you want to consolidate your information. If the consolidation is based on the "raw" data, use the power of the Results/Table section. Sometimes, however, you want to compare one consolidation with another. In these cases, you have several ways to accomplish this. You can use Computed Items that are created at the Chart or Pivot section itself if the comparison is to be used only in that chart or pivot. To create globally useful consolidation comparisons, start in the Results or Table section with Computed Items that summarize at the relevant group levels, then add another Computed Item to calculate the actual comparison. When you add this comparison item to a chart or pivot, change the data function from Sum to something appropriate to eliminate multiple counting

Drilling

If you want to use certain drilling options, they must be predefined in the data model itself. The ability to use pre-defined drill paths, drill-to-detail and incremental drilling all depend on the data model used by the query. They are not controlled in the individual chart or pivot. The ability to turn off drilling altogether is also found in the data model definition.

Sectional Data

Before building your reports, you need to determine the data operations in three section – the Query section, the Results section and (occasionally) one or more Table sections. Each type of section is appropriate for performing different types of data operation tasks.

Query Section

The Query section is where you create the basis for all drilling options. It is also the place where you should create your first level of aggregation of the data for summary analysis charts and pivots. Having the database perform much of the data aggregation improves performance greatly. Server-based computed items are also defined here, as is special data retrieval query logic.

Results Section

This is the section where you create your global items. Anything that you want to use in multiple sections should be created either here or in the Query section that feeds it. These global items fall into several categories:

- **Default Labels**—The column labels that appear in the Results section will be the default labels for all charts, pivots and reports built from this data. You can set these global default labels in either the Query or Results section.

- **Default Formats**—Like column labels, the display format used in the Results section will be the default formatting for all charts, pivots and reports built from this data. These global default formats can only be set in the Results section.

- **Global Computed Item**—Computed items that need to appear in more than one section should be defined here in the Results section.

- **Global Comparions**—Like global computed items, comparisons that need to apply across multiple sections should be defined once in the Results section.

- **Global Groupings**—Groupings that apply across multiple sections are just like other global items. They should be defined in the Results section.

- **Global Transformations**—Global data transformations are a special type of Computed Item. The same rules apply for global transformations.

Table Section

The Table section provides a way to built items that are less global than ones built within the Results section. That is, a Table section can be used to create semi-global items using the same techniques used in the Results section to create global items. Items created in a Table section only apply to those sections built from that Table section. In general, sections built using a Table section are isolated from certain changes that occur in the Results section that feeds the Table section.

The Table section is also useful for:

- **Data Staging**—The Table section can be useful in those cases where you need to stage the data for a particular use. For example, you may want to limit which items a user can drill to but still want to allow Drill Anywhere to work. You can build your chart or pivot based on a Table section that only contains those columns you want the user to access. You also might need to create certain complex calculations in stages. Table sections built from other table Sections can be the answer.

- **Top N / Bottom N Analysis**—This form of analysis, which allows users to easily select how many items to display, usually involves creating a limit. However, Interactive Reporting does not allow a limit to be created on a computed item based on an aggregate calculation. Rank and Sum are such aggregate calculations. When a Computed Item column is added to a Table section, it is converted to the actual value derived from the computation. Since it is now a simple numeric value, a limit can now be applied to the Rank or Sum value in the Table section.

- **Report Section Refresh Control**—To reduce the size of Interactive Reporting document files that contain multiple reports, the Report section is designed to work off the Results or Table sections. However, since they do not carry their own data cache, Report sections do not offer any Refresh Control options. You can create a version of this feature by using a Table section as the basis for your Report. You then create a dummy limit on the Table section to artificially start with zero rows in the display. When you want to refresh the Report section, simply set the dummy limit to "Ignore" (either manually or in JavaScript code).

Special Design Considerations

When you design your charts, pivots and reports, these particular considerations should be foremost in your mind. They will help guide your design decisions.

Charts

Design Considerations for Charts Available design choices for chart sections include:

- **Global versus Local Computed Items**—Consider the best place to create Computed Items before starting the Chart design. Remember that items computed locally in the Chart section only have available to them those fields that have been added to the data layout. Globally computed items can even include database server fields that are not included in the query itself (by using server based computed items).

- **Starting with "raw" versus starting with a Pivot**—Some charts may be easier to create by starting with a pivot and then selecting the "Chart this Pivot" menu item from the Insert menu.

- **2-D,. 3-D**—Weigh the different effects each chart type has on the interpretation of the data.

- **Direct use versus Embedded**—Charts destined for embedding should be designed with that purpose in mind. If you need two similar charts, one for embedding and one for direct use, create one chart, then duplicate it.

Pivots

Design choices for pivot sections include:

- **Global versus. Local Computed Items**—Consider the best place to create Computed Items before starting the Pivot design. Remember that items computed locally in the Pivot section only have available to them those fields that have been added to the data layout. Globally computed items can even include database server fields that are not included in the query itself (by using server based computed items).

- **Starting "raw" versus starting with a Chart**— Starting with another section can be an easy way to create pivots. In addition to making the creation of pivots easier, it also allows for the creation of multiple "views" on the same data. This facilitates a broader range of data interpretation possibilities.

- **Using base values versus. using Surface values**—By default, a pivot section reaches back to the Results or Tables section from which it was created for all computations. This can create what may look like strange values for certain computations. For example, if you add the same fact a second time, you can right-click on the duplicate column and change the data function from "Sum" (the default) to something else, like "Max". By default, the pivot will reach back to the original data rows and display the maximum single row value. This may not be what you wanted displayed. If you instead wanted to display the maximum sum as displayed on the pivot section itself, you will need to switch the pivot to "Use Surface Values" mode. Aggregate calculations will then look at the summarized data as it appears in the pivot section to perform the calculations.

- **Direct use versus Embedded use**—Pivots destined for embedding should be designed with that purpose in mind. If you need two similar pivots, one for embedding and one for direct use, create one pivot, then duplicate it.

Reports

Available design choices for report sections include:

- **Table versus Free-form reporting**—Different design techniques are used to create these two fundamentally different report styles.

- **Global versus. Local Computed Items** In addition to the considerations mentioned for chart and pivot sections, report sections differ in one respect. If you click on a data field in a report and look in the Expression bar, all items in the report section are short embedded JavaScript routines. This provides report designers a huge amount of power, but it means that computed items defined in the report section require a different syntax than globally computed items that reside in the Query, Results or Table sections. You should weigh the relative merits of simplicity versus power when deciding where to define computed items to be used in report sections.

- **Creating tables from scratch versus Embedding a Table section**—One easy way to create table-oriented reports with easily created computed items is to embed an existing Table section instead of designing a table object via the report section Outliner.

- **Single Column versus. Multiple Columns**—The report section supports creation of multi-column reports.

- **Headers & Footers for Reports, Pages, Groups and Sub-Groups**—Only headers for Groups and Sub-Groups are created automatically. You need to specify all other headers and footers if you want to use them.

- **Use the full power of the SmartReport feature** —The SmartReports feature can do a whole lot more than embed charts and pivots into a report. Report section features include:

  - Printing of multiple charts and pivots on one page

  - Creating context-sensitive charts and pivots

  - Alternate reporting styles

  - Alternate ways to create computations

- Use a Dashboard section as a "power report" A Dashboard section with embedded charts, pivots and tables can provide an interactive reporting environment where users can select various parameter values to easily change the data view in the "report".

Sectional Tips

Query Section

Creating Global Cascading Show Values Lists

You can provide users with a Show Values list of values that automatically filters the choices shown to only values applying to previously selected filter values. To create a cascading Show Values list, you build a data model and change the filter setting to show the minimum number of available values.

➤ To create global cascading Show Values lists:

1   Build a data model.

2   Add filter items to the Filter line.

    Filters can be selected from different tables and topics.

3   On the shortcut menu for each filter, select **Variable Filter**.

4   Select **Data Model** , then **Data Model Options**, and then **Filter**.

5   Select **Show minimum value list**.

    This step adds the WHERE clauses automatically based on the previous filter selections (even with fields from different tables) to the Show Values SQL to filter the value lists display.

    Note: If Row_Level Security rules have been defined, the Show Values list may be further restricted for web-based users.

Creating Chart and Pivot Labels with Special Characters

Because the Chart and Pivot sections restrict which characters can be used to rename an object, as a work-around, you can modify the field name in the Request line object of the Query section. This feature provides the ability to use non-text characters in chart and pivot label.

➤ To create chart and pivot labels with special characters:

1   Build a data model.

2   Add items to the Request line.

3   Select an item on the Request line and on the shortcut menu, select **Properties**.

    The Items Properties dialog box is displayed.

4   Edit the value in the **Name** field to include the desired label description.

    For example, you could change the definition "Sales_Figures_Amount_Sold" to "Amount Sold ($US).

Filtering by Sub-Totals

You can provide a query result filtered by a summary calculation rather than a simple row value.

➤ To provide a results sets filtered by summary calculations:

1 Build a data model.

2 Add items to the Request line and include the fact item from which to create the summary calculation.

3 Select the fact item on the Request line and on the shortcut menu, select **Data Functions**, then (Function).

4 For example, you could select the Sum data function to filter by the sub-total.

5 Drag fact item to which the data function was applied to the Filter line.

6 Apply the filter criteria like you would any other filter.

These steps create a SQL HAVING clause in the resulting query that are sent to the database. The query only returns rows for the groups in the other fields where the total for the group matches the filter criteria.

Creating Read-Only Computed Items

You can include standard computed items in a query that are read-only (cannot be modified). To do this, you build a query with one or more computed items pre-defined in the data model, and then promote the data model of the query to a read-only located Master Data Model.

➤ To create read-only computed items:

1 Build a regular data model.

2 Select each topic item and on the shortcut menu, select **Promote to Meta-Topic**.

Computed items in a data model can only be added to a meta topic.

When a topic has been promoted to a meta topic, it is renamed to "meta" + the topic name.

3 Rename each meta-topic by double clicking meta topic and changing the name in the Topic Name field on the Topic Properties dialog box.

4 Select the meta-topic to which to add the computed items, and then select **DataModel** , then **Add Meta Topic Item** to define the new computed item.

Meta topic computed items can be either server-based or locally computed. Server items are based on what can be computed by way of SQL, and are computed at the database server. Local items can use the full power of the Interactive Reporting JavaScript interpreter, but require all items in the computation to be included on the Request line. If a required field is not already present on the Request line, adding the local computed item to the Request line adds all fields automatically needed by the computation.

Note: To define a local meta-topic computed item, all columns referenced by the computed item's formula must be present in the same meta-topic as the computed item. You can drag any fields needed from other tables into the meta-topic to support multi-

table computed items. After which you may hide that field from the topic in which it normally resides.

5   Once all computed items have been defined, select **Data Model**, then **DataModel View**, and then **Meta** and then to set the display to show only meta-topics.

6   To make the items read-only, promote the data model of the query to a master data model by selecting **DataModel** , then **Promote to Master DataModel**.

Note: To keep the master data model locked for web use, deploy the Interactive Reporting document file (BQY) in "Query and Analyze (and not in "DataModel and Analyze" mode).

Date-Stamping a Query

You can provide date-stamp information, such as when the data was retrieved from the database, rather than when the report is viewed or printed.

➤   To date-stamp a query:

1   Build a data model and query.

2   Right-click on the Request line of the query, and on the shortcut menu, select **Add Computed Item**.

3   On the Modify Item dialog box, type the name in the **Name**. For example, type: As Of.

4   Select **Functions**.

The Functions dialog box is displayed.

5   Select **Data Functions** from the Function Categories column.

6   Select the desired date/time function for the database connection.

7   To use the "As Of" field, drag it into the desired Chart, Pivot or Report area.

Creating A "What Is Not" Query

You can build a query that tells you what did not come back from the database. This procedures adds a filter based on a "not in" sub-query Select statement.

➤   To create a "What is not" query:

1   Build a data model that includes one or more tables that would normally contain the absent information.

2   Add a filter on the primary key join filed by either double-clicking on the filed, or dragging it into the Filter line.

3   In the Filter dialog box, select **Custom SQL.**

4   Enter the SQL for a "not in" query that tells the database to fetch data from the primary table where the linking field does not appear in the subordinate table.

For example, you could type:

```
Products.Product_Id nof in (select product_id from sales_fact)
```

You might even add a filtering WHERE clause to the not in sub-select clause to constrain the results further.

Creating A Filter "What Is Not" Query

You can add a filter based on a visually defined "not-in" sub-query.

➤ To create a filter "What Is Not" query:

1 Build a data model that includes one or more tables that would normally contain the absent information.

Build your query based on fields from these tables.

2 Add tables to the data model for the testing condition.

Do not join these tables to the other tables in the data model.

3 Using the query portion of the data model, add a filter on a field that also appears in the test condition portion of the query.

For example, you could use a Store Key item from the Store Description table to the Filter line.

4 In the Filter dialog box, select **Not** to the left of the =Equals operator field.

5 Select **Advanced**.

6 Select **Create Subquery**.

A subquery section is created under the Query section, using the same data model as the query. This is the reason for adding the test condition tables to the data model of the query.

Using the example in Step 3, you could create the subquery to provide a list of Store Keys used to filter items which match the desired select criteria. For example, you could create a subquery that provides a list of Store Keys for stores that do have sales after a target date. The Not Equals operator of the main query's filter shows all stores not in the list.

7 Verify that the Request line of the subquery matches the line used for the filter of the main query.

For example drag the Store Key item from the Sales Figures table to the Request line of the subquery.

8 Add a target date as a variable filter

You might drag from the Date field from the Day Dimension table in the Filter line of the subquery. Then, change the filter operator ">Greater than". Enter a date and select OK

9 Right click on the Date field in the Filter line of the subquery, and on the shortcut menu, select **Variable Filter**.

10 Select the Query section from the Section catalog, and select **Process**.

You are prompted to select a target date.

Displaying the Query Status

You can let the user about the running status of a query by adding a graphic "processing" dialog to a Dashboard section and creating an Alert in an OnPostProcess event.

➤ To show the query status graphically:

1 In Design mode, temporarily create a command button with the following script:

```
Application.Alert("Your query is processing. Please stand by . . . ")
```

2 Switch to Run mode.

3 Select the command button and click **Alt + PrintScreen** to copy the dialog box to the Windows clipboard.

4 Switch back to Design mode and delete the command button.

5 Select **Ctrl + V** to paste the bitmap of the dialog box into your Dashboard section.

6 Name the bitmap **picQSP** for the purposes of this procedure.

7 Create a command button to process the query with the following script:

```
picQSP.Visible = true
ActiveDocument.Sections["Query"].Process()
```

8 Add the following script to the **OnPostProcess** document script:

```
ActiveDocument.Sections["SampleDashboard"].Shapes["picQSP"].Visible = false
Application.Alert(*Your query has complete. \n","Query Status")
```

Controlling the Drill Path

You can show only certain fields in a drill path, especially when some fields are used to build computed items and should remain invisible to the end user. Additionally, you may not want to use a Table section since this would duplicate data. To do this, you add JavaScript code to the OnPostProcess and OnPreProcess document events to remove the undesired fields from the Results section after the query has processed and all computed items have been calculated.

➤ To control the drill path:

1 Build a data model and query.

2 To launch the Script Editor, select **File**, **Document Scripts**.

3 Select the **OnPostProcess** event.

4 To remove items from the Results section after the query has processed, enter script to remove these items. For example, you might enter:

```
// Eliminate fields from the drill path
ActiveDocument.Sections["Results"].Columns["State"].Remove()
ActiveDocument.Sections["Results"].Columns["Date Ordered"].Remove()
```

5 Select the OnPreProcess event and add the script to put the items back into the Results section just before processing the query. Any dependent computed items are properly refreshed. Based on the example in Step 4, you might type:

```
// Add the "hidden" fields so computed items can be recalced
ActiveDocument.Sections["Results"].Columns.Add("Date Ordered")
ActiveDocument.Sections["Results"].Columns.Add("State")
```

Incremental Drilling

You can allow users to drill to further detail without needing to fetch all information into the original document beforehand. Using incremental drilling provides a fast and efficient way to provide drilling while avoiding the creation of huge Interactive Reporting document files. To use incremental drilling, you use a combination of a pre-defined drill path and the drill-to-detail information that is not currently contained in the query.

➤ To use incremental drilling:

1 Build a data model with a single meta-topic.

2 Set the data model view to show only the meta-topic by selecting **DataModel**, then **DataModel**, and then **Meta**.

3 Double-click the title bar of the meta-topic to display the Topic Properties dialog box.

4 Select the **Up** and **Down** buttons to set the drill order of topic items.

5 Enable **Set as Dimension**.

6 To remove a topic item from the drill path (such as a fact items and not dimensions), select a topic item and on the shortcut menu, select **Properties**.

7 Select **Set as Fact** and click **OK**

8 Select **DataModel**, then **DataModel Options**, and then **General**.

9 Select **Allow Drill to Detail** and click **OK**.

10 Create a top-level chart as usual.

When a user double-clicks on a bar in the chart, Interactive Reporting will prompt the user to add the missing dimension to the query, and then copy the query and chart definition to new sections to preserve the original query.

11 Select **yes** to add the missing dimension, and copy the query and chart definition to new sections.

Interactive Reporting copies the query and chart sections, but with a new dimension on the Request line and a new filter based on the bar that has been double-clicked.

12 Double-click on a target bar and Interactive Reporting continues to add new dimensions and filters, and then reprocesses the query automatically as need to perform the drilling.

Partial Importing of Excel Data

You can allow users to import only part of an Excel worksheet and to ignore other portions. To do this, you use a named ranged in Excel to provide a defined data region for a Query section.

➤ To import only a portion of Excel data:

1 In Excel, highlight the range of cells that you want to import into the Query section.

2   In the cell reference edit box (in the upper left of the worksheet), type a name for this cell range.

3   Save the file and exit Excel.

4   In the Windows Control Pane. select **Administrative Tools**, then .**Data Sources (OCBC)**.

5   Using the Microsoft Excel Driver, define a new System DSN.

6   Define the ODBC DSN as shown below using the Select Workbook button to point to your Excel file:

- Data Source Name—Enter **PartialImport**

- Version—Select the Excel version.

- Workbook—Select the workbook to point to the Excel file.

7   Create an Interactive Reporting database connection file named **Partial Import** that uses and ODBC/ODBC connection, and selects the "Partial import ODBC DSN.

8   Create a new Query, and select the Partial.Import.oce file.

The named range is displayed as a table, and can be used to build a query.

Creating Folder-Based Data Models

You can provide with a folder-based display of dimensions and facts to choose from in building a query.

➤   To create folder based data models:

1   Build a data model with a meta topic for each folder that you want to create.

2   Set the data model view to show only the meta-topic view by selecting **DataModel** , then **DataModel View**, and then **Meta**.

3   Drag each meta topic to the bottom of the screen.

Scroll down to gain more screen space and drag the meta topics down further so that they are displayed at the bottom of the screen.

4   Scroll to the data model design area at the top of the screen, hiding the meta topic displayed at the bottom of the screen.

5   Promote the data model to a master data model by selecting **Data Model** , then **Promote to Master Data Model**

This step locks your data model into the folder-based view.

Filter a Show Values List

You can provide users with what appears to be an open ability to filter a query, when in actuality, the list of values is predefined.

➤   To

1   Build a data model.

See Building a Data Model.

2 Promote the data model to a Master Data Model by selecting **Data Model.**, then **Promote to Master Data Model**.

3 In the mast Data Model section, add one or more filters to the data model by dragging a topic item to the **Filter** line.

Any query built using this master data model inherits the filter.

Note: To keep the master data model locked for web users, deploy the Interactive Reporting document file in "Query and Analyze" mode and not in "Data Model" mode.

Results and Tables

Performing A Union All Local Results

You can add the rows of two results sets without adding more columns (which is result of a simple join of local results). Traditionally to add rows, you perform a SQL operation. However, you can simulate a union without having a database. To do this, you add a dummy non-matching column and create an outer join of the results sets.

➤ To perform a union all local results:

1 Create and execute each query.

2 In each results section, add a computed item named **Join Here**.

The formula for the first query's "Join Here" computed item should be "a".

The formula for the second query's "Join Here" computed item should be "b".

3 Create a third query and join the two local results on the "Join Here" fields.

4 Add all fields from both local results from both local results to the R

3

Interactive Reporting: A Tutorial

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 the 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.
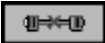
➤   To view all the database tables:

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 1   Connection Status Icons

Icon	Description
	Connected—You must be connected to a database to work in the Query section.
	Disconnected—You do not need to be connected to a database for many Interactive Reporting tasks.

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 **>=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.bqy, 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.

9 Drag **Product Line** to the y 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.

6 Drag **Territory** to the Report Group 1 field of the Groups data layout.

7 Drag **Country** to the Report Group 2 field of the Groups 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.

4

Interactive Reporting Basics

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 52

- "Opening Interactive Reporting Document Files" on page 52

- "Closing Interactive Reporting Document Files" on page 52

- "Saving Documents" on page 52

- "Specifying Save Options" on page 53

- "Save Query Results With Document" on page 53

- "Compressing Interactive Reporting Document Files" on page 53

- "Saving Interactive Reporting Documents to the Reporting and Analysis Repository in Interactive Reporting Web Client" on page 55

- "Password Protecting Interactive Reporting Document Files" on page 54

- "Password Protect Designer Mode" on page 54

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 23.

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 Enterprise Performance Management Workspace, Fusion Edition do not execute the document shutdown scripts when saving the Interactive Reporting document file. These scripts only execute when a 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:

- Specifying 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 90.)

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's 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's 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 57
- "Save To Repository in Interactive Reporting Studio" on page 58
- "Import Data File" on page 59
- "Import SQL" on page 60

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**.

10 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 tot 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 forma.

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

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.

➤ 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** 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 71.

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.

2   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.

3   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.

4   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.

4 **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)
- 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.

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 HMTL 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 e-mail 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 2   Comparison of HTML File Formats

Section	Office HTML Export to Office Excel	Standard HTML Export to Excel
Results/ Table	Formatting is preserved. Formulas are supported. (See Formula Mappings).	Formatting is not preserved. Formulas are not supported.
Pivot	Export preserves formatting. Cell merging is supported.	Export does preserve formatting. Export does not support formulas. Cell margins are not supported.
Chart	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. **Tip:** 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.	The Chart data is written to a simple table.
Report	The Report data is written to a simple table. 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.	The Report data is written to a simple table.
Dashboard	Not supported except as an image in the export feature.	Not supported
EPM Workspace	Export to Office 2000 is not supported by way of Toolbar or Menu	Not supported by way of Toolbar or Menu

Exporting a Section to Microsoft Office Excel Worksheets

Each exported section to Microsoft Office HTML creates one worksheet in the Excel workbook even when there are multiple sections. Exported sections of the same Interactive Reporting document do not reference each other in the Excel worksheet. For example if a Table section is

created from a Results section, both exported sections are displayed as independent worksheets in 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 Excel row maximum, all rows are exported to the source rows available, but only 65, 536 rows are processed for Excel to process any formulas. In this case, 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 71 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 3   JavaScript to Excel Mappings

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

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 4   Computed Items to Excel Function Comparison of HTML File Formats

Interactive Reporting Computed Item	Microsoft Excel Function	Implementation	Aggregation
Decode	H_Decode	VBA	
Nvl	Limited support: Nvl(column, expr).	Excel	
AddMonth	H_AddMonth	VBA	
DayOfMonth	DAY	Excel	
LastDay	H_LastDay	VBA	
MonthsBetween	H_MonthsBetween	VBA	
NextDay	H_NextDay	VBA	
Sysdate	NOW	Excel	
ToChar(value, format)	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.	Excel	
ToChar(column)	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.		
ToDate(x)	(DATEVALUE(x) + TIMEVALUE(x))	Excel	
ToMonth	H_ToMonth	VBA	
ToQtr	H_ToQtr	VBA	
ToYear	YEAR	Excel	
Abs	ABS	Excel	
Atn	ATAN	Excel	
Ceil	H_Ceil	VBA	
Cos	COS	Excel	
Cosh	COSH	Excel	
Exp	EXP	Excel	

Interactive Reporting Computed Item	Microsoft Excel Function	Implementation	Aggregation
Floor	H_Floor	Excel	
Ln	LN	Excel	
Log	LOG	Excel	
Log10	LOG10	Excel	
Max	MAX	Excel	
Min	MIN	Excel	
Mod	H_MOD	VBA	
Power	POWER	Excel	
Round	ROUND	Excel	
Sign	SIGN	Excel	
Sin	SIN	Excel	
Sinh	SINH	Excel	
Tan	TAN	Excel	
Tanh	TANH	Excel	
Trunc	TRUNC	Excel	
Avg	H_Avg	VBA	*
AvgNonNull	AVERAGE	Excel	*
Chr	H_Chr	VBA	
ColMax	H_ColMax	VBA	*
ColMin	H_ColMin	VBA	*
Count	H_Count	VBA	*
CountDistinct	H_CountDistinct 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.	VBA/Excel	*
CountNonNull	COUNTA	Excel	*
CountNull	H_CountNull	VBA	*
Cume	Computed Items with expressions that have exactly one call to Cume() and nothing else. Example: "Cume(Units)"	Excel/VBA	Aggregation and cross references in rows

Interactive Reporting Computed Item	Microsoft Excel Function	Implementation	Aggregation
Next	Function call is converted to appropriate cell reference. Limited support: Cannot be used in a Break or Grand Total expression.	Excel/VBA	Cross references in rows
Prior	Function call is converted to appropriate cell reference. Limited support: Cannot be used in a break or grand total expression.	Excel	Cross references in rows
Sum	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.	Excel/VBA	*
Median	MEDIAN	Excel	*
Mode	H_Mode Mode maps to Excel's H_Mode because Excel's MODE does not support complex ranges. Mode can only be used in break or grand totals. There may be performance considerations associated with this computed item because it uses heavy computations.	VBA	*
Percentile	H_Percentile	VBA	*
Rank(column)	H-Rank Rank maps to Excel's H_Rank function. Rank can only be used in break or grand total expressions. There may be performance considerations associated with this computed item because it uses heavy computations.	VBA	*
RankAsc(column)	H-RankAsc This function can only be used in break or grand totals expressions. There may be performance considerations associated with this computed item because it uses heavy computations.	VBA	*
StdDev	H-StdDev	VBAI	*
StdDevp	H_StdDevp	VBA	*
Var	Var	VBA	*
Varp	Varp	VBA	*
Ascii	CODE	Excel	
Concat	CONCATENATE	Excel	
Initcap	H_Initcap	VBA	
Instr	H_Instr	VBA	

Interactive Reporting Computed Item	Microsoft Excel Function	Implementation	Aggregation
Length	LEN	Excel	
Lower	LOWER	Excel	
Ltrim	H_Ltrim	VBA	
Replace	H_Replace	VBA	
Rtrim	H_Rtrim	VBA	
Substr	MID	Excel	
Translate	H_Translate	VBA	
Upper	UPPER	Excel	
MovingAvg	H_MovingAvg	VBA	Cross references in rows
MovingDiff	H_MovingDiff	VBA	Cross references in rows
MovingMax	H_MovingMax	VBA	Cross references in rows
MovingMed	H_MovingMed	VBA	Cross references in rows
MovingMin	H_MovingMin	VBA	Cross references in rows
MovingSum	H_MovingSum	VBA	Cross references in rows
All aggregate functions with two and three arguments: Ex. Sum(col, break_col, break_value)	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.	Excel/VBA	

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.

	A	B	C	D	E
1	Store	Item Name	Units	Computed	Computed2
2	Bayshore Electronics	1024 MB Drive	45	=Units*2	=SUM(Units_Agg)
3	Bayshore Electronics	EZ Fax Modem-I	30	60	21265
4	Bayshore Electronics	EZ Fax Modem-x	45	90	21265
5	Bayshore Electronics	One Button Mouse-	30	60	21265
6	Bayshore Electronics	The Quad	35	70	21265
7	Bayshore Electronics	One Button	30	60	21265
8	Bayshore Electronics	One Button Mouse-	40	80	21265
9	Bayshore Electronics	One Button Mouse-	35	70	21265
10	**Total Bayshore Electronics**		**475**	=SUM(Sr_13_2)	

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 document section names. Before exporting a section, consider the following limitations that apply to worksheet names:

- The name of section to be exported should not be longer than 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 90.

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 recognizes 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 77) 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 82 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 82
- "Displaying Numbers in Scientific Notation" on page 83

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:

Variation	Description
The number of decimal positions after the decimal separator.	For example, 0.00000E+000 displays six digits of precision.
#	Used after the decimal point to suppress zeros.
	For example, the value .000179 with the format 0.00000E-00 displays as 1.79000E-04. With the format 0.0####E-00, the same value displays as 1.79E-04.
E or e	Controls the case in which the exponent designator displays.
E0 E+0 E-0 (and the lower case equivalents)	If the plus sign is used, the exponential component always displays with a sign (plus or minus).
	If the minus sign or no sign is used, the sign displays only when negative.
E+0 to E+000	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.
	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.
Leading + or – sign	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.

> **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:

Button	Description
	Currency—Applies currency formatting to the selected numeric object(s). The currency formatting applied is the first currency type for the selected locale specified in the BQFORMAT.INI file. To display all the available currency types for the current local, click the arrow to the right of the Currency button.
	Note: The Currency button is ignored if the selected object is not of a numeric type (real or integer).

Button	Description
% ▾	**Percentage**—Applies percentage formatting to the selected numeric object(s). The percentage formatting applied is the first percentage format for the selected locale specified in the BQFORMAT.INI file. To display all the available percentage formats for the current local, click the arrow to the right of the Percentage button. **Note**: The Percentage button is ignored if the selected object is not of a numeric type (real or integer).
,	**Comma Formatting**—Toggles the presence of the thousands-separator character on or off. If on, comma formatting inserts the thousands-separator character into the current format string for the selected object. The thousands-separator character is inserted based on current locale settings. **Note**: Comma formatting does not affect numbers formatted with scientific notation.
.0 .00	**Decimal Increase**—Increases the number of displayed digits after the decimal point (or its locale-specific equivalent). For example, if the current format for the field specifies three decimal digits, such as $#.000, pressing Decimal Increase modifies the format to $#.0000. Pressing Decimal Increase when the number of decimal positions is zero adds the decimal separator character in addition to the one decimal digit. For example, if the format is ##%, pressing Decimal Increase changes the format to ##.0%. **Note**: The Decimal Increase button is ignored if the selected object is not of a numeric type (real or integer).
.00 .0	**Decimal Decrease**—Decreases the number of displayed digits after the decimal point (or its locale-specific equivalent). For example, if the current format for the field specifies three decimal digits, such as $#.000, pressing Decimal Decrease modifies the format to $#.00. If the number of decimal positions goes from one to zero when pressing Decimal Decrease, the decimal separator is no longer part of the format. Pressing Decimal Decrease when the number of decimal positions is already at zero results in no action. **Note**: The Decimal Decrease button is ignored if the selected object is not of a numeric type (real or integer).

Working with Document Sections

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

- "Understanding Document Sections" on page 84
- "Adding Sections" on page 85
- "Viewing Sections" on page 85
- "Moving Between Sections" on page 86
- "Duplicating Sections" on page 86
- "Renaming Sections" on page 86
- "Adding Headers and Footers to Sections" on page 87
- "Deleting Sections" on page 87

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 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

Field	Description
Select a Formatting Locale	Sets the locale or country associated with the default format to use. The locale determines the available number, date, and currency formats.
Date	Sets the default date format.
Timestamp	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.
Time	Sets the default time format.

Field	Description
Month (For "Add Date Groups")	Sets the default month format for the month used in tge Add Date Groups.
Null	Sets the default format for null values. Null values are empty values for which no data exists. Null values are not equal to zero.
Real	Sets the default format for real values.
Integer	Sets the default format for integer values.

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

Table 6   Number Field Descriptions

Option	Description	Examples
0	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.	Apply 0 to show 123 Apply 0.00 to show 123.45
#	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.	Apply #,##0 to show 1,234
( )	Formats with parenthesis options display negative values in parentheses. Otherwise, negative values display with a minus sign.	Apply (#,##0) to show (1,234)
;	A semicolon operates as a separator between two number formats. The semicolon separates a positive integer and a negative integer.	Apply #, ##0;(#,##0) to show 1, 234 or apply (1, 234) for a negative number
$ %	Adds the respective character to numeric values in the same position relative to the decimal point.	Apply $#,##0.00 to show $1,234.56 Apply 0% to show 3%
m d yy	Displays month, day, and year in respective positions for date-coded information.	Apply mm dd yy to show 05 07 99
- /	Adds the respective character to date-coded values in the same position relative to variables.	Apply mm/dd/yy to show 06/23/99
HH MM SS	Displays hour, minute, and second in respective positions for date-coded information.	Apply HH:MM:SS to show 17:45:10
AM PM	Adds the respective character to time-coded values in the same position relative to variables.	Apply HH:MM:SS AM to show 17:45:10 AM

➤ 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

General Option	Description
Auto Logon	Maintains a connection whenever you create a new document. If you are currently logged on, Interactive Reporting prompts you to use the current connection.
Reset Print Properties	Retains the print settings with each section of the document, instead of inheriting the current default print settings.
Convert BMP to PNG	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.

Table 8   Document Save Options and their Descriptions

Document Save Option	Description
Compress All Documents	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.
Create New Documents Compressed	Compresses new documents.
Encrypt Dashboard scripts for new document	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.

Table 9   Date Handling Options

Date Handling Options	Description
When A Two Digit Year Is Entered, Interpret As A Year Between	By default, if you enter a date and type only two digits for the year, Interactive Reporting handles the dates as follows: ● 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. ● 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. 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. **Tip:** Whenever possible, enter the year as four digits; that is, type 2006 instead of 06.

Table 10   Microsoft Office Export Limitation

Limitation Options	Descriptions
Export to Excel2 format	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 fomat (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).
Rows	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.

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

Results	Description
Download Database Totals	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. **Note:** If you open anInteractive 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.
Auto Generate Results When Processing OLAP Query	Creates a Results section for any future OLAPQuery section when that OLAPQuery section is first processed.

Table 13   OLAP Query Options

OLAP Query	Description
Use OLAPQuery section	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).
Show Dimension Root Level (ODBO Level 0/Essbase Generation 1) Data	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.

Table 14   Query Options

Query	Descriptions
Auto Refresh Query	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.

International Options for Interactive Reporting

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

➤ To select international language options:

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 15   International Tab Options

Option	Description
This is the language of menus and messages.	Select the language to use for Interactive Reporting menus and messages. In the Interactive Reporting Web Client, this setting is available only offline.
Default BQY sort order language	Select the sort order language for the Interactive Reporting document. Languages differ in how data is sorted. See your administrator before selecting a language.
Default OCE conversion language	Select the default language of the Interactive Reporting connection file when it is converted to Unicode format.

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, Fusion Edition

● **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
  - Smart View
- **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 "To\day 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

Command	Description	Keyboard Shortcut
Process Query	Processes the current query, all queries in the Interactive Reporting document file, or a customized selection of queries.	
Connection	Allows you to select, log on, log out, modify, or create an Interactive Reporting database connection file.	
Connections Manager	Opens the Connections Manager dialog box.	F11
Save Connection	Saves an Interactive Reporting database connection file with an Interactive Reporting document file.	
Administer Repository	Opens the Administer Repository dialog box, where you can modify object descriptions or groups.	
Launch Dashboard Studio	Launches Dashboard Studio, which is a wizard-driven dashboard development tool for Interactive Reporting and lets you rapidly build dashboards without writing any code.	
Launch Dashboard Architect	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.	
Options	Allows you to set default formats or program options.	
Customize	Opens the Customize dialog box where you can add customized menus or menu items.	
Change Database Password	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.	

Command	Description	Keyboard Shortcut
Change Web Client Version	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.	

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

Option	Description
Break HTML Page Every __ Row	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 "No Page Break" is selected, this option is disabled.
No Page Break	Exports the data to one HTML file.
Export Without Quotes	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.

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:

	Cust Num	Name	Invoice Num	Amount Received
1	100010	Eliot Richards	10	6,221.39
2	100011	Isaiah J Schwartz an|	11	26,143.27
3	100012	Harold Alexander Fir	12	593.7
4	100007	Harriet Bailey	7	1,386.14
5	100008	Clair Butterfield	8	3,812.79
6	100009	Quentin Fields	9	8,994.48
7	100004	Jerry's Junkyard Spe	2	80,980.65
8	100005	Kate's Out of Date Dr	3	724.71
9	100006	Sam Johnson	6	136.75
10	100001	Joe Smith and Comp	5	519.96
11	100002	Corks and Bottles, In|	4	1,398.6
12	100003	Harry's Landmark Di|	1	7,964.17
13	100003	Harry's Landmark Di|	1	8,629.11

Based on the section, Conditional Formatting behaves as follows:

Section	Conditional Formatting Behavior
Results/Tables	Conditional Formatting has the option to format both data cells and total cells.
Report	When a Report table is selected, the Item drop-down includes table columns. Due to the number of possible Report section fields, if a Report Field is chosen, the Item drop-down has a single "current item" label.
Pivot	The Item drop-down includes all row and column labels, and Fact items.
CubeQuery	The Item drop-down has a Data Cell option and it can be applied to data facts only.

➤ 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.**

Operator	Description
(blank)	Empty operator (used to mark an expression for deletion)
= Equal	Item value is exactly as in the expression.
<> Not Equal	Item value is not equal to any value in the expression.
< Less Than	Item value is less than that one in the expression.
> Greater Than	Item value is greater than that one in the expression.
>= Greater or Equal expression	Item value is not less than that one in the expression.
<= Less or Equal	Item value is not greater than that one in the expression.
Begins With	Value in the expression is a symbolic prefix of item value.
Contains	Value in the expression is a symbolic infix of item value.
Ends With	Value in the expression is a symbolic suffix of item value.
Like	Item value matches wildcard string in expression ("%" is used instead of asterisk in wildcards).
Between	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.
IsNull	Item value is null.

5

Querying

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

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).

- **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".

4   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**.

Related Topics

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

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 18   Derived Tables Example

State
CA
CA
CA
FL

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 19   Results of Derived Table

State	State_Count	State_Percentage
CA	3	0.75
FL	1	0.25

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".

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:

You can accomplish these tasks by using the commands on the Query menu.

Note: See 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: TheInteractive 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.

Request		
Request	Region	SUM(Amount Sales)

	Region	Amount Sales
1	Americas	$68,755,835
2	Asia Pacific	$29,556,011
3	Europe	$34,569,833

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.

Request			
Request	Fiscal Year	Region	SUM(Amount Sales)

	Fiscal Year	Region	Amount Sales
1	1999	Americas	$11,411,978
2	1999	Asia Pacific	$4,250,710
3	1999	Europe	$5,066,636
4	2000	Americas	$57,343,857
5	2000	Asia Pacific	$25,305,302
6	2000	Europe	$29,503,197

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.

Request
Request | Fiscal Year | Region | Product Line | AVG(Amount Sales)

	Fiscal Year	Region	Product Line	Amount Sales
1	1999	Americas	Books	$11,717
2	1999	Americas	Music	$6,073
3	1999	Americas	Videos	$8,007
4	1999	Asia Pacific	Books	$6,883
5	1999	Asia Pacific	Music	$4,755
6	1999	Asia Pacific	Videos	$6,294
7	1999	Europe	Books	$6,469
8	1999	Europe	Music	$4,051
9	1999	Europe	Videos	$5,542
10	2000	Americas	Books	$16,215
11	2000	Americas	Music	$7,334
12	2000	Americas	Videos	$11,285
13	2000	Asia Pacific	Books	$9,546
14	2000	Asia Pacific	Music	$6,342
15	2000	Asia Pacific	Videos	$7,688
16	2000	Europe	Books	$8,735
17	2000	Europe	Music	$5,657
18	2000	Europe	Videos	$6,982

➤ 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, SUM(Units) or 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:

Query Operator	Description
Union	All distinct rows selected by either query are retrieved. No duplicate rows are retrieved.
Union All	All rows selected by either query, including duplicate rows, are retrieved.
Intersection	All distinct rows selected by both queries are retrieved.
Except	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.")

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 (.CVS) 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 20   Supported Data Types

Data Type	Descriptions
Automatic	A data type is determined automatically given the data type of the reference items and the computations performed.
Blob	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 134.
Byte	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.
Date	Calendar date in server default format (typically mm/dd/yy).
Integer (16-bit)	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.
Integer (32-bit)	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.
Long Text	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.
Packed Real	Real numbers packed for use with EDA middleware. The results in the Interactive Reporting are the same as real numbers.
Real	Decimal numbers up to 5 positions right of the decimal.

Data Type	Descriptions
String	Text strings to a maximum length of 256 characters.
	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.
Time	Time in format set by user preference.
TimeStamp	Date/time combination in format set by user preference.

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).

● 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:

```
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 .

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 delete, all sections that refer to 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 dialogs.

Table 21   Picture General Properties

Field	Description
Type	Displays the file format of the picture.
Size	Displays the size of the picture.
Imported	Displays the date and time on which the picture was imported to the Resource Manage.
Data Source	Displays the file path of the disk where the picture is located.
Configurable options	Displays the picture percent scale (height and width), and picture effect (none, stretch, clip and title).

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

Command	Description	Keyboard Shortcut	Shortcut Menu
Process Results To Table	Allows you to create a table in the database to store your results set.		
Estimate Query Size	Queries the database to see how many records your query will retrieve.		
Show Remarks	Displays any remarks recorded about a topic or topic item.	**Ctrl+I**	*
Add Request Item(s)	Adds the selected topic item to the Request line.		*
Add Filter(s)	Allows you to create a filter for the selected topic item.	**Ctrl+L**	*
Add Sort(s)	Adds the selected topic item to the Sort line.		
Add Computed Item	Allows you to add a new data item derived from server-side calculations performed on an topic item.		*
Data Functions	Applies a prebuilt data function to the selected Request item.		*
Variable Filter	Designates the selected filter item as variable, which causes Interactive Reporting to prompt the user for the filter values when the query is processed.		*
Customize Filter	Allows you to control access to the features on the Filter dialog box.		*
Append Query	Allows you to combine two or more queries in one Results set.		
Stored Procedures	Loads a stored procedure and displays it as a query object in the Content pane.		
Query Options	Opens the Query Properties dialog box where you can specify options for rows returned, time filters, and so on.		

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.

5 Click **Yes**.

Option	Description
Master Datamodel	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.
Logged on Connection	Uses the current Interactive Reporting database connection file.
No connection	The new Interactive Reporting document file is not associated with anInteractive Reporting database connection file, allowing you to work offline.
Options	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.

➤ 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)" on page 487

➤ 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

5 Select an Interactive Reporting database connection file from the Recent Database Connection Files list.

6 If you don't see the desired Interactive Reporting database connection file, click **Browse** to display the Select Connection dialog box.

7 Navigate to the desired Interactive Reporting database connection file and click **Open**.

The Connection Password dialog box is displayed.

8 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:** TheInteractive 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 23 Process Results to Database Table

Option	Description
Table Name	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.
Create Table	Creates a new table in which Request items form columns.
Append to Table	Append Requests items as new columns in a table.
Grant **Access to Table**	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.
Tables Created by	Lists tables created by Interactive Reporting under your user account.

Option	Description
Delete Table	Deletes a selected table.

6

Results

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 24   Results Column Formatting

Option	Description
Remove Column	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.
Select Column	To select a column, click anywhere inside the column.

Option	Description
Size Column Width	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.
Modify Column	To modify a computed column or a grouped column, select the column and choose **Results**, then **Modify Column**.
Hide/Show a Column	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**.
Hide/Show Column Titles	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.
Size Titles	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**.
Suppress Duplicates	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.
Wrap Text	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.
Group Columns	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.
Sort Ascending	To sort column values in ascending order, click **Sort Ascending** on the Results menu, shortcut menu, or standard toolbar.
Sort Descending	To sort column values in descending order, click **Sort Descending** on the Results menu, shortcut menu, or standard toolbar.
Move Column	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.

Formatting Results Rows

Interactive Reportingoffers the following formatting options for Rows in the Results section:

Table 25   Results Rows Formatting

Option	Description
Hide/Show Row Numbers	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.
Size Rows	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.
Eliminate Duplicate Rows	To eliminate duplicate rows, choose **Query**, then **Query Options**, and enable the **Return Unique Rows** option in the dialog box.
	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

Option	Description
Number	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.
Alignment/ Justify	To modify the way an item is justified within a row or column, use one of the following options:
	● Select an item, choose **Format**, then **Justify**, and select the desired justification.
	● Select an item, click **Alignment** on the shortcut menu, and enter the desired changes in the Alignment Properties dialog box.
Font	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.
Borders and Background	To modify the border of a column, select the column and choose **Format**, then **Borders and Background**.
	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.
Grid lines	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.

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 149
- "Sorting Results Data" on page 150
- "Adding Computed Items to Results" on page 150
- "Applying Data Functions to Results" on page 151
- "Adding Columns Automatically" on page 156

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 392.**

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 388

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 15, "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 480,

"Sorting Data" on page 479

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 14, "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 27   Break Total and Grand Total Data Functions

Function	Description
Sum	Returns sum of underlying values.
Average	Returns average of underlying values.
Minimum	Returns lowest of underlying values.
Maximum	Returns highest of underlying values.
Count	Returns number of underlying values.
Other	Allows you to create a custom function using JavaScript.

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

Option	Description
At every break in	The column on which to break.
Break total function	The data function to apply. Available data functions include: ● **Sum**—Returns the sum of underlying values. ● **Average**—Returns the average of underlying values. ● **Minimum**—Returns the lowest of underlying values. ● **Maximum**—Returns the highest of underlying values. ● **Count**— Returns the number of underlying values. ● **Other**—Allows you to create a custom function using JavaScript. See Working with Custom Functions for more information.
Add break total to	The column or columns on which to apply the break total.

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

Option	Description
Grand total function	The data function to apply. Available data functions include: ● **Sum**—Returns the sum of underlying values. ● **Average**—Returns the average of underlying values. ● **Minimum**—Returns the lowest of underlying values. ● **Maximum**—Returns the highest of underlying values. ● **Count**—Returns the number of underlying values. ● **Other**—Allows you to create a custom function using JavaScript. See Working with Custom Functions for more information.
Add grand total to	The column or columns on which to apply the grand total.

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.

Modifing 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)*

Option	Description
Sum	Returns the total of a column of numbers.
Average	Returns the average (arithmetic mean) of values in a number column. (The Avg includes NULL values when calculating the arithmetic mean.)
Maximum	Returns the largest of items in a column.

Option	Description
Minimum	Returns the smallest of items in a column.
Count	Counts and returns the number of rows in a column.
Other	Applies a custom function.

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.

➤ 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 31   Grouped Column Options

Option	Description
Column Name	Names the new grouping column in the Results window.
New Groups	Creates a custom group to be displayed as a value in the new grouping column.
Groups	Select 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.
Options	Indicates how to represent unassigned values within the grouping column. Options include: ● 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 value the name originally assigned to it.

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 157
- "Deleting Columns" on page 157

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 `sysdate` 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

Save Query	Save Computed Columns (as Snapshot)	What Happens
Selected	Selected	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.
Selected	Not Selected	Results are saved with the document but computed columns are *not* saved as a snapshot. Computed columns are recalculated when the document is opened.
Not Selected	Selected	either Results nor computed columns are saved with the document.
Not Selected	Not Selected	Neither Results nor computed columns are saved with the document.

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 61. 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's Hyperion® SQR® Production Reporting, and Oracle Hyperion Financial Reporting, Fusion Edition 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's 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, Fusion Edition, Oracle's Hyperion® Web Analysis and Oracle's 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 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

Command	Description	Keyboard Shortcut	Shortcut Menu
Filter	Opens the Filter dialog box.	Ctrl+L	*
Sort Ascending	Sorts the selected column values in ascending order (alphabetical or numeric).		*
Sort Descending	Sorts the selected column values in descending order (alphabetical or numeric).		*
Add Computed Item	Opens the Insert Computed Item dialog box.		*
Add Grouping Column	Opens the Grouped Column dialog box. Use to merge dimension labels into new groupings and aggregate the associated data.		*
Add Date Groups	Separates date-type items into year, quarter, and month items.		
Modify Column	Use to modify a computed column or a group column.	Ctrl+M	
Remove	Removes the selected column (or data layout item).	Del	*
Break Total	Opens the Insert Break Total dialog box.		*
Grand Total	Opens the Insert Grand Total dialog box.		*
Hide Column	Hides the selected column from view.		*
Unhide Column	Opens the Unhide Column dialog box.		*
AutoAdd Columns	Automatically adds columns in the Content pane for all requested items. If not selected, no columns are displayed in the Content pane. Turn this option off to manually add columns for requested items.		
Aggregate Shared Members	If the "Aggregate Shared Members" option is enabled, the total is included for the initial shared member and for any parent members above it:		

7

Tables

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.

➤ To create a table based on the Results section data:

1 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 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 34   Break Total and Grand Total Data Functions

Function	Description
Sum	Returns sum of underlying values.
Average	Returns average of underlying values.
Minimum	Returns lowest of underlying values.
Maximum	Returns highest of underlying values.
Count	Returns number of underlying values.
Other	Enables you to create a custom function using JavaScript.

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 35   Table Menu Commands

Command	Description	Keyboard Shortcut	Shortcut Menu
Filter	Opens the Filter dialog box.	Ctrl+L	*
Sort Ascending	Sorts the selected column values in ascending order (alphabetical or numeric).		*
Sort Descending	Sorts the selected column values in descending order (alphabetical or numeric).		*
Add Computed Item	Opens the Insert Computed Item dialog box.		*
Add Grouping Column	Opens the Grouped Column dialog box. Use to merge dimension labels into new groupings and aggregate the associated data.		*
Add Date Groups	Separates date-type items into year, quarter, and month items.		
Modify Column	Use to modify a computed column or a group column.	Ctrl+M	
Remove	Removes the selected column (or data layout item).	Del	*
Break Total	Opens the Insert Break Total dialog box.		*
Grand Total	Opens the Insert Grand Total dialog box.		*
Hide Column	Hides the selected column from view.		*
Unhide Column	Opens the Unhide Column dialog box.		*

8 Querying Multidimensional Databases

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 175

● "Database-specific OLAPQuery Options" on page 176

➤ 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

Results	Description
Download Database Totals	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. **Note:** If you open anInteractive 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.
Auto Generate Results When Processing OLAP Query	Creates a Results section for any future OLAPQuery section when that OLAPQuery section is first processed.

Table 37   OLAP Query Options

OLAP Query	Description
Use OLAPQuery section	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).
Show Dimension Root Level (ODBO Level 0/Essbase Generation 1) Data	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.

Table 38   Query Options

Query	Descriptions
Auto Refresh Query	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.

Database-specific OLAPQuery Options

Database-specific OLAPQuery 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

The tab for Essbase databases is displayed as follows:

Table 39   Display Option for Essbase Database

Option	Description

Query Options	Defines query options in the following areas:
	
	- **Suppress Missing Rows**—Suppresses the retrieval of any missing rows where all cells are null.
	- **Suppress Zero Rows**—Suppresses the retrieval of any zero rows where all cells are null.
	- **Specify the number of decimal places to retrieve**—Sets the number of decimal places that the server will return.
	- **Enable Hybrid Analysis Value Retrieval**—This feature allow 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 <Essbase Integration Server product.
	
	    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 EssbaseReport Script keyword:
	
	    < HYBRIDANALYSISON>
	
	    Conversely, to disable Hybrid Analysis retrieval, the instruction keyword
	
	    <HYBRIDANALYSISOFF>
	
	    The applicable keyword used sent to the Essbase Server can be viewed on the Query Log.
	
	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."
	`Physical Member Name = 0199`
	`Default Alias Table Value = January11999`
	`Alias Table 1 Value    = Jan99`
	`Alias Table 2 Value = Fiscal Month 1`

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

Option	Description
Suppress Empty Rows	Suppresses the retrieval of any empty rows for which there is no measure data.

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 41 OLAPQuery Member and Level Rules

Data Layout item(s)	Component Type	Query returns:
Country (All)	Level	All countries
State (All)	Level	All states regardless of country
City (All)	Level	All cities regardless of state or country
Canada	Member	Canada only
CA	Member	California only
San Francisco	Member	San Francisco only
State (All), San Francisco	Level, Member	All States regardless of country and San Francisco

Data Layout item(s)	Component Type	Query returns:
Country (All), CA	Level, Member	All Countries and California (not just the USA)
Country (All), State (All), CA, NV, San Francisco	Level, Level, Member, Member, Member	All Countries, All States

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), 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 183.

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 14, "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 42   OLAPQuery Functions

Function	Type of Function	Description
% of Column	Interactive Reporting	Calculates the value of the specified measure as a percentage of the total for the column.
% of Row	Interactive Reporting	Calculates the value of the specified measure as a percentage of the total for the row.
% of Total	Interactive Reporting	Calculates the value of the specified measure as a percentage of the total for all rows and columns.
% Change	Interactive Reporting	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.

Function	Type of Function	Description
Absolute Change	Interactive Reporting	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.
Avg	MDX	Calculates the average of the selected measure evaluated over the specified dataset.
Correlation	MDX	Returns the correlation of a dataset against two measures.
Count	MDX	Calculates the number of members for the specified dimension in the report.
Covariance	MDX	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.
Linregpoint	MDX	Calculates the linear regression of a dataset and returns the value of "b" in the regression line y = ax + b.
Linregr2	MDX	Calculates the linear regression of a dataset and returns r2 (the coefficient of determination).
Linregslope	MDX	Calculates the linear regression of a dataset and returns the value of "a" in the regression line y = ax + b.
Linregvariance	MDX	Calculates the linear regression of a dataset and returns the variance that fits the regression line y = ax + b.
Max	MDX	Returns the maximum value of the selected measure evaluated over the specified dataset.
Median	MDX	Calculates the median value of the selected measure evaluated over the specified dataset.
Min	MDX	Returns the minimum value of the selected measure evaluated over the specified dataset.
Stdev	MDX	Calculates the standard deviation of the selected measure evaluated over the specified dataset.
Sum	MDX	Calculates the sum of the selected measure evaluated over the specified dataset.
Variance	MDX	Calculates the variance of the selected measure evaluated over the specified dataset.

➤ 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 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 Result**s (see).

➤ 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 190.

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 43   How Databases Access the Filter Dialog Box

Database	Access Points
Essbase	Side Labels pane, Top Labels pane, Measures pane, Slicer pane
OLE DB	Side Labels pane, Top Labels pane, Slicer pane

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 190.

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

Operator Type	Description
Select Members from DB	Creates a filter based on a member(s) retrieved from the database. (Supported for the Essbase, MSOLAP and SAP/BW databases).
Select by Measure	Creates a filter based on a measure that you specify.
Top N	Retrieves only the top N values where each top N value is at least the specified Index value.
Top N %	Retrieves only the top N % values where each top N % value is at least the specified Index value.
Top Sum	Retrieves the top N (the smallest number possible) values, such that their sum is at least the specified Index value.
Bottom N	Retrieves only the bottom N values where each bottom N value is at least the specified Index value.
Bottom N %	Retrieves the bottom N% where each bottom N % value is at least the specified Index value.
Bottom Sum	Retrieves the bottom N (the smallest number possible) values such that their sum is at least the specified Index value.
Data Operator	Description
= Equal	Retrieves only records where the filtered item equals the specified value(s).
< > Not Equal	Retrieves only records where the filtered item does not equal the specified value(s).

Operator Type	Description
< Less than	Retrieves only records where the filtered item is less than the specified value(s).
< = Less than or Equal to	Retrieves only records where the filtered item is less than the specified value(s).
> Greater than	Retrieves only records where the filtered item is greater than the specified value(s).
> = Greater than or Equal to	Retrieves only records where the filtered item equals, or is greater than the specified value(s).

Table 45   Essbase Operator Types and Data Operators

Operator Type/ Data Operator	Definition
Equal	Retrieves only records where the filtered item equals the specified value(s).
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 ("").
Data Operator	Description
= Equal	Retrieves only records where the filtered item equals the specified value(s).
< > Not Equal	Retrieves only records where the filtered item does not equal the specified value(s).
< Less than	Retrieves only records where the filtered item is less than the specified value(s).
< = Less than or Equal to	Retrieves only records where the filtered item is less than the specified value(s).
> Greater than	Retrieves only records where the filtered item is greater than the specified value(s).
> = Greater than or Equal to	Retrieves only records where the filtered item equals, or is greater than the specified value(s).

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:

- **Missing Rows** (*Essbase only*)—Suppresses the retrieval of any missing rows where all cells are null.

- **Zero Rows** (*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 176 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 [icon] 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 46   OLAPQuery Data Functions

Data Function	Description
Sum	Returns sum of all values. This is the default function in all report sections.
Average	Returns average of all values.
Count	Returns number of values.
Maximum	Returns highest value.
Minimum	Returns lowest value.
% of Column	Returns surface values as a percentage of their respective column item.

Data Function	Description
% of Row	Returns surface values as a percentage of their respective row item.
% of Grand	Returns surface values as a percentage of all like values in the report.

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 47   OLAPQuery Drill Through Options

Option	Description
Dimensions	The dimensions in the current OLAP query. Click a dimension to select it.
Selected Dimension	The dimension selected by the user. This is the dimension to use for mapping.
Specify Relational Query	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.
Relational Topics	The topics contained in the selected relational query. Click a topic to select it.
Selected Topic	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.
Map/UnMap Dimension	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.
Specify Fact Topic	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.

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 48   OLAPQuery Menu Commands

Command	Description	Keyboard Shortcut	Shortcut Menu
Retrieve Dimensions	Refreshes the dimension values in the Catalog pane.	**F9**	
Add Side Label	Adds the selected item to the Side Labels pane.		
Add Top Label	Adds the selected item to the Top Labels pane.		
Add Fact/Measure	Adds the selected item to the Measures pane.		*
Add Computed Item	Opens the Computed Items dialog box. (This feature is for MS OLAP only)		*
Drill Down	Allows you to progressively narrow your focus on a selected item.		*
Drill Up	Returns the original view of data.		*
Drill Through	Drills through from a multi-dimensional database to a relational database.		
Remove Total	Removes local totals from selected dimensions.	**Del**	*

Command	Description	Keyboard Shortcut	Shortcut Menu
Hide Items	Removes selected items from the OLAPQuery report.		*
Show Hidden Items	Retrieves hidden items from the selected row or column to the OLAPQuery report.		*
Show All Items	Retrieves all hidden items to the OLAPQuery report.		*
Download To Results	Downloads the OLAPQuery data set to an OLAPResults section for offline work.		
Show As Chart	Charts the OLAPQuery data set; automatically creates OLAPChart and OLAPResults sections.		
OLAP Query Options	Accesses the OLAPQuery Options dialog box, where you can set options for your OLAPQuery.		
Data Functions	Recalculates the surface values in the OLAPQuery data set.		*
Set Drill-Through	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.		

9

CubeQuery

About CubeQuery

The CubeQuery section 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.

About Essbase

Essbase is a multidimensional database application that enables you to analyze multiple aspects of your business continually in comparison to one another. The Essbase database has the following characteristics:

- Works with multidimensional data and roll-up hierarchies in dimensions.

- Retrieves information from other systems.

- Handles some level of summarized data, not transaction.

- Adapts to multiple reporting and analysis environments.

Additionally, the Essbase database uniquely blends an innovative technical design with an open, client-server architecture. Essbase can be used for a broad range of online analytical processing (OLAP) applications, including:

- Budgeting
- Forecasting and seasonal planning
- Financial consolidations and reporting
- Customer and product profitability analysis
- Price, volume, and mix analysis
- Executive information systems

About Multidimensional Databases

The Essbase multidimensional database stores and organizes data. It is optimized to handle applications that contain large amounts of numeric data and that are consolidation or computation-intensive. The database organizes data to reflect how users want to view the data.

Definition of Multidimensional

A multidimensional database is an extended form of a two-dimensional data array, such as a spreadsheet, generalized to encompass many dimensions. Dimensions comprise values called members, which are arranged in a hierarchical structure. A *dimension* is a perspective or view of a specific dataset. A system that supports simultaneous, alternate views of datasets is *multidimensional*. Dimensions are categories such as time, accounts, product lines, markets and so on. Each dimension contains additional categories that have various relationships. *Members* exist within a dimension. A dimension can contain an unlimited number of members.

Number	Name
1	Dimension
2	Member

Dimensions and Members

Dimensions and members are the metadata of the Essbase database (also known as the Essbase outline). The Essbase outline determines how data is stored in Essbase. Following are typical examples of dimensions and their members:

- Time—Hours, Days, Months, Quarters, Years

- Measures—Profit, Inventory, Ratios

- Product—Individual products

- Market—North, South, Central

- Scenario—Actual, Budget, Variance, Variance %

Non-aggregate Dimensions

Some dimensions are aggregates of their members. For example, 24 hours make a day, 7 days compose a week, 52 weeks form a year, and so on. In these cases, the Time dimension represents the sum of its components. However, some dimensional hierarchies are not the sums of their components. For example, the Measures dimension is commonly composed of business measures such as Sales, Profit, and Cost of Goods Sold. When the dimension label cannot represent the sum of contents, a protocol exists to use one member set in place of the dimension label. Essbase uses the first child in the dimension outline as the implied share instead of non-aggregate dimension labels.

Familial Relationships

To simplify the concept of hierarchical order, Essbase uses familial relationships.

- Parent—Direct ancestor of an element in a hierarchy; for example, Quarters are the parent of Months.

- Child—Direct descendant of any element in a hierarchy; for example, Months are children of Quarters.

- Descendants—Elements of greater detail; for example, Days are descendants of Weeks.

- Ancestors—Elements of more generality; for example, Weeks are ancestors to Days.

- Siblings—Members with the same parent

Number	Name
1	Ancestor
2	Parent
3	Child
4	Descendent
5	Siblings (not marked)

Generations and Levels

Hierarchical structure also can be described in terms of generations and levels. Dimension members on the same layer of the dimensional hierarchy are referred to collectively as generations or levels. The relationships are defined as follows:

- Generations—Counted down from the highest ancestor. The root member, or dimension name, is Generation 1. In the following figure, Market is Generation 1

- Levels—Counted from bottom to top: Level 0, Level 1, Level 2, and so on. In the following figure, days are at Level 0

- Leaf nodes—Lowest points in a hierarchy

Number	Name
1	Ancestor
2	Parent
3	Child
4	Descendent

Attributes

In addition to dimension member names, locations, and relationships, Essbase stores characteristics, called attributes, about members. For example, the Product dimension indicates that in Women's Apparel, Shirts and Blouses, there is a cotton T-shirt; an attribute indicates that the cotton T-shirt is red, cyan, lime, or pink. Essbase does not store data for attribute dimensions as part of the multidimensional database but instead dynamically calculates it upon request. Attribute dimensions are displayed in dimension hierarchies, in the same manner as other dimensions, despite their data being stored differently.

CubeQuery Catalog

The Catalog displays one or all dimensions in a tree model. The tree can be expanded to display additional members, but generations are not included in the hierarchy. For cubes with many members in a dimension, paging facilitates navigation. The number of members that can be displayed in the Catalog and Member Selection is defined in the Number of Members to display option in Query Options. The default is 50.

➤ To display the Catalog in single-dimension mode, select a dimension from the Dimension Selection.

To expand the tree to display additional members in the dimension hierarchy, next to the dimension, click +.

➤ To display the Catalog in all dimensions mode, click .

OLAPQuery and CubeQuery Data Layout Differences

OLAPQuery is the pre release 9.3 section used to query a multidimensional database. The data layout tool in OLAPQuery and CubeQuery have these differences:

OLAPQuery Data Layout	CubeQuery Data Layout
Slicer	Filters
Side	Rows
Top	Columns
Filters from multiple members may be selected and applied from a single hierarchy.	One filter member can be selected per dimension. Filters for multiple members within the same dimension are not supported.
Measures are treated as distinct dimensions and can be placed only in the Facts pane of the data layout.	Measures can be placed in the Row, Column, and Filter panes of the data layout.
Member selection in the data layout is unavailable.	Member selections in the Row, Column and Filter panes of the data layout can be edited in the Member Selection dialog box.
Shortcut menu enables access to the Filter dialog box and Remove option.	Shortcut menu enables access to Remove option, Member Selection dialog, Data Filter option (rows and columns), Variable option and Filter option

Searching Members from the Catalog

Member search is available for member names, aliases or both in one or more dimensions from the Catalog. If the Catalog shows members in single dimension mode, the Search is performed on a selected dimension. If all dimensions are displayed, Search is performed on all dimensions.

Note: Only the member names and aliases are displayed and not the fully qualified name (for example, [West].[Salem]) in the Member Selection, Catalog or query results.

➤ To search members from the Catalog:

1 Select to search by name, alias, or both.

2 Enter the text on which to search.

If you use wildcards in searches, only trailing wildcards are accepted (for example, Cola*), and not leading wildcards (for example, *-10).

The ?? (placement position) is a valid search criteria. The ? substitutes one occurrence of a character; and can be placed anywhere in the string.

3 Click .

The search results are displayed in the Catalog, and they can be added to the data layout.

➤ To close Search, click .

The Search results are cleared.

Building a CubeQuery Section

When you query against an Essbase cube, by default the CubeQuery section is created in the Interactive Reporting document. To use this section, you must have access to an Essbase 7.x or 9.x server data source. Interactive Reporting also requires an Essbase 9.3 Runtime or greater client installation.

➤ To create a CubeQuery section:

1 Select **Tools**, then **Options**, then **Program Options**, then **OLAP** and then clear **Use Old OLAP Query section**.

By default, new Essbase queries use the new section type (CubeQuery).

2 Select **Insert**, then **New Query**.

The Insert New Query dialog box opens.

3 Select an Interactive Reporting database connection file (.oce) associated with an Essbase 7.x or 9.x server used as a data source.

For information on connecting to Essbase, see Connecting to Essbase (CubeQuery only).

4 Enter your name in the **Host Name** field and password in the **Host Password** field and click **OK**.

Interactive Reporting creates a Query section in the Section pane. The Catalog shows the dimensions available to add to the query.

To toggle the Catalog between single and all dimension modes, click .

5 In the Catalog, select a member dimension and select **Query**, then **Add to Rows** or **Query**, then **Add to Columns**.

Note: Members from the same dimension cannot be split across columns, rows and filters.

6 In the Catalog, select a measure dimension and select **Query**, then **Add to Rows** or **Query**, then **Add to Columns**.

7 To filter a member, select a member in the Catalog and select **Query**, then **Add to Filters**.

8   Click **Process**.

Note: If missing values are not displayed, missing rows and columns may be suppressed. This option is enabled in Query options.

Note: When using Essbase metaread filters that restrict Essbase metadata/members for specific users or groups, a user may see member in the Interactive Reporting document file that do not have metaread access in Essbase. This results because Interactive Reporting document files save data and metadata in the physical file. When the query is processed, they may then receive an "unknown member" error because they may not have access to one or members in the query due to their metaread access. When using CubeQuery with metaread filter access, the following is recommended. First, dynamic member selections (for example. Children, Descendants, etc.) should be used for any dimensions that have metaread filters applied. Second, members that users might not have access to should not be individually selected. Third, do not process the query before saving and/or publishing the Interactive Reporting document file.

Member Selection

An Essbase database may contain hundreds or even thousands of members. Use the Member Selection Browse and Search tabs to refine query member criteria for a selected dimension. The Member Selection dialog box includes dynamic selection functions such as children, descendants, bottom, siblings, or subsets (UDA, Attribute Dimensions, Level and Generation).

The Available pane lists the member hierarchy for the selected dimension. An additional node for substitution variables lists substitution variables for all dimensions; however, if they are selected from another dimension, an error appears when adding the member to the Selected pane.

The Selected pane lists all member selections. Selections can be a single member or a dynamic selection function (for example, children).

Icon	Dynamic Function	Description
	Member	Currently selected member
	Children	Children of the selected member (one level below)
	Descendants	Descendants of the selected member
	Bottom	Descendants of the selected member on the lowest level of the hierarchy (Level 0)
	Siblings	Members on the same level with the same parent as the selected member
	Same Level	Dimension members on the same level as the selected member

Icon	Dynamic Function	Description
	Same Generation	Dimension members on the same generation as the selected member
	Select Next/Previous	Opens the Select Next/Previous dialog box so that a user can select a number of next or previous members at the same dimension level as the selected member
	Subset	Opens the Subset dialog box to further define the member criteria for selections such as level, generation, user defined attributes, and attribute dimensions
	Date Time Series	Opens the Date Time Series dialog so that a user can select the latest time period to be reported

Note: When a Shared Member is individually selected, Essbase treats it as the initial occurrence, and not the shared occurrence of the member. If selection functions such as "Siblings" and "Select Next/Previous" are used with shared members, the returned members reflect the original occurrence of the member. For example in the Sample Basic Product dimension, if the shared member "100-20" (under the Diet member) is selected with the Siblings selection type, the results are (100-10, 100-20 & 100-30) and not (100-20, 200-20, & 300-30). This occurs because the original occurrence of the member, under the parent "100" is used by Essbase

Browsing Members

Use the Member Selection Browse tab to navigate through and select individual members to include in the query.

➤ To browse and select members:

1 In the data layout, right-click a dimension member and select **Edit**.

   The Member Selection dialog box is displayed.

2 Select **Browse**.

3 Select a member and click **+** to the left of the member name to expand it.

4 To include an individual member in the query, select the member in the Available pane and click .

   The member is added to the Selected pane.

5 To include a dynamic function, on the shortcut menu, select the member in the Available pane and select the function from the submenu:

   ● Member

   ● Children

   ● Descendants

   ● Bottom

- Siblings

- Same Level

- Same Generation

- Select Next/Previous

- Subsets

- Date Time Series

- Disable or Enable (selected members only)

6   Click **OK**.

➤ To change the member function in the Selected pane:

1   **Select the member to modify.**

Use Shift + click to select multiple and consecutive items. Use Ctrl + click to select multiple, but nonconsecutive items.

2   **To include or exclude a member (for example, "Member+Children" versus "Children"), select the check box next to the member name.**

3   Click **OK**.

➤ To clear an item used in the query without removing it from the Selected pane:

1   **Select the item and choose Disable from the shortcut menu.**

2   Click **OK**.

➤ To remove a member or member element:

1   **Select the member in the Selected pane and click** .

Use Shift + click to select multiple and consecutive items. Use Ctrl + click to select multiple, but nonconsecutive items.

The member is moved from the Selected pane.

2   Click **OK**.

➤ To remove *all* members and member elements, click .

The Selected pane is depopulated.

➤ To toggle between including or excluding a member:

1   **Select a member in the Selected pane.**

2   **Select Disable to exclude a member, or Enable to include a member.**

➤ To preview a list of all members selected for a dimension, click ▣.

The Preview window is displayed.

Date Time Series

Date Time Series members (DTS) are predefined database members used in dynamic, to-date reporting, such as year-to-date or month-to-date values. DTS results are displayed as individual members, summing Level 0 Time dimension members in Essbase based on the current selection.

Use the Date Time Series dialog box to select the latest time period in to-date calculation. The calculated value of the Date Time Series is derived when you define the latest time period to be reported. For example, suppose you need sales data for the quarter up to the current month when the level 0 members of the Year dimension are the months of the year, and the current month is August. In this case, you select a quarter-to-date calculation, which shows sales data for the months of July and August.

Note: Because MDX is used to query Essbase, and MDX does not have an exact equivalent of the Date Time Series (DTS) functions, a combination of xTD (refers to Year-, Quarter-, Month— and Week-to-date periods) and Sum functions are used. As a result, there may be different data output than a traditional DTS for some members; for example, those that use "Time Balance" functionality and "Two Pass Calc" members.

➤ To make a Date Time Series selection:

1   In the data layout, double-click a Time dimension in the data layout.

The Member Selection dialog box opens.

2   In the Selected pane, right-click a Time dimension member, and select **Date Time Series**.

The Date Time Series dialog box opens.

3   Select the definition to use for the to-date calculation and click **OK**.

Substitution Variables

Substitution variables define global variables that represent Essbase-specific values. For example, CurMonth can be a substitution variable for displaying the latest time period. Rather than constantly updating queries when a value changes, you change only the value for the substitution variable on the server. Essbase creates substitution variables, but you can specify a substitution variable when selecting members.

➤ To add a substitution variable:

1   Right-click a member in the data layout and select **Edit**.

The Member Selection dialog box is displayed.

2   Select **Browse**.

3   In the Available pane click + to expand the Substitution Variable node.

4   To include a substitution variable, select the member in the Available pane and click .

5   To include a dynamic function on a substitution variable, right-click the member in the Available pane and select the desired function from the sub-menu:

- Member
- Children
- Descendants
- Bottom
- Siblings
- Same Level
- Same Generation
- Select Next/Previous
- Subsets

6   Click **OK**.

Variables

A variable is a constraint that a user selects when the query is processed.

➤ To use all selected members as a variable (the user is prompted for the values when the query is processed):

1   In the data layout, right-click a dimension member and select **Edit**.

2   Select **Use as variable**.

3   Click **OK**.

You can also a member in the data layout and on the shortcut menu, select Variable.

Searching Members

Use the Member Selection Search tab to find members within a dimension that match a pattern string in Essbase.

Note: The Search feature is also available in the Catalog.

Note: Only the member names and aliases are displayed and not the fully qualified name (for example, [West].[Salem]) in the Member Selection, Catalog or query results.

➤ To search for members:

1 In the data layout, right click a dimension and select **Edit**.

2 Select **Search**.

3 From the Member Type, select the type of member to search:

- Name—Member Name

- Alias—Alternate Names for database members

- Both—Member Name and Alias Name

Note: To search on alias names, enable the Use Alias Tables option and select an alias table in Query Options.

4 Enter the text string or numeric value to search (a member name or pattern).

Trailing asterisks and wildcard strings are acceptable. Examples of value text strings: Ja*, M?n, and M??n. Examples of invalid text strings: *-10 and J*n.

5 To locate all member within the selected dimension that matches the text string or numeric value, click

![search button].

The results of the search populate the Available pane.

6 **Optional:** To add a member to the Selected pane, click ![add button].

7 Click **OK**.

Member Selection Exclusions

To synchronize the Member Selection dialog selections with the results of grid navigation such as Drilling and Keep/Remove Only, additional member selections are sometimes added to the Selected pane to reflect member selections that are excluded from the query. The exclusions appear when dynamic member selections are used (for example, Product (Children)).

To arrive at the proper member selection, exclusions are included in the MDX query to avoid having metadata queries executed. For example, a query has the following member selection in the rows: EntityABC (Member+Children), and the result includes the parent member and 10,000 children. When performing a Remove Only on one of the child members, instead of querying the 10,000 children to update the member selection dialog, an exclusion entry is created for the removed member. Additional examples are as follows, with exclusion entries in bold:

Initial Query Selection	Query Operation	Resulting Member Selection
Year (Member+Children)	Remove Only: Qtr2	Year (Member+Children) **Qtr2 (-Member)**
Year (Member+Children)	Keep Only: Qtr2	Qtr2 (Member)

Initial Query Selection	Query Operation	Resulting Member Selection
Market (Member - Descendants)	Drill Up: Florida	Market(Member +Descendants)**East (- Descendants)** East (Member)

Subsets

Use subsets to define rules that select dimension member subsets, which satisfy specific conditions. These rules can be defined using the following data source constructs:

- Level—Members belonging to a specified level of the dimensional hierarchy

- Generation—Members belonging to a specified generation of the dimension hierarchy

- UDAs—Members with a specified user-defined attribute

- Name/Alias—Specified member name or alias. You cannot type in * or ? as part of the Name/Alias string.

- Masked Name or Alias—Members matching a pattern of wildcard characters.

- Attribute Dimensions—Members with a specified database-defined attribute

- Conditional logic—Members satisfying advanced subset member selection criteria

➤ To define a member subset selection:

1 Select a dimension member in the data layout and select **Edit** on the shortcut menu.

   The Member Selection dialog box opens.

2 Select a member in the Selected pane and on the shortcut menu, select **Subsets**.

   The Subset dialog box opens.

3 Select a subset type:

   - Levels

   - Generation

   - UDA

   - Name

   - Alias

   - Masked Name or Alias

   - [*Attribute Dimension(s)*]

4 Select an operand for the subsetting criteria.

   For example, *is* (equal to) or *is not* (not equal to).

5 Enter of select a value for the subsetting criteria from the third drop-down list.

6 To include criteria to the Conditions text box, click **Add**.

7 **Optional**: Repeat steps 3–6 for each subsetting criteria.

8 Click **OK**.

You must add at least one rule to the Subset definition for it to be used

Subset	Description
Select member " "and its descendants where (enter or select the values from all three fields)	Returns the subsetting methods available based on the type of item: ● Levels—Members belonging to a selected level of the dimensional hierarchy ● Generation—Members belonging to a selected generation of the dimension hierarchy ● UDA—Members with selected user defined attributes ● Name—Members matching a specified name of the dimension hierarchy ● Alias—Members matching a specified alias of the dimension hierarchy ● Masked Name or Alias—Members matching a pattern of wildcard characters. ● Attribute Dimension—Base dimension members associated with the specified Attribute dimension member based on the attribute dimension member type. Note that if the Member Selection dialog dimension does not have associated Attributes dimensions, none will be listed in the list box.
[second drop-down]	Displays operand (comparative filtering) criteria. For example, when you select **is**, the filtering criteria must equal the specified value. When you select **is not**, the filtering criteria excludes the members that meet the selected condition. Valid comparative filtering criteria includes: ● is ● is not ● greater than ● greater than or equal to ● less than ● less than or equal to
[third drop-down]	Displays the corresponding values for the subsetting methods.
Add	Adds the corresponding values to the Conditions text box.
Remove	Removes a selected item from the Conditions text box.
Remove All	Removes all items from the Conditions text box.
AND	Inserts the AND operand at the end of the currently selected rule. The AND operand is used by default when multiple rules are added to the Definition.AND logic requires that the selection(s) meet the current condition and the preceding condition.
OR	Inserts the OR operand at the end of the currently selected rule. OR logic requires that the current condition meet the current condition *or* the preceding condition.
(	Adds a left parenthesis to the selected item. Use parentheses to group two or more subsetting values to determine the order priority for conditions. Each item in the Conditions text box can have either the left or right parentheses, but not both.
)	Adds a right parenthesis to the selected item. Use parentheses to group two or more subsetting values to determine the order priority for conditions. Each item in the Conditions text box can have either the left or right parentheses, but not both.
Remove()	Removes all left or right parentheses from the selected item in the Conditions text box.
Conditions text box	Displays the subsetting conditions. Interactive Reporting uses these conditions to include or exclude members based on the conditions defined.

Subset	Description
Preview	Previews members meeting the subsetting selections.

Also Select Next/Previous

Use the Also Select Next/Previous dialog box to select the next or previous members of a selected member. For example, when the results for the Product dimension are:

Product

    Colas

    Root Beer

    Cream Soda

    Fruit Soda

    Diet Drinks

and Cream Soda is selected, the results for the next two members are Fruit Soda and Diet Drinks. If the two previous members are selected, the results are Colas and Root Beer.

➤ To select the next or previous member:

1   In the data layout, right-click a dimension member in the data layout.

The Member Selection dialog box opens.

2   Right-click a dimension member and choose **Select Next/Previous**.

The Also Select Next/Previous dialog box opens.

3   From the first drop-down, select next or previous.

4   From the second drop-down, select how many next or previous members to include.

5   Click **OK**.

Filter Member Selection

All data object intersections are relative to filter member selections, which focus intersections and data values, and, consequently, analysis. Filter-axis dimensions are by default represented by the highest aggregate member defined in the data source. To focus analysis on members other than the highest aggregate, you can select one filter member per dimension. Filter member selections do not rearrange dimensions or reorganize pages but focus analysis on intersections.

Use Filter Selection tabs to navigate through, search, or select the members to use in a filter definition.

Browsing Filter Members

Use the Filter Selection Browse tab to navigate through and select individual members to use as a filter.

➤ To browse and select member:

1   In the Filter pane of the data layout, right-click a dimension member and select **Edit**.

    The Filter Selection dialog box opens.

2   Select a member and click **+** to the left of the member name to expand it.

3   To include a member as a filter in the query, in the Available pane, select the member and click .

    The member is added to the Selected pane.

4   Click **OK**.

Searching Filter Members

Use the Search tab of Filter Selection dialog to search for members within selected dimensions.

Note: Only the member names and aliases are displayed and not the fully qualified name (for example, [West].[Salem]) in the Member Selection, Catalog or query results.

➤ To search for members:

1   In the Filter pane of the data layout, right click a member and select **Edit**.

2   Select **Search** tab.

3   From the **Member Type**, select the type of filter member to search:

    ● Name—Member Name

    ● Alias—Alternate Names for database members

    ● Both—Member Name and Alias Name

4   Type the text string or numeric value to search in the text box.

    Trailing asterisks and wildcard strings are acceptable. Examples of value text strings: Ja*, M?n, and M??n. Examples of invalid text strings: *-10 and J*n.

5   Click to locate all member within the selected dimension that matches the text string or numeric value.

    Hovering the mouse over an item in the Location column shows a tooltip with the members full location path.

    The results of the search populate the Available pane.

6   **Optional**: To add a member to the Selected pane, click .

7   Click **OK**.

Note: To add a different member as a filter, first remove the selected member.

Note: A Search by Alias name searches the currently selected Alias Table in Query Options when the Use Aliases option is enabled.

Data Filters

Data filters limit the data retrieved from the database by specific member selections for one or more dimensions. That is, data is returned from the database only if it meets the specified conditions set up in the filter. Data filters are applied on the server side (at the database), not on the client side.

Use the Data Filter dialog to do the following:

- Filter data based on top *n*/bottom *n* values

- Apply sorts to data/metadata

- Apply show or hide criteria to display or suppress members from appearing in a query. For queries with multiple dimensions in the rows or column, Show/Hide can be used to create asymmetric reports (queries that include nested dimensions that differ by at least one member across an axis).

The top *n* feature retrieves the number of highest data values from the Results set sorted in specified order. The bottom *n* option retrieves the lowest number of lowest data values from the Results set sorted by specified order.

The Sorting feature sorts rows or columns of the query result set in ascending or descending numeric order. Sorting definitions are dynamic and are applied as the document is drilled, pivoted, and changed.

The Show/Hide Only feature includes or excludes members by member name, tuples, or data value criteria. It is an effective means of focusing analysis by values.

The Data Filter dialog box is opened by right clicking on a dimension in the data layout and selecting Data Filter. Data Filters are applied by way of MDX queries to the Essbase server, which processes them, not locally by Interactive Reporting.

For example, if the "Product" dimension is in the Rows pane of the data layout and you select the Data Filter option, the heading of the Data Filter dialog box reads: Rows Filter, and the member context appears accordingly. The Member context is the member selection that is required for the Data Filter criteria.

For each of the types of filters, the following is the respective context. Assume a report with "Product" members in the row pane and the "Year" member in the column pane of the data layout. The Data Filter is applied to rows:

	Qtr1	Qtr2	Qtr3	Qtr4
100	7,048	7,872	8,511	7,037
200	6,721	7,030	7,005	7,198
300	5,929	6,769	6,698	6,403
400	5,005	5,436	5,698	5,162
Diet	7,017	7,336	7,532	6,941

Top/Bottom N—Because the "Product" members are filtered to show, for example, the "Top 5", the criteria for the data values must be defined on a specific column. As a result, the Data Filter dialog box displays a "Selected Columns" section from a "Year" dimension column is selected for the member context (bottom left):

Rows Filter

Filter type: Top/Bottom N

Show
☑ Top 5
☐ Bottom

Sort order:
Ascending

Sort on:
Qtr1
Qtr2
Qtr3
Qtr4

Selected:
Top 5 (Qtr1): Asc

Help   OK   Cancel

Sort—In a similar manner to Top/Bottom N, Sort requires the definition of the member context on the opposite axis. In the example, with "Product" dimension in the Rows pane of the data layout, the "Selected Columns" section from the "Year" dimension column is selected for the member context:

Rows Filter

Filter type: Sort

Sort order:
Ascending

Sort on:
Qtr1
Qtr2
Qtr3
Qtr4

Selected:
Top 5 (Qtr1): Asc

Help   OK   Cancel

Show/Hide—when the Data Filter dialog box is opened for a dimension, Show/Hide applies to the dimension selected, where the member context are members from that dimension. For example, with the "Product" dimension in the Rows pane of the data layout, the Data Filter dialog box displays a list of "Product" dimension members from which you can show or hide data:

Rows Filter

Filter type: Show/Hide

Select method:
Show

Where
● Member
○ Value   Condition

Members:
100-10
100-30
200-10
200-30
200-40
300-10
300-20
400-10
400-20
400-30
100-20
200-20

Selected:
Show (100-10):
Show (200-30):

Help   OK   Cancel

If two or more dimensions are in the columns or rows pane, the Data Filter dialog box displays all dimension combinations in the Member list pane, applicable. For example if the "Year" and "Scenario" are in the columns pane, a Rows Filter dialog shows all member combinations under Sort On:

Note: Filters for tuples not present in the grid of the Data Filter dialog box are greyed out.

➤ To filter a selected number of top rows or columns or a selected number of bottom rows or columns:

1   In the data layout, right click on a dimension (either in the Rows or Columns pane) on which to set the filter and select **Data Filter**.

The Data Filter dialog box is displayed.

2   Select **Filter Type**, then **Top/Bottom N**.

3   To retrieve the top *n* rows or columns of a dimension, select **Top** and in the text edit box. specify the number of rows or columns to retrieve. Enter a whole number only; decimal points and negative amounts are not accepted.

4   To retrieve the bottom *n* rows or columns of a dimension, select **Bottom** and specify the number of rows or columns to retrieve in the text edit box. Enter a whole number only; decimal points and negative amounts are not accepted.

5   From the **Sort Order** drop-down, select to sort rows or columns in **Ascending** or **Descending** order.

6   From the **Sort on** pane, select a column or row member from which to run the filter.

For Top/Bottom *n* and Sort data filters, an alternate axis member selection is required to apply the filter. For example, if you are applying a data filter against a column member, a row member must be selected.

7   Click to add the column member to the Selected pane.

8   Click **OK**.

➤ To sort data in ascending or descending order:

1   In the data layout, right click on a dimension (either in a row or column) on which to set the filter and select **Data Filter**.

The Data Filter dialog box opens.

2   Select **Filter Type**, then **Sort**.

3   From **Sort Order**, select to sort rows or columns in **Ascending** or **Descending** order.

4   From **Sort On**, select a column or row member from which to run the filter.

For a Top/Bottom *n* data filter and Sort data filter, an alternate axis member selection is required to apply the filter. For example, if you are applying a data filter against a column member, a row member must be selected.

5   Click to add the column member to the Selected pane.

6   Click **OK**.

➤   To show or hide a member or value in an asymmetric report:

1   In the data layout, right-click a dimension (in either the Rows or Columns) which to set the and select **Data Filter**.

   The Data Filter dialog box opens.

2   From the Filter type drop-down, Select **Show/Hide**.

3   From **Select method**, choose **Hide** or **Show**.

   By default Show is selected.

4   To show or hide a member in the row label, select **Member** from the Where pane and from the Members pane, choose the members.

5   To show or hide dimension members when all dimension member values satisfy a condition, from the Where pane, select **Values** from the Where pane, and from the Members pane, choose the values.

6   Set a condition on a value by selecting **Condition** and indicating a comparative condition on the Set Condition dialog.

7   Click to add the column member to the Selected pane.

   The selected member is displayed in the Selected pane as: Show or Hide ("member_name").
   The selected value condition is displayed in the Selected pane as Show or Hide
   (*"member_name"): val condition ("value"*).

8   Click **OK**.

Set Condition

Use the Set Condition dialog box to specify standardized comparison operators such as greater than, less than or equal to for dimension member values. This function is available for the Show/ Hide feature of Data Filters.

➤   To define a value condition for the Show/Hide feature:

1   from the Where pane of the Show/Hide definition, select **Value** for a dimension member value.

2   Click **Condition**.

   The Set Condition dialog box is displayed.

3   Select a comparison operator from (Operator), which includes:
   - = (equal to)
   - <> (less than, greater than) or (not equal to)
   - < (less than)
   - <= (less than, equal to)
   - > (greater than)

- >= (greater than, equal to)

4   **In the edit box, enter the numeric value to use as a condition.**

Periods and hyphens (for negative amounts) can be entered in the edit box.

5   **Click OK.**

The Data Filter dialog box is reopens.

6   **Click** **to add the conditioned value to the Selected pane.**

7   **Click OK.**

Persisting Data Filters

Data filters (Top/BottomN, Sort, or Hide/Show) are persisted even after the members selected for the data filter are no longer in the query. As long as the overall dimension selections in the data layout are not modified, the data filter can still be applied. This behavior enables users to retain the data filter definition even when the query is modified or rebuilt. If the dimension selections in the data layout are changed and the data filter definition is not removed or modified, the results set may not be synchronized with what has been requested.

To alert users to pre-existing data filter definitions when the members selected for the data filter are not in the query, CubeQuery includes the prompt: "A Data Filter definition can no longer be applied. Would you like to Leave the definition, Remove or Update it?"

Valid responses include:

- Leave—Preserves the data filter definition regardless of members that have been added or removed from the query.

- Remove—Removes the entire data filter definition.

- Update—Launches the Data Filter dialog box so that the user can update the data filter definition.

Under certain circumstances, the user may *not* be prompted:

- the user removes the selected "Member" selection type (via Member selection dialog) from the axis in which the definition is defined

- the user drills on a dimension axis in which the definition is defined and where the "Include selection" option is disabled (selected "Member" selection type is removed)

- the user select to keep only the dimension on the axis in which the definition is defined (selected "Member" selection type is removed)

- Members are added to the same dimensional rows and columns (all methods). For example, if members are added to a row axis where a Top/BottomN is defined for the Market dimension, and additional members are added to the Market dimension, then no prompt is displayed.

- Members are removed from the same dimensional rows and columns (all methods). If a member selection change is made to the opposite axis in which the data filter is defined, there is no prompt as long as the selected member is not removed.

- Member Selection (all methods including variables) changes are made from the same dimensional rows and column. If Member selection changes are made to an opposite axis, the prompt is not invoked unless the member is changed. For example, Year is the selected dimension, and Year is the selected member. In this case, Qtr1 and Qtr4 can be modified without a prompt, but if Year changed, the user is prompted.

- Drilling (all methods) does not invoke the prompt if drilling on the dimension where the Top/BottomN definition is applied, only the members. If the drilling is performed on the dimensional axes with the selected member (for example "Year") and the member remains, the prompt is not invoked.

- Keep Only (all methods)—same as drilling

- Remove Only (all methods)—same as drilling

Query Options

Use the Query Options to define global and local display behavior and drill options. The Query Options dialog box comprises the Global, Display, and Drill tabs.

➤ To display query options, select **Query**, then **Query Options**

Global Options

Use the Global tab to set global display behavior in the CubeQuery section.

Design Options	Description
Auto-Refresh Query	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.

Catalog Options	Description
Number of members to display	Sets the maximum number of members to display in the Catalog List and Member Selection. The number can be: ● unlimited ● 5 ● 10 ● 20 ● 50 ● 100 ● 250 ● 500 The default number of members is 50. In both the Catalog List and Member Selection, the "more…" node expands to the next set of members.

Results Options	Description
Download Database Totals	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.
Allow Relational Aggregation	Enables relational aggregation in the following sections: ● Break/Grand Totals in Results section ● Data Functions, Totals, Cume in Pivot ● Data Functions in Chart ● Data Functions, Break Totals, Show Column Totals in Report. When the Download Database Totals option (see above) is enabled, aggregations are disabled for all the sections dependent on the Results sets with database totals. This option protects users from performing aggregations, where the results may be counted twice. By default this option is disabled.
Include Consolidation Information	Enabling this option includes consolidation type/unary operator information of all members from the Data Layout when Download To Results is executed. Disabling the option excludes the consolidation type/unary operator information for each member. By default this option is disabled for newly inserted CubeQuery sections.

Display Options

Use the Display tab to set row and column suppression criteria, alias information, replacement values, and label indentation parameters.

Suppress	Description
#Missing Rows	Suppress the return of data rows that contain only missing values. If one value is in the row, the row is not suppressed. By default, this option is enabled.
Zero Rows	Suppress the return of data rows that contain only zeros. By default, this option is disabled.

Suppress	Description
Shared Members	Suppress the return of members tagged as shared. By default this option is disabled. When enabled, shared members are suppressed from the Catalog, query results and the Member Selection dialog. **Note:** When using shared members with non-unique Essbase outlines, the shared members are not recognized for suppression by Essbase. If "Suppress Shared Members" is enabled in a non-unique outline, shared members are not suppressed when they appear in a query. This occurs if the members are selected (for example. Member Selection dialog or Catalog) or navigated to (for example, Drilling, Keep/Remove Only).
#Missing Columns	Suppress the return of data columns that contain only missing data. By default, this option is enabled.
Zero Columns	Suppress the return of data columns that contain only zeros. By default, this option is disabled.

Aliases	Description
Use Aliases	Display aliases when performing database retrievals rather than database member names. Aliases are alternate names for database members. You can retrieve data that uses the database name, which is often a stock number or product code, or an alias, which can be more descriptive. When this setting is updated, this message is displayed: "The OLAP Tree will be updated automatically with new Alias information and the OLAP Query will be re-processed."
Select an Alias Table	Specify the alias table to use for alias names. Each database can contain one or more alias tables.

Replacement	Description
#Missing Label	Specify a label for missing values. For example, you might enter: `#Missing` By default the replacement value for a missing label is blank.
#NoAccessLabel	Specify a label for values from the Essbase cube to which a user does not have security access. For example, you might enter: `NoAccess`
Zero Label	Specify a label for zero values. By default the replacement value for a zero label is blank.

Label Indentation	Description

Indent Each Generation By	Select the number of characters by which each generation in the hierarchy is indented. For example, you can change the following indentation from 1 to 3. This option can take options from 0 to 25.
	
	indentation at 1:
	
	    100–10
	
	    100–20
	
	    100–30
	
	100
	
	indentation at 3:
	
	      100–10
	
	      100–20
	
	      100–30
	
	100

Drill Options

Use the Drill tab to define the next level of data displayed when you drill down in a CubeQuery.

Additionally, the Member Retention option group contains items that enable you to customize drilling retention characteristics.

Drill Level	Description
(Drill Level drop-down)	Defines the next level of data displayed when you drill down from the Query menu, or when you double click a member. Any drill down selection made from the shortcut menu overrides the selection made here. For example, if you select Bottom Level, data for the lowest level of members in a dimension is retrieved. A drill-down on Year retrieves: Jan, Feb, Mar, Apr, Ma, Ju, Jul, Aug, Sep, Oct, Nov, and Dec. Valid drill down levels are: ● Next Level ● Bottom Level ● All Descendants ● Sibling ● Same Level ● Same Generation

Member Retention	Description
Include Selection	Retains the selected member along with the other members retrieved as a result of a drill down. For example, if you drill down on Qtr1, the data for Jan, Feb, and Mar, as well as Qtr1, is retrieved. When this option is disabled, the data for only for Jan, Feb, and Mar, is retrieved: Qtrl is eliminated. By default this option is enabled.

Member Retention	Description
Within Selected Group	Applies (drilling) to only the group of members in which the selection is made. By default this setting is disabled. It is meaningful only when the report contains two or more dimensions of data down a report as rows or across a report as columns. This type of report is considered asymmetric, which is characterized by groups of nested members that differ by at least one member.
Remove Unselected Groups	Removes all dimension groups that are not in the selected group.

Navigating CubeQuery

For navigating and maintaining data in the CubeQuery section, see:

- Using Swing
- Keep Only
- Remove Only
- Drilling
- Drilling to a Shared Member

Using Swing

Use the Swing feature to swap dimensions between rows and columns from the query. This feature enables you to view your data in new ways, and easily compare the new data to data in the originating table. You swing dimensions by dragging the dimension headers (either column or row) from one position to another using the swing handles. By default swing handles are transparent. They only appear when you select a dimension header. When the swing handle appears, it is shown in white with a light grey handle. Once the selection is off, the handle is transparent again.

➤ To swing a dimension column:

1 Select a dimension column header (number 1 below) by clicking on the border to the left of the header.

The swing handle (number 2 below) is displayed to the right of the rows. In this example the *Scenario* dimension header is selected.

	Actual				Budget			
	Qtr1	Qtr2	Qtr3	Qtr4	Qtr1	Qtr2	Qtr3	Qtr4
100–10	5,096	5,892	6,533	5,206	6,510	7,410	8,220	6,760
100–20	1,359	1,534	1,528	1,287	2,240	2,480	2,470	2,340
100–30	593	446	400	544	1,040	770	750	950
100	7,048	7,872	8,511	7,037	9,790	10,660	11,440	10,050
200–10	1,697	1,734	1,833	1,887	2,630	2,730	2,900	3,380
200–20	2,963	3,079	3,149	2,834	3,600	3,640	3,700	3,790
200–30	1,153	1,231	1,159	1,093	1,240	1,370	1,300	1,140
200–40	908	986	814	1,384	1,010	1,100	930	1,490
200	5,721	7,030	7,005	7,198	8,480	8,840	8,830	9,800

2   Select the swing handle and drag it to the new position (left/bottom or right/top).

In the example, the handle has been swung downward and to the left of the *Product* dimension:

	Actual				Budget			
	Qtr1	Qtr2	Qtr3	Qtr4	Qtr1	Qtr2	Qtr3	Qtr4
100–10	5,096	5,892	6,583	5,206	6,510	7,410	8,220	6,760
100–20	1,359	1,534	1,528	1,287	2,240	2,480	2,470	2,340
100–30	593	446	400	544	1,040	770	758	950
100	7,048	7,872	8,511	7,037	9,790	10,660	11,440	10,050
200–10	1,697	1,734	1,883	1,887	2,630	2,730	2,900	3,380
200–20	2,963	3,079	3,149	2,834	3,600	3,640	3,700	3,790
200–30	1,153	1,231	1,159	1,093	1,240	1,370	1,300	1,140
200–40	908	986	814	1,384	1,010	1,100	930	1,490
200	6,721	7,030	7,005	7,198	8,480	8,840	8,830	9,800

As a result, the *Scenario* dimension is positioned to the left of the *Product* dimension:

		Qtr1	Qtr2	Qtr3	Qtr4
Actual	**100-10**	5,096	5,892	6,583	5,206
	100-20	1,359	1,534	1,528	1,287
	100-30	593	446	400	544
	100	7,048	7,872	8,511	7,037
	200-10	1,697	1,734	1,883	1,887
	200-20	2,963	3,079	3,149	2,834
	200-30	1,153	1,231	1,159	1,093
	200-40	908	986	814	1,384
	200	6,721	7,030	7,005	7,198
Budget	**100-10**	6,510	7,410	8,220	6,760
	100-20	2,240	2,480	2,470	2,340
	100-30	1,040	770	750	950
	100	9,790	10,660	11,440	10,050
	200-10	2,630	2,730	2,900	3,380
	200-20	3,600	3,640	3,700	3,790
	200-30	1,240	1,370	1,300	1,140
	200-40	1,010	1,100	930	1,490
	200	8,480	8,840	8,830	9,800

➤ To swing a row dimension:

1 Select an entire dimension row header.

The swing handle is displayed at the bottom of the query:

		Qtr1	Qtr2	Qtr3	Qtr4
Actual	100–10	5,096	5,892	6,583	5,206
	100–20	1,359	1,534	1,528	1,287
	100–30	593	446	400	544
	100	7,048	7,872	8,511	7,037
	200–10	1,697	1,734	1,883	1,887
	200–20	2,963	3,079	3,149	2,834
	200–30	1,153	1,231	1,159	1,093
	200–40	908	986	814	1,384
	200	6,721	7,030	7,005	7,198
Budget	100–10	6,510	7,410	8,220	6,760
	100–20	2,240	2,480	2,470	2,340
	100–30	1,040	770	750	950
	100	9,790	10,660	11,440	10,050
	200–10	2,630	2,730	2,900	3,380
	200–20	3,600	3,640	3,700	3,790
	200–30	1,240	1,370	1,300	1,140
	200–40	1,010	1,100	930	1,490
	200	8,480	8,840	8,830	9,800

2   Select and drag the swing handle to the new position (left/right or right/top).

In this example the handle is swing to the right and upward of the query:

		Qtr1	Qtr2	Qtr3	Qtr4
Actual	100-10	5,096	5,892	6,583	5,206
	100-20	1,359	1,534	1,528	1,287
	100-30	593	446	400	544
	100	7,048	7,872	8,511	7,037
	200-10	1,697	1,734	1,883	1,887
	200-20	2,963	3,079	3,149	2,834
	200-30	1,153	1,231	1,159	1,093
	200-40	908	986	814	1,384
	200	6,721	7,030	7,005	7,198
Budget	100-10	6,510	7,410	8,220	6,760
	100-20	2,240	2,480	2,470	2,340
	100-30	1,040	770	750	950
	100	9,790	10,660	11,440	10,050
	200-10	2,630	2,730	2,900	3,380
	200-20	3,600	3,640	3,700	3,790
	200-30	1,240	1,370	1,300	1,140
	200-40	1,010	1,100	930	1,490
	200	8,480	8,840	8,830	9,800

The results of the swing is:

	100-10				100-20				100-30			
	Qtr1	Qtr2	Qtr3	Qtr4	Qtr1	Qtr2	Qtr3	Qtr4	Qtr1	Qtr2	Qtr3	Qtr4
Actual	5,096	5,892	6,583	5,206	1,359	1,534	1,528	1,287	593	446	400	544
Budget	6,510	7,410	8,220	6,760	2,240	2,480	2,470	2,340	1,040	770	750	950

Note that the Product dimension is positioned under the Year dimension.

Keep Only

The Keep Only feature enables you to focus on one member by allowing you to clear all other dimension member selections except the selected member from the Results set. Non-kept members are not available on the Member Selection dialog box.

➤ To keep specific set members:

1   Select one member or a range of members.

Ctrl + click to keep nonadjacent cells.

2   Right-click a dimension label and select **Keep Only**.

Only the selected members are shown in the report.

Remove Only

Clears a dimension member to remove it from the query result set. At least one member must be retained to use this feature. Removed members are not included in the results set, but display on Member Selection dialog as excluded from the query (flagged with a - or minus sign). To add a member back to the query, the exclusion must be from the Selected pane. See also Member Selection Exclusions.

➤ To remove a selected member:

1 **Select one member or a range of members.**

   Ctrl + click to remove nonadjacent cells

2 **Right-click a dimension label and select Remove Only.**

   The member is removed from the report and the query. To add the member to the query again, add it from the Catalog to the data layout, or select it on the Member Selection dialog box.

Suppressing Missing and Zero Values

Suppressing missing and zero values in rows and columns prevents irrelevant information from being returned, reduces network traffic, and increases query speed. Query options box enables you to customize how CubeQuery handles missing and zero values, including the text label displayed for these values by default. You can suppress the values on the shortcut menu.

➤ To suppress missing values in rows, right-click a row and select **Suppress**, then **Missing rows**.

By default, missing values are blank.

➤ To suppress missing values in columns, right-click a column and select **Suppress**, then **Missing columns**.

➤ To suppress zero values in rows, right-click a row and select **Suppress**, then **Zero rows**.

By default zero values are shown as 0 (zero).

➤ To suppress zero values in columns, right-click a column and select **Suppress**, then **Zero columns**.

Drilling

Drilling increases or decreases the display of report detail for dimensions with large amounts of level data. It can consist of drilling down (in many forms) or drilling up. Drilling down retrieves more detailed data within a dimension. You can drill down into more detailed data until the lowest level of a dimension as defined in the database outline is reached.

A before and after drill down example is shown below:

Before drilling:

After drilling:

In this example for October, you can drill down to Audio and Visual. Drilling up is the opposite of drilling down, where you move up the hierarchy within a dimension to less detailed data. Query Options enables you to customize the behavior of the drill level. Shortcut menus also control the drill levels.

Note: You cannot drill down on a Date Time Series member.

➤ **To drill down to more detail:**

1   **Select the member.**

2   **Double-click the member to drill down using the default Drill Level as defined in Query Options.**

   Tip: You can also right-click a member, and select Drill, then Down to drill down using the default drill level defined in .

   Additional drill down options include:

   - Down—Drills down to more dimension detail using the default drill level defined in Query Options.

   - Up—Drills up to less dimension detail.

   - Next—Drills down to the children. This is the default Drill Level. For example, a drill on Year retrieves Qtr1, Qtr2, Qtr3, and Qtr4

   - Bottom—Drills down to the lowest level of members in a dimension. For example, a drill on Year retrieves Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, and Dec.

   - All Descendants—Retrieves data for all descendants. For example, a drill on Year retrieves all quarterly and monthly members.

   - Siblings—Retrieves data for siblings. For example, a drill on Jan retrieves Jan, Feb, and Mar.

- Same Level—Retrieves data for all members at the same level. For example, a drill on Sales might retrieve values for COGS, Marketing, Payroll, Misc, Opening Inventory, Additions, Ending Inventory, Margin %, and Profit %.

- Same Generation—Retrieves data for all members of the same generation as the selected member or members. For example, a drill on Sales retrieves COGS, Marketing, Payroll, and Misc.

Drilling to a Shared Member

You can drill down or drill up on a member that has a shared member defined in the Essbase outline. Essbase determines which members are eligible — the base member or the shared member, and returns drilled or stored members based on the drill path.

This is an Essbase outline:

Product

    100

        150 (stored member)

            100-10

            100-20

    Brand1

        150 (shared member)

The stored member 150 has children; the shared member 150 does not. Drilling up and down on 150 gives different results:

- Drilling down on 150 returns nothing if it is interpreted as the shared member; or, returns 100-10 and 100-20 if it is interpreted as the regular member.

- Drilling up on 150 returns Brand1 if it is interpreted as the shared member; or returns 100 if it is interpreted as the regular member.

The proximity of the shared member to the regular member gives different results when drilling down. When drilling up on a member that has a shared member, Essbase looks at the members to determine which one is being queried. For example, drilling up on 100-20 sometimes returns Diet and returns 100, depending on whether Diet or 100 is closest to the member 100-20. If 100-20 is alone, then Essbase determines that this is the regular member. If Diet is closer, then 100-20 may be interpreted as the shared member.

Drill Through To Relational

The Drill Through to Relational feature enables users to drill from a data cell or meta data label in a CubeQuery section to an existing relational query section within the same Interactive Reporting document file. Drill Through to Relational has these characteristics:

- Essbase dimensions are mapped to the target relational fields, so that the member context in the Essbase query can be passed to the relational query as a filter.

  ○ Fact mapping is not available. Facts are defined in the relational query section only.

  ○ Mapping from the topic and topic items to the Essbase dimensions is done in CubeQuery. One relational query can be defined for a drill through for each CubeQuery section.

- You can specify which dimension context is passed or drilled on. If a relational field is not mapped to an Essbase dimension, then context for that dimension is not be passed. Drilling on column, row or cell defines which context is passed, based on the cell's member intersection or metadata label.

- Drilling can be accomplished at any level of Essbase members in the hierarchy.

- You can drill a to new or existing Results section.

Note: The Drill to Relational feature does not include EIS Drill Through Report support, as the target relational report must exist in the same document.

Setting Drill-Through to Relational Mapping

The mapping of Essbase dimensions to relational fields takes place in the Essbase query on the Set Drill Through dialog box.

➤ To set the drill to relational feature:

1   Verify that a relational query is in the Interactive Reporting document file.

The relational query is used as the target for the drill through.

If a relational query does not exist in the Interactive Reporting document file, the Set Drill Through menu item is disabled.

2   Select **Query**, then **Set Drill Through**.

The Set Drill Through dialog box is displayed.

3   In the **Relational Query** drop down box, select the relational query to use as a target for the drill through.

4   In the **Dimension to Topic Mapping** section, select the **Topic** (relational table) and **Topic Item** (fields for the selected table/topic) from the drop down boxes.

   Note: Multiple topic items can be mapped to the same dimension. There are some cases where multiple levels/generations in the Essbase hierarchy might exist in different topic items, therefore both items need to be mapped. In addition, the member name and alias might exist in separate topic items and both can be mapped to the same Essbase dimension.

5   In **Dimension** drop down box, select the Essbase dimension to be mapped.

Dimension lists all dimensions in the Essbase cube. The dimension "Attribute Calculations" is never included in this list.

6 Click ![checkmark button] to add the mapping to the list.

The dimension is added to the Selected area.

The two columns in the mapping list, Relational and OLAP, can be resized by dragging the column label.

7 To allow drilling on a data cell in CubeQuery, select **Enable Cell Drilling**.

8 To display indicators for data cells and metadata labels where drilling can be executed, select **Enable Indicators**.

By default indicators are not shown.

Since all data cells and/or all member labels in a dimension can be drilled on, the blue cell indicators are only displayed for the top/left cell or member label as shown below

		New York	Massachusett
Jan	100-10	678	494
	100-20		
	100-30		
	200-10	61	126
	200-20		
	200-40	490	341
Feb	100-10	645	470
	100-20		
	100-30		
	200-10	61	128
	200-20		
	200-40	580	340

9 From the **Drill To** drop down box, select where to show the results of the drill. Valid options are **New Results** or **Existing Results**.

10 Select **OK**.

➤ To exclude a dimension from the drill path:

1 Select **Query**, then **Set Drill Through**.

The Set Drill Through dialog box is displayed.

2 Uncheck the "active" box in the Selected area next to the dimension.

☑ All	
Relational	**OLAP**
☑ Products.Product Id	Year
☐ Stores.Store Id	Year

Excluding dimensions in this manner allows mapping definitions to be defined, but not necessarily included in the drill action. The All check box allows for enabling or disabling of all items in the list.

3   Click **OK**.

Executing a Drill-Through To Relational

In the Query section, you execute the drill through to relational. If drill through indicators have been enabled, the blue cell indicators are only displayed for the top/left cell or member label because all data cells and all member labels in a dimension can be drilled on.

➤ To execute a drill through:

1   Select a single cell or member label and on the shortcut menu, select **Drill Through**.

Drill through can only be performed on one label or cell at a time. If multiple metadata labels and/or cells are selected, the option to drill through is not be available on the shortcut menu.

The Drill to Results dialog is displayed.

This dialog box is only displayed if the option "Existing Results" was selected in the Drill To drop down box. You are prompted to enter a user name and password to the relational query.

By default there is New Results, and the dialog box is not shown.

2   Select the target Results section and click **OK**.

Sample Drill-Through

The sample below shows the drill through feature applied to a CubeQuery with multiple dimensions in the rows.

The initial CubeQuery section includes:

Rows

Jan, Feb, March

100 (children)

200 (children)

Columns

    East (children)

Filters

    Sales

    Cogs

		New York	Massachusetts	Florida	Connecticut	New Hampshire
Jan	100-10	678	494	210	310	120
	100-20			200		
	100-30					93
	100	678	494	410	310	213
	200-10	61	126	190	180	90
	200-20			180	130	
	200-40	490	341			65
	200	551	467	370	310	155
Feb	100-10	645	470	200	325	114
	100-20			206		
	100-30					101
	100	645	470	406	325	215
	200-10	61	128	220	155	104
	200-20			180	130	
	200-40	580	340			68
	200	641	468	400	285	172
Mar	100-10	675	492	210	309	119
	100-20			214		
	100-30					107
	100	675	492	424	309	226
	200-10	63	125	227	150	107
	200-20			188	124	
	200-40	523	325			75
	200	586	450	415	274	182

The relational section includes:

| Request | Product | Product Na | Month | State | City | Sales | Cogs |

The initial results of the relational query are:

	Product	Product Na	Month	State	City	Sales	Cogs
1	100-10	Cola	Jan	Colorado	Aspen	168	73
2	100-20	Diet Cola	Jan	Colorado	Aspen	179	72
3	100-30	Caffeine Free Cola	Jan	Colorado	Aspen	169	68
4	200-10	Old Fashioned	Jan	Colorado	Aspen	29	11
5	200-20	Diet Root Beer	Jan	Colorado	Aspen	141	56
6	100-10	Cola	Feb	Colorado	Aspen	167	72
7	100-10	Cola	Mar	Colorado	Aspen	166	72
8	100-10	Cola	Apr	Colorado	Aspen	4	1
9	100-10	Cola	May	Colorado	Aspen	121	52
10	100-10	Cola	Jun	Colorado	Aspen	64	27

Product Product Na Month State City Sales Cogs

The following topic mappings are defined on the Set Drill Through dialog:

Table 49   Topic Mappings defined in the Set Drill Through Dialog

Option	Value
Relational Query	Query1
Topic	Details
Dimension	Year
Topic	State
All	Enabled
Enabled Cell Drill	Enabled
Enable Indicators	Enabled

Option	Value
Drill To	Existing Results

When drill on the first top/left cell, the context of New York, 100-10 and Jan. is passed as a filter from the CubeQuery to the relational query:

Filter		Product	Month	State			

	Product	Product Na	Month	State	City	Sales	Cogs
1	100-10	Cola	Jan	New York	Buffalo	478	191
2	100-10	Cola	Jan	New York	Ithaca	9	3
3	100-10	Cola	Jan	New York	Long Island	56	22
4	100-10	Cola	Jan	New York	Manhattan	128	52
5	100-10	Cola	Jan	New York	Millbrook	4	1
6	100-10	Cola	Jan	New York	Poughkeepsie	0	0
7	100-10	Cola	Jan	New York	Yonkers	0	0
8	100-10	Cola	Jan	New York	Yorktown	3	2

Downloading to Results

Download to Results is an instrument for rendering a flat table representation of the multi-dimensional Essbase data cubes in Interactive Reporting. The data is loaded into a Result section of Interactive Reporting document, and can be used for further analysis, report creation and joining with data from other relational data sources. In this case the data processing (e.g. aggregation, totaling, custom calculations, table joins) is performed on the Interactive Reporting side.

By default Download to Results creates separate columns in the results set for every generation in every dimension. In addition, only the lowest level member values are imported, with any ancestors for each member (included in the original query) appearing in additional columns. In the following example, separate columns appear for the two Product member Generations from the query. Only the lowest level *Product* members data are in the query. Data for the upper level *Product* members is not included:

	Category	Product SKU	Gen2,Scenario	Gen2,Year	Value
1	100	100-10	Actual	Qtr1	5,096
2	100	100-20	Actual	Qtr1	1,359
3	100	100-30	Actual	Qtr1	593
4	200	200-10	Actual	Qtr1	1,697
5	200	200-20	Actual	Qtr1	2,963
6	200	200-30	Actual	Qtr1	1,153
7	200	200-40	Actual	Qtr1	908
8	100	100-10	Budget	Qtr1	6,510
9	100	100-20	Budget	Qtr1	2,240
10	100	100-30	Budget	Qtr1	1,040
11	200	200-10	Budget	Qtr1	2,630
12	200	200-20	Budget	Qtr1	3,600
13	200	200-30	Budget	Qtr1	1,240
14	200	200-40	Budget	Qtr1	1,010
15	100	100-10	Actual	Qtr2	5,892
16	100	100-20	Actual	Qtr2	1,534
17	100	100-30	Actual	Qtr2	446
18	200	200-10	Actual	Qtr2	1,734

For information on using the Download Database Totals option with this feature, see Download Database Totals .

The Results set can be integrated with the Chart, Table, and reporting sections.

Results are only updated when you select the Download to Results feature, or when you take the downloaded results set and process the results again.

If you expect the query to retrieve a small to medium sized data set, it is recommended that you enable the automatic download feature from the Query, then Tools, then Options, then Program Options, then OLAP, and then Auto Generate Results When Processing OLAP Query. If you choose this option, the Results set is not created for the current CubeQuery section, but only for new CubeQuery sections. In some circumstances when querying large amounts of data, the automatic creation of a Results section may result in a slight reduction in query performance.

Note: Shared members can be excluded from the query by way of Query Options, however there are some cases where customers might want to include Shared Members in the query and results set, but not in the totals. If you want to include shared members in the results set, the parent context needs to exist in the query. In other words, if a shared member's parent does not exist in the query, Download to Results does not recognize that it is a shared member.

Note: If you open anInteractive 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.

➤ To download the query to results, select **Query**, then **Download to Results.**

The behavior of the Download to Results feature varies for different components in CubeQuery:

- Separate Columns for Metadata Labels
- Measure Behavior in Columns
- Shared Members
- Including Totals

Separate Columns for Metadata Labels

Separate columns for metadata labels are displayed for each Essbase generation in the hierarchy.

Table 50 CubeQuery Requested Items (Profit Member Applied In the Filter)

Product	Qtr1	Qtr2	Qtr3	Qtr4
Cola	5096	5892	6583	5206
Diet Cola	1359	1534	1528	1287
Caffeine Free Cola	593	446	400	544
Cola	7048	7872	8511	7037
Old Fashioned	1697	1734	1883	1887
Diet Root Beer	2963	3079	3149	2834
Sarsaparilla	1153	1231	1159	1093
Birch Beer	908	986	814	1384
Root Beer	6721	7030	7005	7198

Table 51 Results Set

Category	Product SKU	Quarter	Profit
Colas	Cola	Qtr1	5096
Colas	Diet Cola	Qtr1	1359
Colas	Caffeine Free Cola	Qtr1	593
Root Beer	Old Fashioned	Qtr1	1697
Root Beer	Diet Root Beer	Qtr1	2963
Root Beer	Sarsaparilla	Qtr1	1153
Root Beer	Birch Beer	Qtr1	908

If only one generation of the hierarchy is in the query, additional columns do not show the parent members in the Results set as shown below:

Table 52   CubeQuery Requested Items (Profit Member is in the Columns)

	Qtr1	Qtr2	Qtr3	Qtr4
	Profit	**Profit**	**Profit**	**Profit**
Cola	5096	5892	6583	5206
Diet Cola	1359	1534	1528	1287
Caffeine Free Cola	593	446	400	544
Old Fashioned	1697	1734	1883	1887
Diet Root Beer	2963	3079	3149	2834
Sarsaparilla	1153	1231	1159	1093
Birch Beer	908	986	814	1384
Dark Cream	2544	3231	3355	3065

Table 53   Results Set

Product SKU	Quarter	Profit
Cola	Qtr1	5096
Cola	Qtr2	5892
Cola	Qtr3	6583
Cola	Qtr4	5206
Diet Cola	Qtr1	1359
Diet Cola	Qtr2	1534
Diet Cola	Qtr3	1528
Diet Cola	Qtr4	1287
Caffeine Free Cola	Qtr1	593
Caffeine Free Cola	Qtr2	446
Caffeine Free Cola	Qtr3	400
Caffeine Free Cola	Qtr4	544

Measure Behavior in Columns

In the CubeQuery section, the measure dimension is handled like any other dimension (it can be placed in rows or columns), but a separate Results set column is created for each Measure member, whether it is in a row, column, or filter.

Table 54   Measures in CubeQuery

		Qtr1	Qtr2
Sales	Cola	14585	16048
	Diet Cola	7276	7957
COGS	Cola	5681	6136
	Diet Cola	3496	3871
	Caffeine Free Cola	1493	1606

Table 55   Measures in Results Set

Product SKU	Quarter	Sales	COGS
Cola	Qtr1	14585	5681
Diet Cola	Qtr1	7276	3496
Caffeine Free Cola	Qtr1	3187	1493

If no Measures are in the row, column or filter, all data values appear in a Value column.

Table 56   Values Column in Results Set

Product SKU	Quarter	Value
Cola	Qtr1	14585
Diet Cola	Qtr1	7276
Caffeine Free Cola	Qtr1	3187

Ragged Hierarchies

In Essbase, an individual hierarchy has the same number of members above it as any other member at the same level. In a ragged hierarchy, the logical parent member for one member is not in the level directly above the member.

In the following example Qtr2 does not have children in the query. Because only the lowest level members are included in the query, the Results set includes: Jan, Feb, Mar, and Qtr2. For the "month" value for Qtr2, the label is blank.

Table 57   Ragged Hierarchy in CubeQuery Section

	Product
Jan	8024
Feb	8346
Mar	8333
Qtr1	24703
Qtr2	27107

Table 58   Ragged Hierarchy in Results Set

Quarter	Month	Gen1,Product	Value
Qtr1	Jan	Product	8024
Qtr1	Feb	Product	8346
Qtr1	Mar	Product	8333
Qtr2	(blank)	Product	27107

If a child member does not have a parent member in the original query, the parent or any other ancestor is included in the Results set:

Table 59   Ragged Hierarchy in CubeQuery Section

	Product
Jan	8024
Feb	8346
Mar	8333
Qtr1	24703
Dec	8780

Table 60   Ragged Hierarchy in Results Set

Quarter	Month	Gen1,Product	Value
Qtr1	Jan	Product	8024
Qtr1	Feb	Product	8346
Qtr1	Mar	Product	8333
Qtr4	Dec	Product	8780

Shared Members

Shared member data can be included in a query, or their totals can be excluded. Each shared member column in the Results set corresponds to an Essbase Generation/Field in the Results set and query. For Shared Members, the parent context must exist in the query. In other words, if the parent of a shared member is not in the query, the Download to Results feature does not recognize that it is a shared member and determines the parent. That is, the parents of a shared member must exist in the query in order to determine that they are shared for the results set. For example, in the Sample Basic Product dimension, for the second instance of "100-20" (shared member) to be properly accounted for in the Download to Results, its parent "Diet" must be included in the query such as "Diet (Children)".

Note: Shared members are in bold below.

Table 61   Shared Members in CubeQuery Section

	Qtr1
100–10	5096
100–20	1359
100–30	593
100	7048
200–10	1697
200–20	2963
200–30	1153
200–40	908
200	6721
300–10	2544
300–20	690
300–30	2695
300	5929
400–10	2838
400–20	2283
400–30	–116
400	5005
100–20	1359

	Qtr1
200–20	2963
300–30	2695
Diet	7017
Product	24703

Table 62   Shared Members in Results Set

Product SKU, Shared	Gen1, Product	Category	Product SKU	Quarter	Value
	Product	100	100–10	Qtr1	5096
	Product	100	100–20	Qtr1	1359
	Product	100	100–30	Qtr1	593
	Product	200	200–10	Qtr1	1697
	Product	200	200–20	Qtr1	2963
	Product	200	200–30	Qtr1	1153
	Product	200	200–40	Qtr1	908
	Product	300	300–10	Qtr1	2544
	Product	300	300–20	Qtr1	690
	Product	300	300–30	Qtr1	2695
	Product	400	400–10	Qtr1	2838
	Product	400	400–20	Qtr1	2283
	Product	400	400–30	Qtr1	–116
100–20	Product	Diet	100–20	Qtr1	1359
200–20	Product	Diet	200–20	Qtr1	2963
300–30	Product	Diet	300–30	Qtr1	2695

If multiple generation dimensions have shared members, one "shared" indication per dimension
is created:

Note: Shared members are in bold below.

Table 63   Multiple Dimensions with Shared Members in CubeQuery Section

		Qtr1
Actual	100–10	5096
	100–20	1359
	100	7048
	100–20	1359
	Diet	7017
Budget	100–10	6510
	100–20	2240
	100	9790
	100–20	2240
	100	9790
	100–20	2240
	Diet	8910
Actual	100–10	5096
	100–20	1359
	100	7048
	100–20	1359
	Diet	7017

Table 64   Multiple Dimension With Shared Members in Results set

Scenario, Gen2, Shared	Scenario, Gen2	Product SKU, Shared	Category	Product SKU	Quarter	Value
	Actual		100	100–10	Qtr1	5096
	Actual		100	100–20	Qtr1	1359
	Actual	**100–20**	Diet	100–20	Qtr1	1359
	Budget		100	100–10	Qtr1	6510
	Budget		100	100–20	Qtr1	2240
	Budget	**100–20**	Diet	100–20	Qtr1	2240
Actual	Actual		100	100–10	Qtr1	5096
Actual	Actual		100	100–20	Qtr1	1359

Scenario, Gen2, Shared	Scenario, Gen2	Product SKU, Shared	Category	Product SKU	Quarter	Value
Actual	Actual	**100–20**	Diet	100–20	Qtr1	1359

If a dimension contains shared members at multiple generations, a column is added to the Results set for each CubeQuery generation and Results set column:

Table 65   Shared Members At Multiple Generation Level in the CubeQuery Section

	Qtr1
100–10	5096
100–20	1359
100–30	593
100	7048
...	
400–10	2838
400–20	2283
400–30	–116
400	5005
100–20	1359
200–20	2963
300–30	2695
Diet	7017
400	5005
Product	24703

Table 66   Shared Members at Multiple Generation Level in the Results Set

Category, Shared	Product SKU, Shared	Gen1, Product	Category	Product SKU	Quarter	Value
		Product	100	100–10	Qtr1	5096
		Product	100	100–20	Qtr1	1359
		Product	100	100–30	Qtr1	593
...						

Category, Shared	Product SKU, Shared	Gen1, Product	Category	Product SKU	Quarter	Value
		Product	400	400–10	Qtr1	2838
		Product	400	400–20	Qtr1	2283
		Product	400	400–30	Qtr1	–116
	100–20	Product	Diet	100–20	Qtr1	1359
	200–20	Product	Diet	200–20	Qtr1	2963
	300–30	Product	Diet	300–30	Qtr1	2695
400		Product	400		Qtr1	5005

Including Totals

To prevent data values from being counted twice, totals and upper level members are not included in the downloaded results set.

In the Pivot section, shared members can be excluded from totals by disabling the Aggregate Shared Members option.

In addition, in the Pivot section all results are summed. Some Essbase aggregations may not sum member results due to outline calculations and member unary operators (+, –, *, /, ~). As a result, there may be difference shown between Essbase and Pivot totals depending on the outline structure and calculations of the cube.

Note: Shared members are in bold below.

Table 67   Shared Members (bolded) in the CubeQuery Section

	Qtr1
100–10	5096
100–20	1359
100–30	593
100	7048
...	
400–10	2838
400–20	2293
400–20	–116
400	5005

	Qtr1
100–20	1359
200–20	2963
300–30	2695
Diet	7017
Product	24703

Note: The Product total does not include the "Diet" member total of "7017" because it is configured in the Analytic Service outline to "no consolidate."

Note: If the same results are inserted into a pivot, the Product total includes the value for the Diet member. Shared members are included in totals if the aggregation option is disabled in the Results section, but the "Product SKU, Shared" column is added to the data layout in the Pivot section. If the aggregation option is enabled in the Results section, share members are included in totals in the Pivot section whether the "Product SKU, Shared" is the data layout or not.

Table 68   Aggregate Shared Members Enabled in the Results Set

	Qtr1
100–10	5096
100–20	1359
100–30	593
Total	7048
...	
400–10	2838
400–20	2283
400–30	–116
Total	5005
100–20	1359
200–20	2963
300–30	2695
Total	7017

	Qtr1
Total	31720

If the "Aggregate Shared Members" option is disabled, the total is included for the initial shared member, but not for any parent members above it:

Table 69 Shared Members Disabled in the Pivot Section

	Qtr1
100–10	5096
100–20	1359
100–30	593
Total	7048
...	
400–10	2838
400–20	2283
400–30	–116
Total	5005
100–20	1359
200–20	2963
300–30	2695
Total	7017
Total	24703

Download Database Totals

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 also included in the results. By default this option is disabled. Once this option is enabled, any existing Results set is deleted. If this option is enabled through **Tools**, then **Options**, then **Program Options**, then **OLAP**, then the option is enabled also in a newly created CubeQuery section.

Note: If you use the Download to Results feature, only separate columns in the Results set are created for every generation in every dimension. In addition, only the lowest level member values are imported, with any ancestors for each member (included in the original query) appearing in additional columns. For an example, see Downloading to Results .

In the following example:

- All *Product* members in the query, are listed in only one column and not separated into multiple columns based on Generation. In addition, the data values for the upper level Product members are also included in the results.

- Each dimension has an additional column that displays parent members for each member in the query.

  ○ The *Year* dimension members Qtr1, Qtr2, etc. have their parent member (Year) displayed in the "Year Parent" column.

  ○ The *Scenario* dimension members Actual and Budget have their parent member (Scenario) displayed in the "Scenario Parent" column.

  ○ The *Product* dimension members have their respective parent members displayed in the "Product Parent" column.

	Product Parent	Product	Scenario Parent	Scenario	Year Parent	Year	Value
1	100	100-10	Scenario	Actual	Year	Qtr1	5,096
2	100	100-20	Scenario	Actual	Year	Qtr1	1,359
3	100	100-30	Scenario	Actual	Year	Qtr1	593
4	Product	100	Scenario	Actual	Year	Qtr1	7,048
5	200	200-10	Scenario	Actual	Year	Qtr1	1,697
6	200	200-20	Scenario	Actual	Year	Qtr1	2,963
7	200	200-30	Scenario	Actual	Year	Qtr1	1,153
8	200	200-40	Scenario	Actual	Year	Qtr1	908
9	Product	200	Scenario	Actual	Year	Qtr1	6,721
10	100	100-10	Scenario	Actual	Year	Qtr2	5,892
11	100	100-20	Scenario	Actual	Year	Qtr2	1,534
12	100	100-30	Scenario	Actual	Year	Qtr2	446
13	Product	100	Scenario	Actual	Year	Qtr2	7,872
14	200	200-10	Scenario	Actual	Year	Qtr2	1,734
15	200	200-20	Scenario	Actual	Year	Qtr2	3,079
16	200	200-30	Scenario	Actual	Year	Qtr2	1,231
17	200	200-40	Scenario	Actual	Year	Qtr2	986
18	Product	200	Scenario	Actual	Year	Qtr2	7,030
19	100	100-10	Scenario	Actual	Year	Qtr3	6,583
20	100	100-20	Scenario	Actual	Year	Qtr3	1,528
21	100	100-30	Scenario	Actual	Year	Qtr3	400
22	Product	100	Scenario	Actual	Year	Qtr3	8,511

Note: If you open anInteractive 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 will not be the same as it was with version 11.1.1.

➤ To enable download database totals:

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

The Query Options dialog box is displayed.

2 Select the **Global** tab.

3 Enable **Download Database Totals** and click **OK**.

Relational Aggregation

When selecting Download Database Totals, aggregations are disabled for all the sections dependent on the Results with database totals. This feature protects users from performing aggregations, where the results maybe double counted. Enabling the option Allow Relational Aggregation allows usage of aggregation in the following sections:

- Break/Grand Totals in Results section
- Data Functions, Totals, Cume in Pivot
- Data Functions in Chart
- Data Functions, Break Totals, Show Column Totals in Report.

➤ To set relational aggregation:

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

2   Enable **Allow Relational Aggregation** and click **OK**.

Showing as Chart

Use the Show as Chart option to chart the CubeQuery data set and automatically create a Results set. Selecting this feature downloads the CubeQuery to results and create a table section as a parent for the chart section. If a table or chart section already exists for the CubeQuery section, a new section is not created.

➤ To show the CubeQuery as a Chart:

1   Build and process the CubeQuery.

    (See Building a CubeQuery Section)

2   Select **Query**, then **Show As Chart**.

Exporting a CubeQuery

The CubeQuery can be exported to these formats:

- Microsoft Office 2000 HTML—Results can available in Microsoft Excel 2000
- Microsoft Office Web Archive (*.mhtml)—Results are available in Microsoft Excel XP and 2003
- HTML (*.htm)—Results are available as a web page, complete with include files
- PDF (*.pdf)—Adobe Portable Document Format (PDF) that is readable by way of Adobe Reader.

➤ To export a CubeQuery results set:

1   Select **File**, then **Export**, and then **Section**.

The Export Section dialog box is displayed.

2 From **Save as type**, select the export format type.

3 Click **Save**.

Launching Smart View from a CubeQuery Section

Note: This feature is only in the Interactive Reporting Web Client and EPM Workspace. It is not available for Interactive Reporting Studio.

A CubeQuery section can be exported to Oracle® Hyperion® Smart View for Office. Smart View is a centralized interface which enables simultaneous use of multiple Oracle Hyperion products, and improves integration with Microsoft Office. You can export a CubeQuery section to Microsoft Excel as query-ready HTML. When you export content as query-ready HTML, the current page of the current CubeQuery section is converted to HTML and Hyperion-specific formatting is removed. This enables you to re-query the data source independent of the Web application.

For more information about exporting a CubeQuery with Smart View, see *Oracle® Hyperion Oracle Hyperion Smart View for Office, Fusion Edition For Office, Fusion Edition, Release 11.1.1.*

➤ In Interactive Reporting Web Client, to launch Smart View from a CubeQuery section, select **Query**, then **Launch Smart View**.

Formatting CubeQuery Items

Common formatting options are available for CubeQuery items on the Format and shortcut menus including:

- Font
- Style
- Number
- Justify
- Border
- Row Height
- Column Width
- Auto-Size Width
- Auto-Size Height
- Data Labels (available only for an OLAPQuery section, and not a CubeQuery section).
- Corner Labels
- Conditional Formatting

- Defining Export Properties for HTML in the Pivot and OLAP Sections

➤ To modify a font format:

1 Select an item and select **Format**, then **Font.**

The Font tab of the Properties dialog box opens.

2 Select font, style, size, or effects and click **OK.**

➤ To modify the style of an item:

1 Select an item and select **Format**, then **Style.**

A submenu of styles is opens.

2 Select a style:

- Bold
- Italics
- Underline
- Overline
- Double Overline

3 Click **Select.**

➤ To modify a number format:

1 Select a data cell and select **Format**, then **Number.**

The Number tab of the Properties dialog box opens.

2 **Select a Formatting Locale**, Choose a locale.

3 From the **Category** field, select a date type. Valid options are:

- All
- Number
- Currency
- Percentage
- Date
- Time
- Custom

The available number formats depends on the formatting locale and category. If you select a custom category, an edit box enables you can enter a custom number format.

4 Select a format.

5 Click **OK.**

➤ To justify text horizontally and vertically, choose a data cell and on the shortcut menu, and select **Justify**.

Available justification selections include:

- left
- center
- right
- top
- middle
- bottom

➤ To modify row or column borders, choose a row or column and on the shortcut menu, select **Borders**.

Available border styles include:

- none
- horizontal
- vertical
- horiz and vert (both horizontal and vertical)
- 3–D raise
- 3–D sink

➤ To modify the row height:

1 Select the row header, and on the shortcut menu, select **Row height**.

The Row height dialog box is displayed.

2 In the **Row height (pixels)** field, specify the height of the row in pixels.

In general the greater the number of pixels, the greater amount of detail in the rendered image.

3 Click **OK**.

➤ To modify the column width:

1 Select the column header, and on the shortcut menu, select **Column Width** field.

The Column width dialog box is displayed.

2 In the **Column width (pixels)** field, specify the width of the column in pixels.

In general the greater the number of pixels, the greater amount of detail in the rendered image.

3 Click **OK**.

➤ To auto-size the text in a cell horizontally, choose a row or a column, and select **Format** , then **Auto-Size Width**.

Auto-sizing by width is useful for displaying horizontal values without truncation.

➤ To auto-size the text in a cell vertically, choose a row or a column, and select **Format**, then **Auto-Size Height**.

Auto-sizing by height is useful for displaying vertical values without truncation.

➤ To display the name of the data layout values as corner labels, select **Format**, then **Corner Labels** and on the shortcut menu, specify the position in which to place the corner label.

Valid corner label positions are:

- None
- Column
- Row
- Both

➤ To export the CubeQuery section, select **Format**, then **Export Properties**.

For more information, see

CubeQuery Menu Command Reference

The following table is a quick reference to the commands and shortcuts.

Table 70 CubeQuery Menu Commands

Command	Description	Keyboard Shortcut	Shortcut Menu	Query Menu	Double-click Enabled
Retrieve Dimensions	Refreshes the dimension values in the Catalog	F9		*	
Add to Rows	Adds selected item to the Rows pane in the data layout		*	*	
Add to Columns	Adds selected item to the Columns pane in the data layout		*	*	
Add to Filters	Adds the selected item to the Filters pane in the data layout		*	*	
Drill Down	Retrieves more detail from the selected member		*	*	*

Command	Description	Keyboard Shortcut	Shortcut Menu	Query Menu	Double-click Enabled
Drill Up	Drills up to less dimension detail for the selected member		*	*	*
Drill Next	Retrieves data for the children of the selected member or members		*	*	
Drill Bottom	Retrieves data from the lowest level descendants of the selected member		*	*	
Drill All Descendants	Retrieves all descendants of the selected member		*	*	
Drill Sibling	Retrieve data for the siblings of the selected member or members		*	*	
Drill Same Level	Retrieves data for all members at the same level as the selected member or member		*	*	
Drill Same Generation	Retrieves data for all members of the same generation as the selected member or members		*	*	
Download to Results	Downloads the CubeQuery data set to an Results section for offline work			*	
Show as Chart	Charts the CubeQuery data set; automatically creates an Chart and Results sections			*	
Query Options	Accesses the Query Options dialog box, where you can set options for your CubeQuery			*	
Keep Only	Clears all other dimension member selections for the selected Dimension, leaving only the selected member		*		

Command	Description	Keyboard Shortcut	Shortcut Menu	Query Menu	Double-click Enabled
Remove Only	Clears a dimension member, removing it from the query result set.		*		
Suppress Missing Rows	Suppresses the return of data rows that contain only missing values. If one value is in the row, the row is not suppressed		*	*	
Suppress Missing Columns	Suppresses the return of data columns that contain only missing columns. If one value is in the column, the column is not suppressed		*	*	
Suppress Zero Rows	Suppresses the return of data rows that contain only zeros		*	*	
Suppress Zero Columns	Suppresses the return of data columns that contain only zeros		*	*	
Suppress Shared Members	Suppresses shared members from the query			*	
Row Height	Modifies the row height by pixel size.		*		
Column Width	Modifies the column width by pixel size:		*		
Cancel Query		Alt+End			

10

Pivot Tables

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.**

2 If the data layout is not visible, click **Data Layout** on the Section title bar.

3 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 Iten**.

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 71 Formatting Pivot Items

Format	Description
Font	To modify the font of an item in a pivot table, select the item and click **Font** on the Format menu or shortcut menu. When the Font tab of the Properties dialog box is displayed, apply any desired changes.
Style	To modify the appearance of an item in a pivot table, select the item and click **Style** on the Format menu or shortcut menu, and select the style. Styles include: bold, underline, italic formats, and so on.
Number	To modify the way numbers, dates, and currency are displayed in a pivot table, select the item and click **Number** on the Format menu or shortcut menu. When the Number tab of the Properties dialog box is displayed, apply any desired changes.
Justify	To modify the way an item is justified within a row or column in a pivot table, select an item, click **Justify** on the Format or shortcut menu. Select the type of justification.

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 268
- 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 [icon] 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 the [icon] 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 72   Pivot Data Functions

Function	Description
Sum	Returns sum of all values. This is the default setting.
Average	Returns average of all values.
Count	Returns number of values.
Maximum	Returns highest value.
Minimum	Returns lowest value.
% of Column	Returns surface values as a percentage of their respective column item.

Function	Description
% of Row	Returns surface values as a percentage of their respective row item.
% of Grand	Returns surface values as a percentage of all like values in the pivot table.
Increase	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.
% Increase	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.
Non-Null Average	Returns average of values; null values excluded.
Null Count	Returns number of null values.
Non-Null Count	Returns number of values; null values excluded.

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:

```
(Units % 50) +1
```

where "%" represents modulo (remainder) operator. In other words, the formula is defined as:

```
Integer remainder of ("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:

$12.5 = (8+12+10+20) \div 4$

An average of the surface values yields a result of:

$25 = (20+30) \div 2$

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 73   Example of Weighted Averages

Customer ID	Units Sold	Type of Store	Weight
1435	80	Electronics	8
1539	200	Computer	10
1634	60	Electronics	8
1213	900	Discount	2

➤ 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 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 you 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.

	City	Units Sold
1	Los Angeles	500
2	Los Angeles	400
3	Los Angeles	100
4	Los Angeles	100
5	Miami	300
6	Miami	200
7	Miami	100
8	Seattle	250
9	Seattle	50
10	Tacoma	100

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.

	Units Sold
Los Angeles	1,100
Miami	600
Seattle	300
Tacoma	100

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:

	Units Sold
Los Angeles	1,100
Miami	600
Seattle	300
Tacoma	100
Total	2,100

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:

	Units Sold	Units Sold * 2
Los Angeles	1,100	2,200
Miami	600	1,200
Seattle	300	600
Tacoma	100	200

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:

		Units Sold
CA	Los Angeles	1,100
FL	Miami	600
WA	Seattle	300
	Tacoma	100

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.)

	Month	Sales
1	Jan	10
2	Feb	15
3	Mar	17
4	Apr	20
5	May	22
6	Jun	20
7	Jul	25
8	Aug	27
9	Sep	30
10	Oct	35
11	Nov	37
12	Dec	40

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).

	Month	Sales
1	Jan	10
2	Feb	15
3	Mar	17
4	Apr	20
5	May	22
6	Jun	20
7	Jul	25
8	Aug	27
9	Sep	30
10	Oct	35
11	Nov	37
12	Dec	40

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.

- **Minimum**—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:

	Month	Sales
1	Jan	10
2	Feb	15
3	Mar	17
4	Apr	20
5	May	22
6	Jun	20
7	Jul	25
8	Aug	27
9	Sep	30
10	Oct	35
11	Nov	37
12	Dec	40

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

```
(500 + 400 + 100 + 100) / 4 = 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 263) the Pivot section display from above changes to:

	Month	Sales
1	Jan	10
2	Feb	15
3	Mar	17
4	Apr	20
5	May	22
6	Jun	20
7	Jul	25
8	Aug	27
9	Sep	30
10	Oct	35
11	Nov	37
12	Dec	40

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:

	Units Sold	Average Of Units Sold	Count Of Units Sold	Max Of Units Sold	Min Of Units Sold
Los Angeles	1,100	275	4	500	100
Miami	600	200	3	300	100
Seattle	300	150	2	250	50
Tacoma	100	100	1	100	100
Total	2,100	210	10	500	50

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

```
(500 + 400 + 100 + 100 + 300 + 200 + 100 + 250 + 50 + 100) / 10
= 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:

	Units Sold	Average Of Units Sold	Count Of Units Sold	Max Of Units Sold	Min Of Units Sold
Los Angeles	1,100	1,100	1	1,100	1,100
Miami	600	600	1	600	600
Seattle	300	300	1	300	300
Tacoma	100	100	1	100	100
Total	2,100	525	4	1,100	100

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):

	Units Sold	Average Of Units Sold	Count Of Units Sold	Max Of Units Sold	Min Of Units Sold
Los Angeles	1,100	275	4	500	100
Miami	600	200	3	300	100
Seattle	300	150	2	250	50
Tacoma	100	100	1	100	100
Total	2,100	210	10	500	50
Total - Average	210	210	210	210	210

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:

	Units Sold	Average Of Units Sold	Count Of Units Sold	Max Of Units Sold	Min Of Units Sold
Los Angeles	1,100	275	4	500	100
Miami	600	200	3	300	100
Seattle	300	150	2	250	50
Tacoma	100	100	1	100	100
Total	2,100	210	10	500	50
Total - Sum	2,100	2,100	2,100	2,100	2,100
Total - Average	210	210	210	210	210
Total - Count	10	10	10	10	10
Total - Max	500	500	500	500	500
Total - Min	50	50	50	50	50

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:

	Units Sold	Average Of Units Sold	Count Of Units Sold	Max Of Units Sold	Min Of Units Sold
Los Angeles	1,100	275	4	500	100
Miami	600	200	3	300	100
Seattle	300	150	2	250	50
Tacoma	100	100	1	100	100
Total	2,100	210	10	500	50
Total - Sum	2,100	2,100	10	2,100	2,100
Total - Average	210	210	10	210	210
Total - Count	10	10	10	10	10
Total - Max	500	500	10	500	500
Total - Min	50	50	10	50	50

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:

	Units Sold
Los Angeles	1,100
Miami	600
Seattle	300
Tacoma	100

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):

	Units Sold	Units Sold * 2	Sqrt Units Sold	ColMax Units Sold	Average Units Sold
Los Angeles	1,100	2,200	33.16625	500	210
Miami	600	1,200	24.4949	500	210
Seattle	300	600	17.32051	500	210
Tacoma	100	200	10	500	210

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.

	Units Sold	Units Sold * 2	Sqrt Units Sold	ColMax Units Sold	Average Units Sold
Los Angeles	1,100	2,200	33.16625	1,100	525
Miami	600	1,200	24.4949	1,100	525
Seattle	300	600	17.32051	1,100	525
Tacoma	100	200	10	1,100	525

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:

	Units Sold	Units Sold * 2	Sqrt Units Sold	ColMax Units Sold	Average Units Sold
Los Angeles	1,100	2,200	33.16625	500	210
Miami	600	1,200	24.4949	500	210
Seattle	300	600	17.32051	500	210
Tacoma	100	200	10	500	210
Total	2,100	4,200	45.82576	500	210

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:

	Units Sold	Units Sold * 2	Sqrt Units Sold	ColMax Units Sold	Average Units Sold
Los Angeles	1,100	2,200	33.16625	1,100	525
Miami	600	1,200	24.4949	1,100	525
Seattle	300	600	17.32051	1,100	525
Tacoma	100	200	10	1,100	525
Total	2,100	4,200	45.82576	1,100	525

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) 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 74   Pivot Menu Commands

Command	Description	Keyboard Shortcut	Shortcut Menu
Add Selected Items	Adds the selected item as a Column Label, Row Label, or Fact.		*
Remove Selected Items	Removes the selected item.	Del	*
Modify	Modifies the selected computed item.	Ctrl+M	*
Sort	Reorders the selected item by labels, by values, ascending, or descending.		
Add Totals	Adds the selected item to the Measures pane.		*
Add Computed Item	Adds a new data item derived from calculations performed on an item.		*
Add Cume	Adds cumulative totals to break totals by dimension and restarts them at each dimensional grouping.		*
Data Function	Applies a prebuilt data function to the selected item.		*
Drill Anywhere	Enables you to drill to any item.		*
Drillup	Returns the original view of data that you drilled.		*
Focus On Items	Updates the pivot table to include only the selected data.		*
Hide Items	Hides the selected item.		*
Show Hidden Items	Restores the selected hidden item.		*
Show All Items	Updates the pivot table to include all items.		*
Group Items	Groups the selected dimensions.	Ctrl+G	
Ungroup Items	Ungroups the selected dimension.	Ctrl+U	
Restore Name	Restores the original name of a renamed item.		*
Refresh Data	Updates the data according to the selected option. Select between After Process, When Section Displayed, Manually, or Refresh Now.		
Pivot Options	Enable surface values, true computed totals, and null facts in computed items.		

11

Charting Data

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 75   Chart Terminology

Term	Definition
Axes	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.
Values	Graphic indicators that represent data. Bar charts display values in either vertical or horizontal bars. Pie charts use wedge-shaped slices to represent values.
Plot Area	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.

Term	Definition
Planes	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.
Legend	An information box containing color-keyed labels used to identify different data values represented on a chart.
Grid lines	Straight horizontal and vertical lines arranged in scaled increments that provide calibrated guidelines for value interpretation.
Bar	A linear measure of a value used in bar charts.
Slice	A spacial measure of a value used in pie charts.
Chart top title	Text description related to the entire chart.
Chart subtitle	Additional text used to describe the chart.
Y axis label	Text description of the vertical quantity axis.
Y axis values	Text or numbers which indicate specific values along the y axis
X axis label	Text description of the horizontal quantity axis.
X axis values	Text or numbers which indicate specific values along the x axis.
Z axis label	Text description of the depth axis.
Z axis values	Text or number which indicate specific values along the z axis
Chart graphic	Central chart picture which contains Chart bars, background and also Chart axes.
Chart graphic border	Border which controls the position of the Chart's central bars, background and axes labels. The border can be resized and repositioned.
Chart border	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.
Inserted text string	Optional text strings that can be placed anywhere within the Chart border.
Right y axis label	Optional label that can be placed to the right of the right-hand vertical edger of the Chart graphic.
Right y axis values	Optional values that can be displayed on the right hand vertical edge of the Chart graphic
Bar value	Optional text which represents numeric value of the bar.

Table 76   Chart Terminology (Pie Specific)

Term	Definition
Pie slice	Individual y axis value representation for current x axis item.
Pie slice label	Text to identify x axis item.
Line to label	Line from pie slice label to pie slice to aid in slice identification.

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 77   Chart Data Layout

Data Layout Pane	Description
X Axis **Slice**	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.
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.

Data Layout Pane	Description
Fact Fact (Stack) Fact (Depth)	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. 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. For the **Fact (Depth)** pane, numeric data extends the length of the chart along the z-axis.

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) in to 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 sub-menu.

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 292
- "Scatter Charts" on page 294
- "Bubble Charts" on page 296
- "Two-dimensional Bar Charts" on page 298

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:

A	B	C	D
Degree of latitude	Tons per acre 95	Degree of latitude 98	Tons per acre 98
31	4750	31	5000
32	4800	32	5200
33	4900	33	5400
34	4850	34	5500
35	4800	35	5150
36	4900	36	5650
37	4450	37	5100
38	4250	38	4950

If the two data series a rendered in chart, the data is shown as:

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 are 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.

	A	B	C
1	Number of products	Sales	Market Share %
2	14	$12,200.00	15%
3	20	$60,000.00	23%
4	18	$24,400.00	10%

In the following bubble chart, the size of the bubble corresponds to the values in the third column of the sample data (Market share %):

Multiple data values can be plotted in the bubble chart.

Bubbles with zero size can rendered 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 are 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

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 331.

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 determines 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:

Date	Total Sales
1/1/2005	$2,583,200
2/1/2005	$3,551,700
6/1/2005	$2,011,210
7/1/2005	$1,149,210

A Line Chart that has no Time Aware axis, shows the data as follows:

Total Sales

The four ticks corresponding to the month are arranged uniformed 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:

Total Sales

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 ate/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 78   Reference Line Properties

Property	Description
Fixed	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.
(edit box)	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.

Property	Description
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 79 Reference Line Style Properties

Property	Description
Automatic	Enables the display of the default line color, width and pattern style.
None	Option is not available.
Custom	Enables you to select the line color, width and pattern style.
Color	Select to launch the palette and select another color. This option is only available for a custom line style.
Width	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.

Property	Description
Style	Select the pattern of the line. Available styles are: ● Solid ● Dash ● Dot ● DashDot ● DashDotDot This option is only available for a custom line style.

Reference Line Label Format Properties

Use the Reference Line Label Format dialog box to define line label (plot area) and legend text properties.

Table 80   Reference Line Label Format Properties

Property	Description
(Text Format for Plot Area or Legend)	Formats text for the plot area or legend from the drop-down. If the "Use the same format in plot area and legend field" is enabled, this option is disabled.
(Custom Format)	Specify a custom format for the plot area or legend. A custom format can combine constant text and generated strings for a statistical function or the value of the reference line. Constant text accepts the following tags: ● [FC]—fact name ● [VL]—value of reference line 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. 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. 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. 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. The complex tag, {; [DL] }inserts custom text as a delimiter. For example, the format "{[FN]([FC])}{ = [DL]}[VL]" may result in one of the following strings depending on conditions: ● Average(Sales) ● Average(sales) = $126000
Auto format	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.

Property	Description
Default	Show the default text: "<Function name>(<Fact name>) = <Value>".
Function	Shows the statistical function in auto-generated text, for example: Average.
Value	Shows the value of the reference line, for example: Value = 0.7348.

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 Nordfhein 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 3 D 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.

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 effect 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 81   Trend Line General Properties

Property	Description
Fact	Select the fact item on which to assign the trend line. Multiple trend lines cannot be assigned to a single fact item.
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 label	Enable to show the label text on the diagram.

Property	Description
Show in legend	Enable to show assigned text of the trend line and short line segment representing the actual color of the line in the legend.

Trend Line Style Properties

Use the trend Line Style Properties dialog box to define line styles.

Table 82   Trend Line Style Properties

Property	Description
Automatic	Enables the display of the default line color, width and pattern style.
None	Option is not available.
Custom	Enables you to select the line color, width and pattern style.
Color	Select to launch the palette and select another color. This option is only available for a custom line style.
Width	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.
Style	Select the pattern of the line. Available styles are: ● Solid ● Dash ● Dot ● DashDot ● DashDotDot This option is only available for a custom line style.

Trend Line Label Format Properties

Use the Trend Line Label Format dialog box to set up the line label and legend text.

Table 83   Trend Line Label Format Properties

Property	Description
(Text Format for Plot Area or Legend)	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.

Property	Description
(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{([FC])}—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.
Default	Shows the default text: "Trend(<Fact name>)"
Equation Type	Shows the equation name in auto-generated text, for example: linear.
	Note: As of Release 11.1.1 only the linear equation type is available.
R-squared	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.

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

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 he 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 84   Chart Data Functions

Data Function	Description
Sum	Returns sum of all values. This is the default setting.
Average	Returns average of all values.
Count	Returns number of values.
Maximum	Returns highest value.

Data Function	Description
Minimum	Returns lowest value.
% of Grand	Returns values as a percentage of all like values in the chart.
Non-Null Average	Returns average of values; null values excluded.
Null Count	Returns number of null values.
Non-Null Count	Returns number of values; null values excluded.

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 enables 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 shows 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:

➤ 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 ![icon].

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 332

- "Auto-Arranging Chart Pie Labels" on page 333

- "Applying Chart Color Schemes for Data Points" on page 334

Related Topics

"Customizing Chart Patterns and Labels" on page 332

Customizing Chart Patterns and Labels

In addition to the generic label axis properties that apply to all charts, you can change properties of individual charts 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:

- Standard chart color palette:

- 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:

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 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 **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 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 **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 85   Chart Menu Commands

Command	Description	Keyboard Shortcut	Shortcut Menu
Sort	Sorts the selected column values in ascending order (alphabetical or numeric).		*
Data Function	Applies a prebuilt data function to the selected item.		
Add Computed Item	Enables you to add a new data item derived from calculations performed on an item.		*
Modify Computed Item	Enables you to modify a computed item.		*
Remove Selected Items	Removes the selected items.	Del	*
Drill Anywhere	Enables you to drill to any item.		*
Drillup	Returns the original view of the data you drilled.		*
Focus On Items	Updates the chart to include only the selected items.		*
Hide Item	Hides the selected item from view.		*
Show Hidden Items	Restores the selected hidden item.		*
Show All Items	Updates the chart to include all items removed by focusing.		*
Group Items	Groups the selected items.	Ctrl+G	*
Ungroup Items	Ungroup the selected grouped item.	Ctrl+U	*

Command	Description	Keyboard Shortcut	Shortcut Menu
Restore Name	Restores the original name of a renamed item.		*
Refresh Data	Updates the data according to the selected option. Select between After Process, When Section Displayed, Manually, or Refresh Now.		
Properties	Opens the Properties dialog box for the chart or selected chart element.		*

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 86   General Chart Properties

Option	Description
Show title	Toggles the display of the chart title as entered in the text field.
Show subtitle	Toggles the display of a subtitle as entered in the text field.
Show legend	Toggles the display of the color-coded chart legend in the Contents pane.
Show border	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.
Show partial view indicator	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.
3-D objects	Toggles the three dimensional display of a chart object, including pie slices, chart bars, side planes, and the perspective view.
Print/PDF all views	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.
Auto Resize	Toggles the resizing of the graphic proportions within the Contents pane automatically whenever the application window is resized.
Smart Scaling	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.
Display all text with smart scaling	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.
Minimum Font Size	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 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.
Color Scheme for Data Points....	Launches the Choose Color Scheme dialog box from which you can select the chart color scheme for data points. Available color schemes are: - Standard—default color scheme used for charts created in Release 8.4 and later. - Legacy—color scheme used for charts in Release 8.3 and earlier - Custom—color scheme defined by the user. 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. Also see, "Applying Chart Color Schemes for Data Points" on page 334

Option	Description
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 87   Chart Labels Axis Properties

Option	Description
(Axis dropdown)	Sets the axis on which to apply options.
Show axis label	(X or Z) Toggles the display of a descriptive axis label as entered in the text field.
Show drill path in labels	(X or Z) Toggles the display of the drill path in the label and legend. By default, this option is enabled.

Option	Description
Time Aware	Check to enable the Time Aware Axis feature. When the feature cannot be used at the present time, the label: Time Aware (inactive now) is displayed. To review what conditions must be met to use the Time Aware Axis, see Time Aware Axis
Auto range	Check to calculate the time range of a Time Aware axis automatically. The minimum and maximum time range values are searched in the current data set. The time range is recalculated automatically when the data set is processed.
Min	Enter the minimum time range interval when calculating the time range of a Time Aware axis manually. Enter the minimum interval in the edit box. The date/time format used to display the values is the same one that you have for x axis labels or default format as in Default formats dialog. When the edit box is focused, it displays date/time in full format "mm/dd/yyy HH:MM:SS AM". You should enter values in full format, which allows you to enter any values regardless of current label format. This option is disabled if the Auto range option is enabled.
Max	Enter the maximum time range interval when calculating the time range of a Time Aware axis manually. Enter the maximum interval in the edit box. The date/time format used to display the values is the same one that you have for x axis labels or default format as in Default formats dialog. When the edit box is focused, it displays date/time in full format "mm/dd/yyy HH:MM:SS AM". You should enter values in full format, which allows you to enter any values regardless of current label format. This option is disabled if the Auto range option is enabled.
Auto frequency	Check to calculate tickmarks and labels automatically. If this option is disabled, values are not reset unless changed manually.
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.

Option	Description
Show values	Check to (X or Z) Toggles the display of individual labels along the axis.
Auto X Axis frequency and Show... fields (X Axis only)	(X Axis only) Custom—Sets the X Axis scale by toggling between auto-spacing, or manually specifying recurring intervals at which to display a limited number of values or tick marks.
Maximum bars displayed	Check to specify the number of bars to display per page when scrolling on a page. Enter the maximum number of bars in the edit box.
Display	Check to select the number of pages or date/time values when scrolling through a fixed number of intervals for a Time Aware axis. Available values include: ● Pages ● Year(s) per page ● Month(s) per page ● Week(s) per page ● Day(s) per page ● Hour(s) per page ● Minute(s) per page ● Second(s) per page

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 88   Chart Values Axis Properties

Option	Description
X Axis, Y Axis	Sets the axis (X or Y) for the axis scale parameter (see below). This option is available for scatter and bubble charts only.
Show left axis label	Toggles the display of a descriptive axis label as entered in the text field for the main value axis.
Show right axis label	Toggles the display of a descriptive axis label as entered in the text field for the secondary value axis.
Show tick marks at intervals	Toggles the display of tick marks along the axis.
Show values at intervals	Toggles the display of numeric scale values along the axis.
Show values at right	Toggles the display of numeric scale values at the right.

Option	Description
Logarithmic Scale	Plots the line chart with a logarithmic (base 10) values axis.
Baseline	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.
(Baseline Amount)	Specify the baseline amount.
Axis Scale (Left and Right)	Set the axis scale between selected minimum and maximum values, or click Auto to auto-assign the scale.
Interval	Enter an interval value to separate axis values, or click Auto to auto-assign the interval.

Bar Chart Properties

Bar chart properties control the general attributes for the various bar chart formats, including:

➤ 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 89   Bar Chart Properties

Option	Description
Show values on bars	Toggles the display of data values for all bars.
Cluster bars by items in "Y" axis	Uses the item assigned to the y axis to define a bar cluster. For example, if you have "Quarter" on the y axis, each cluster consists of four bars. (clustered bar charts only).
Cluster bars by items in "Z" axis	Uses the item assigned to the z axis to define a bar cluster. For example, if you have "Product Line" on the z axis, each cluster consists of a bar for each product line in your data. (clustered bar charts only).

Table 90   Bar Chart Properties

Option	Description
Show values on bars	Toggles the display of data values for all bar lines.
Show line values	Toggle the display of numeric values on each bar line.
Ignore null values for lines	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.
Shift Points to Left	Places line chart plot points at left of bars (bar-line charts only).
Shift Points to Center	Places line chart plot points centered on bars (bar-line charts only).
Stack bars along "Y" axis	Uses the item assigned to the "Y" axis to define a stacked bar line. For example, if you have "Quarter" on the "Y" axis, each stacked bar line consists of two lines.

Option	Description
Cluster bars along "Y" axis	Uses the item assigned to the "Y" axis to define a bar line cluster. For example, if you have "Quarter" on the "Y" axis, each cluster consists of four lines.

Area and Ribbon Chart Patterns Properties

Use Pattern tab to select the design of your Area and Ribbon charts.

Table 91   Area and Ribbon Chart Patterns Properties

Option	Description
Automatic	Sets the default fill pattern for the Area or Ribbon chart. The default pattern is solid.
None	Sets a transparent fill pattern for the Area or Ribbon chart. Only the borders of the Area or Ribbon chart element are displayed.
Custom	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.
Fill Pattern	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.
Foreground	Displays the part of the area or bar chart view nearest to the user that will be represented.
Preview	Displays the pattern as it will be displayed to the user when the settings have been applied.

Bar Chart Patterns Properties

Use the Bar Chart Patterns tab to apply a pattern to a bar chart element.

Table 92   Bar Chart Patterns Properties

Option	Description
Automatic	Sets the default fill pattern for the bar chart. The default pattern is solid.
None	Sets a transparent fill pattern for the bar chart. Only the borders of the chart element are displayed.
Custom	Enables you to select your own fill pattern for the bar chart. Valid patterns can be selected from the Fill Pattern pull-down list.
Fill Pattern	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.
Foreground	Toggles the display the part of the bar chart view nearest to the user that will be represented.
Preview	Displays the pattern as it will be displayed to the user when the settings have been applied.

Bar Chart Data Labels Properties

Use the Data Labels tab to toggle data labels on bar chart elements.

Table 93   Bar Chart Data Labels Properties

Option	Description
None	Suppresses the display of the numeric value of each bar chart element in a data label.
Show values	Displays the numeric value of each bar chart element in a data label.
Inside top	Inserts the data label at the top of the bar.
Inside middle	Inserts the data label in the middle of the bar.
Inside bottom	Inserts the data label at the bottom of the bar.

Line Chart Patterns Properties

Use the Line Chart Patterns tab to apply a pattern to elements and markers.

Table 94   Line Chart Patterns Properties

Option	Description
Automatic	Sets the default color, width and style pattern for the line.
None	Sets a transparent color, width and style for the line.
Custom	Enables you to select your own color, width and style pattern.
Color	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.
Width	Sets the width (weight) of the line. To select a different width, click another point size from the pull-down list.
Style	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
Preview	Displays the line or marker pattern as it will be displayed to the user when the settings have been applied.

Table 95   Line Chart Patterns Properties

Option	Description
Automatic	Sets the defaults color, size, border and fill color pattern for the marker.
None	Sets a transparent color, width and style for the line for the marker.
Custom	Enables you to select your own color, width and style pattern for the marker.
Style	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

Option	Description
Size	Sets weight of the marker. To select a different weight, click a point size from the another point size from the pull-down list.
Border Color	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.
Fill Color	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.

Line Chart Axis Properties

Use the Axis tab to plot values on the primary or secondary axis.

Table 96   Line Chart Axis Properties

Option	Description
Primary (left) axis	Toggles to plot (locate) values based on the primary or left axis.
Secondary (right) axis	Toggles to plot (locate) values based on the secondary or right axis.

Line Chart Data Labels Properties

Use the Data Labels tab to toggle data labels for line chart elements.

Table 97   Line Chart Data Label Properties

Option	Description
Show values	Displays the numeric value of each bar chart element in a data label.
Ignore Null Values	Toggle to include null values in the chart. Null values disrupt the line, area or ribbon.
Above	Inserts the data label above the marker.
Below	Inserts the data label below the marker.
Left	Inserts the data label to the left of the marker.
Right	Inserts the data label to the right of the marker.

Line Chart Properties

Line chart properties control the general attributes of the line chart.

Table 98   Line Chart Properties

Option	Description
Ignore Null Values	Toggle to include null values in the chart. Null values disrupt the line, area or ribbon.
Fill area under ribbon	Toggles between ribbon and area charts, creating the latter by filling the area under a ribbon. This option is activated by default.

Pie Chart Patterns Properties

Use the Pie Chart Patterns tab to apply a pattern to a pie chart slice.

Table 99   Pie Chart Patterns Properties

Option	Description
Automatic	Sets the default fill pattern for the slice. The default pattern is solid.
None	Sets a transparent fill pattern. Only the borders of the chart element are displayed.
Custom	Enables you to select your own fill pattern. Valid patterns can be selected from the Fill Pattern pull-down list.
Fill Pattern	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.
Foreground	Toggles the display the part of the chart view nearest to the user that will be represented.
Preview	Displays the pattern as it will be displayed to the user when the settings have been applied.

Pie Chart Data Labels Properties

Use the Data Labels tab to toggle data labels on each pie chart element.

Table 100   Pie Chart Data Label Properties

Option	Description
Show labels	Toggles the display of pie slice labels.
Show values	Toggles the display of the numeric value of each pie slice.
Show percentages	Toggles the display of the percentage value of each pie slice in a selected number format.
Show lines to labels	Toggles the display a line running from a data label to a selected pie slice.
Show negative values as positive	Toggles the representative display of negative values as pie slices, as opposed to not including them.

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.

4   Click the fill color and select a color from the color palette.

5   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

quarter. In this example, the divisions within the Quarter category (Q1, Q2, Q3, and Q4) would be represented vertically, together reaching to the total sum for the year.

➤ 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).

➤ 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 Drag two value items from the Contents pane to the z pane (which is the default value pane) in the data layout.

Note: To change the colors of each stack item, change the Legend to the z axis. This feature enables you to stack item and select a particular color from the Properties dialog box.

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.

12

Report Designer

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 of 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*.

❍ **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 I**nsert 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%.

Tool	Name	Description
	Rulers	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.
	Grid	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.
	Design Guides	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.

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.

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.

Expression Line Component	Description
Data Function ▼	Shows available data functions that can be applied to table columns.
✕	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.

➤ To toggle the Expression line, select Σ^+_x Expression .

➤ To add a expression:

1 Select Σ^+_x 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 101   Report Expression Syntax

Report Element	Syntax
Table Dimension	`CurrBreak.Value["Column Name"]` "Column Name" refers to the dimensional column name. The column name can be modified.

Table Fact	`Tables["Section"].Columns`
	
	`["Column Name"].Sum[currBreak]`
	
	"Section Name" refers to either the Results or Table section. Column Name refers to the fact column name. Sum refers to the return of underlying values and can also be an applied Data functions such as `Avg`, `Max`, and `Min`.

Group Label	`CurrBreak.Value["Column Name"]` "Column Name" refers to the dimension or fact column name.

Group Fact	`Tables["Section"].Columns`
	
	`["Column Name"].Sum[currBreak]`
	
	"Section Name" refers to either the Results or Table section. "Column Name" refers to the fact column name.
	
	To apply a break at the parent level (the highest level), use the following syntax:
	
	`Tables["Section Name"] .Columns`
	
	`["Column"].Sum[currBreak. Parent]`

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.

```
"Total: " + Format (Tables("Results").Columns("Amount").Sum(currBreak), "#,##0")
```

1996	Total: 1,439,047

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%")
```

Quarter 1	23%

If you plan to use this feature, add the labels in a computed field. For information on how to add a computed field, see "Adding Computed Items" on page 5-34.

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.

➤ To create a basic report:

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

Table Column Formatting Options

The following table lists the column formatting options available in Interactive Reporting.

Table 102   Table Column Formatting Options

Options	Description
Changing the Column Name Heading (Title)	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.
Hide/Show Column Name Heading (Title)	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.

Options	Description
Remove Column	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.
Hide/Show Column Total	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.
	This option is available only for fact columns.
Select Column	Click anywhere inside the column.
Suppress Duplicates	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.
Text Wrap	To wrap text within a column, select the column and select **Format**, then **Text Wrap**.
Sort Ascending/ Descending	To sort column values, click **Sort Ascending** or **Sort Descending** on the Table menu, shortcut menu, or standard toolbar.
Reordering Columns	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.

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.

➤ 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.

Item to Insert	Steps to Take
text	Select **Text Label** from the Graphics folder, and drag the text label to the header/footer area.
lines, rectangles, or oval graphic	Select the graphic from the Graphics folder and drag it to the report header/footer area, or select **Report**, then Insert Graphic.
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 ofInteractive 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.
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**.
common report values (such as page number or page total)	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.

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:

```
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.

Item to Insert	Steps to Take
text	Select **Text Label** from the Graphics folder, and drag the text label to the header/footer area.
lines, rectangles, or oval graphic	Select the graphic from the Graphics folder and drag it to the report header / footer area, or select **Report** , then **Insert Graphic**.
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 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 ofInteractive 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 that any Chart/Pivot dropped into a header/footer that is "owned" by data is focused by that piece of data.
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**.
common report values (such as page number or page total)	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.

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:

Working with Graphic Elements

Interactive Reporting provides a complete set of graphic elements to assist you in designing presentation-quality reports (see Table 103, "Report Section Graphic Elements ," on page 367). 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 ofInteractive 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.

➤ 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 103   Report Section Graphic Elements

Element	Description
Line	Creates a line that you can rotate.
Hz Line	Creates a horizontal line.
Vt Line	Creates a vertical line.
Rectangle	Creates a rectangle.
Round Rectangle	Creates a rectangle with rounded corners.
Oval	Creates an oval.
Text Label	Creates a text label that you can use as a caption.
Picture	Enables you to insert bitmaps (`.bmp` extension).
Name	Sets the name of the object.
Visible	Enables the object to be visible during run mode. Use this to hide objects.
Locked	Locks an object. Use this to lock objects so they cannot be changed.

➤ 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 104, "Report Section Fields," on page 367). 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 104   Report Section Fields

Field Name	Description
Field	Inserts a computed field.
Query Filter	Inserts a selected query filter.
Result Filter	Inserts a selected result filter.
Query SQL	Inserts the last SQL sent to the database when you process a query.

Field Name	Description
Page Number	Inserts a page number.
Number of Pages	Inserts the total number of pages.
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 EPM 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/DDYYYY 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/DDYYYY 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:

Report Element	Repeats
Report Group Header	Every Header band
Report Body	Every Body band
Report Header	Very first page
Report Footer	Very last page
Page Header	Every page
Page Footer	Every page

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 370
- "Using Grids" on page 370
- "Using Design Guides" on page 370

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 371
- "Specifying Page Margins" on page 371
- "Setting Up Page Columns" on page 372

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:

Type of Measurement	Size
Inches	.5
Pixel	.48
Centimeters	1.28

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. Column layout is applied by page and the default layout is one per report. You can set up any number of columns in the body of the report as you want.

➤ 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:

Type of Column Measurement	Column Number(s)	Size
Inches	1	8 "
Inches	2	3.75"
Inches	3	2.33"
Inches	4	1.63"
Spacing		0.5"
Pixels	1	768 px
Pixels	2	360 px
Pixels	3	224 px
Pixels	4	156 px
Spacing		48 px
Centimeters	1	20.33 cm
Centimeters	2	9.53 cm
Centimeters	3	4.13 cm
Centimeters	4	1.27 cm
Spacing		1.27 cm

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

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 *Hyperion System 9 BI+ Interactive Reporting Object Model and Dashboard Development Services Developer's Guide, Volume 1: Dashboard Design*.

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.

mag

USA

Store Manager	Amount Sales	Level 1 Bonus	Level 2 Bonus
Lee	6,315,561.82	2,000	
Wayne	32,713,875.41	2,000	3,000
White	7,245,879.45	2,000	
Wilson	6,353,954.04	2,000	
Wah	7,266,260.34	2,000	
Eadie	1,359,464.58		
	61,254,995.65	**2,000**	**3,000**

Level 1 Bonus JavaScript syntax:

Expression ☒

Data Function ▼ `if (Tables("Results").Columns("Level_1_Bonus").Sum(currBreak) >=5000000) (2000)`

Level 2 Bonus JavaScript syntax:

Expression ☒

Data Function ▼ `if (Tables("Results").Columns("Level_1_Bonus").Sum(currBreak) >=10000000) (3000)`

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 105 Prebuilt Data Functions in Report section

Data Function	Description
Column Name	Returns the table column name.
Sum	Returns sum of underlying values. This is the default data function.
Average	Returns average of underlying values.
Count	Returns number of underlying values.
Maximum	Returns highest of underlying values.

Data Function	Description
Minimum	Returns lowest of underlying values.
% of Category	Returns group total percentage of the value.
Null Count	Returns number of nulls among underlying values.
Non-Null Count	Returns number of underlying values; null values are excluded.
Non-Null Average	Returns average of underlying values; null values are excluded.

➤ 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 An average of the surface values yields the results $25 = (20 + 30) / 2$.

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:

Results 1	Region	Sales		Results 2	Region	Units
	East	10			East	100
	West	20			West	200
					South	400

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:

Region	Sales	Units
East	10	100
West	20	200

- 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 106   Report Formatting Options

Format	Description
Number	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.
Style	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
Alignment / Justify	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.
Font	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.
Keep Together	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.
Keep With Next	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.

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 107   Report Menu Commands

Command	Description	Keyboard Shortcut	Shortcut Menu
Design Guides	Toggles the display of design guides.		
Grid	Toggles the display of grid lines.		
Rulers	Toggles the display of rulers.		
Section Boundaries	Toggles the display of section boundaries (bands).		
Page Margins	Toggles the display of page margins		
Sort	Sorts the selected column values in ascending order (alphabetical or numeric).		*
Data Function	Applies a prebuilt data function to the selected item.		*
Add Computed Item	Opens the Insert Computed Item dialog box.		*
Break Total	Opens the Insert Break Total dialog box.		*

Command	Description	Keyboard Shortcut	Shortcut Menu
Focus On Items	Updates the report to include only the selected data.		*
Hide Item	Hides the selected item from view.		*
Show Hidden Items	Restores the selected hidden item.		*
Show All Items	Updates the report to include all items removed by focusing.		*
Insert Table	Inserts a blank table in the report.		*
Insert Field	Inserts a blank field in the report.		
Insert Predefined Field	Inserts a predefined field in the report.		
Insert Graphic	Inserts a vector graphic in the report.		
Picture	Inserts a picture in the report.		
Remove Selected Items	Removes the selected item from the report.	Del	*
Headers and Footers	Toggles the display of report and page headers and footers.		
Report Setup	Opens the Report Page Setup dialog box for adjusting page size, margins, and columns.		

13

Using Filters

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:

- ()—Encloses suboperations.

- **Var**—Indicates a variable.

- **AND**—Retrieves data that meets both condition.

- **OR**—Retrieves data that satisfies either of two conditions.

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.

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 108 Logical Operators for Defining Filters

Operator	Description
Equal (=)	Retrieves records where the filtered item equals the specified value(s).
Not Equal (<>)	Retrieves records where the filtered item does not equal the specified value(s).
Less Than (<)	Retrieves records where the filtered item is less than the specified value(s).
Less or Equal (<=)	Retrieves records where the filtered item is no greater than the specified value(s)
Greater Than (>);	Retrieves records where the filtered item is greater than the specified value(s).
Greater or Equal (>=);	Retrieves records where the filtered item is no less than the specified value(s)
Begins With	Retrieves records where the filtered item begins with the specified value(s) up to and including the end value.
Contains	Retrieves records where the filtered item contains the specified value(s) regardless of location.
Ends With	Retrieves records where the filtered item ends with the specified value(s).
Like (with wildcards)	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
Is Null	Retrieves records where the filtered item has no value; for example a field in which no data has been entered.
Between	Retrieves records where the value of the filtered item lies between (and does not equal) the specified values.
Not (with operator)	Negates the operator it precedes, reversing the results of the equation

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 *City = Cleveland* and *SUM(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.

- **Prompt**—Supply explanatory comments or instructions.
- **Values**—Disable Show Values to reduce database hits or remove the Custom SQL or Custom Values option.
- **Options**—Disable elements and lock in operators.

3   Click **OK**.

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 men.

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 109   Customize Filter

Feature	Description
Title	Add a title or text to the Filter to instruct users on setting the filter, or describe the value.
Prompt	Supply explanatory comments or instructions.
Values	Disable the Show Values option to reduce database hits, or remove the Custom SQL or Custom Values options.
Option	Disable elements and lock in operators.

14

Computed Items

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 `{Units * Amount = Revenue}`

- Build logical expressions `{If Score > =50 Then "Pass" Else "Fail"}`

- Build mixed expressions `{Tax = Revenue *.35)`

- Apply functions `{(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 110 Data Types and Specifications

Data type	Specification
Automatic	A data type is determined automatically given the data type of the reference items and the computations performed.
Byte	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.
Date	Calendar date in server default format (typically mm/dd/yy).
Integer (16-bit)	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.
Integer (32-bit)	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.
Packed Real	Real numbers packed for use with EDA middleware. The results in Interactive Reporting are the same as real numbers.
Real	Decimal numbers up to 5 positions right of the decimal.
String	Text strings to a maximum length of 256 characters.
Time	Time in format set by user preference.
TimeStamp	Date/time combination in format set by user preference.

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 results 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:

Store	Amount	Computed	Computed2
Computer Central	1,982.66667	1,982.66667	1,982.66667
Total Computer Cen	**1,982.66667**		
Computer City	1,883.53333	1,883.53333	1,883.53333
Total Computer City	**1,883.53333**		
Computer Super Sto	1,784.4	1,784.4	1,784.4
Total Computer Sup	**1,784.4**		
Computer Town	892.2	892.2	892.2
Total Computer Tow	**892.2**		
Computer World	1,784.4	1,784.4	1,784.4
Total Computer Wor	**1,784.4**		
Discount Electronics	892.2	892.2	892.2
Total Discount Elect	**892.2**		
Grapevine Periphera	1,189.6	1,189.6	1,189.6
Total Grapevine Per	**1,189.6**		
Hi-Tech Mart	1,784.4	1,784.4	1,784.4
Total Hi-Tech Mart	**1,784.4**		
Power Computers	1,487	1,487	1,487
Total Power Compu	**1,487**		
Wolfe"s Disc. Electro	2,200.76	2,200.76	2,200.76
Total Wolfe"s Disc.	**2,200.76**		
	15,881.16	**15,881.16**	**1,625.78667**

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.

		Units	Computed
CA	Burbank	2,395	46
	Glendale	990	41
	Oakland	910	11
	San Diego	7,400	1
	Westwood	695	46
	Total	12,390	41
NY	New York	4,640	41
	Rochester	1,615	16
	Total	6,255	6
Total		18,645	46

In this example, the *Sum* data function has been applied. The "Computed" column is defined by the following formula:

```
(Units % 50) +1
```

where "%" represents modulo (remainder) operator. In other words, the formula is defined as:

```
Integer remainder of ("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.

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.

Option	Description
Use Surface Values	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.

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.

Arithmetic Operators

Arithmetic operators take numerical values (either logical or variables) as their operands and return a single numerical value.

Table 111 Arithmetic Operators

Operator	Name	Description
+	Add	Used at both the server level and the local metatopic level for all sections.
-	Subtract	Used at both the server level and the local metatopic level for all sections.
*	Multiply	Used at both the server level and the local metatopic level for all sections.
/	Divide	Used at both the server level and the local metatopic level for all sections.
(	Begin suboperations	Used at both the server level and the local metatopic level for all sections.
)	End suboperations	Used at both the server level and the local metatopic level for all sections.
++	Increment	Used at both the server level and the local metatopic level for all sections except the Query section.
--	Decrement	Used at both the server level and the local metatopic level for all sections except the Query section.
Mod (%)	Modulus The modulus operator returns the remainder of dividing var1 by var2. For example, 5% 4 returns 1.	Used only at the local metatopic level.

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, *var1* has been assigned the value 3, and *var2* has been assigned the value *4*.

Table 112   Comparison Operators (Local Metatopic Level)

Operator	Description
==	Returns true if the operands are equal, for example, 3 == var1
!=	Returns true if the operands are not equal, for example, var1 !=4
<	Returns true if the left operand is less than the right operand, for example, var1 < var2
<=	Returns true if the left operand is less than or equal to the right operand, for example, var <=var2, var2 <=5
>	Returns true if the left operand is greater than the right operand, for example, var2 > var1
>=	Returns true if the left operand is greater than or equal to the right operand, for example, var2>=, var1>=2

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.

Table 113   if...else Statement

Statement	Description
if...else	if executes a set of statements if a specified condition is true. The specified condition may be another statement and can include other nested if statements. Braces, {}, must enclose multiple statements. If the condition is false, another set of statements can be executed if the optional else statement has been included in the script. A sample *if ... else* statement looks likes this: ```if (condition) {

statements1

}

else {

statements2

}``` |

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 114   Logical Operators (Local Metatopic Level)

Logical Operator	Description
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:
	`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:
	`: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:
	`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 412

● "Data Functions" on page 420

● 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:

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 115   Conditional Scalar Functions

Function	Description
Decode (c,exp,val, exp,val...def)	Compares value of item *c* to one or more expressions *exp*, and returns the value val matched to each expression, or a default *def*.
	Example:
	`decode (region, 'South America','Priority 1', 'Asia','Priority 2','Europe','Priority 3',null)`
	substitutes priority values for the specified regions and leaves other regions null.

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 116   Date Scalar Functions

Function	Description
AddMonths (d,n)	Adds *n* months to date *d*. **Example:** AddMonths ('5/13/99',4) = 9/13/99
DayOfMonth (d)	Returns the day of month for date *d*. **Example:** `DayOfMonth ('11/02/99') = 2`
LastDay (d)	Returns date of the last day of the month containing date *d*. **Example:** `LastDay ('12/6/9') = 12/31/99`
MonthsBetween (d1,d2)	Returns the number of months between dates *d1* and *d2* as a real number (fractional value). **Example:** `MonthsBetween ('12/5/99','5/6/99') = 6.9677`
NextDay (d,s)	Returns the date of the first weekday *s* after date *d*. If s is omitted, add one day to *d*. **Example:** `NextDay ('12/16/99',"Monday") = 12/22/99`
Sysdate ()	Returns the current system date and time for each record in item *c*. **Example:** `Sysdate() = 2/11/96 19:54:36`

Function	Description
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', "mmmm-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 117   Math Scalar Functions

Function	Description
Abs (n)	Returns the absolute value of number *n*. **Example:** `Abs (-3) = 3`

Function	Description
Atn (n)	Returns arc tangent of number *n* radians. **Example:** `Atn (1) = .7854`
Ceil (n)	Returns the smallest integer value greater than or equal to number *n*. **Example:** `Ceil (5.6) = 6`
Cos (n)	Returns cosine of number *n* radians. **Example:** `Cos (0.5) = .8776`
Cosh (n)	Returns hyperbolic cosine of number *n* radians. **Example:** `Cosh (0.5) = 1.1276`
Count (c)	Returns the number of row values in *c* (including nulls). **Example:** `Count (units) = tally of rows in units`
Exp (n)	Returns *e* (2.718) raised to exponential power *n*. **Example:** `Exp (4) = 54.598`
Max (a,b)	Returns the larger of items *a* and *b* for each new value. **Example:** `Max (7, 10) = 10`
Min (a,b)	Returns the smaller of items *a* and *b* for each new value. **Example:** `Min (7, 10) = 7`
Mod (n,m)	Returns the integer remainder of number *n* divided by number *m*. If *m* is larger, the default value is *n*. **Example:** `Mod (6,2) = 0`
Power (n,m)	Returns number n raised to exponential power *m*. **Example:** `Power (10,5) = 100,000`

Function	Description
Round (n,m)	Returns *n* rounded to m decimal places. If *m* is omitted, round to 0 decimal places. **Example:** `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:** `Sign (-4) = -1`
Sin (n)	Returns sine of number *n* radians. **Example:** `Sin (86) = -0.924`
Sinh (n)	Returns hyperbolic sine of number *n* radians. **Example:** `Sinh (.5) = .5211`
Sqrt (n)	Returns square root of number *n*. **Example:** `Sqrt (81) = 9`
Tan (n)	Returns tangent of number *n* radians. **Example:** `Tan (30) = -6.405`
Tanh (n)	Returns hyperbolic tangent of number *n* radians. **Example:** `Tanh (45) = 1`
Trunc (n,m)	Returns number *n* truncated to number m decimal places. The default value for *m* is 0. **Example:** `Trunc (56.0379,2) = 56.03`

Numeric Scalar Functions

Table 118   Numeric Scalar Functions

Function	Description
Avg (numbers, break_col, break_value)	Returns the average (arithmetic mean) of values in a number column. The average includes null values when calculating the arithmetic mean. Avg example
AvgNonNull (numbers, break_col, break_value)	Returns the average (arithmetic mean) of values in a number column, excluding null values. AvgNonNull example
chr (n)	Returns string converted from ASCII numeric code *n*. **Example:** `Chr (65) = A`
ColMax (numbers, break_col, break_value)	Returns the largest value in a column of numbers. ColMax example
ColMin (numbers, break_col, break_value)	Returns the smallest value in a column of number. ColMin example
Count (numbers, break_col, break_value)	Counts and returns the number of rows in a column. Count example
CountDistinct (numbers, break_col, break_value)	Counts and returns the number of values in a column. CountDistinct example
CountNonNull (numbers, break_col, break_value)	Counts the number of rows in a column. CountNonNull example
CountNull (numbers, break_col, break_value)	Counts the number of rows in a column that contains null values. CountNull example
Cume (numbers, break_col)	Returns a cumulative running total for each value in a column of numbers. Cume example
Next (c)	Returns the next row value of the referenced item *c*.
Prior (c)	Returns the prior row value of the referenced item *c*.
Sum (numbers, break_col, break_value)	Returns the total of a column of numbers.

Picture Functions

Table 119   Picture Function

Function	Description

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:
	
	`if (Amount_Sales >= 10000) {Picture ("C:\\graphics\\smile.gif" )}`
	`        else{Picture("C:\\graphics\\crying.gif" )}`

Statistical Scalar Functions

Table 120   Statistical Scalar Functions

Function	Description
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 nth 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

Function	Description
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 121   String Scalar Functions

Function	Description
Ascii (s)	Returns an ASCII code of the first symbol in the string. *s*. **Example:** `Ascii ("AZ") = 65`
Concat (s1, s2)	Returns text strings *s1* and *s2* concatenated. **Example:** `Concat ("interactive","reporting") = interactivereporting`
Initcap (s)	Returns string *s* with the first letter of each word capitalized, and remaining characters in lower case. **Example:** `Initcap ("santa fe") = Santa Fe`

Instr (s1,s2,n,m)	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.
	
	Examples:
	
	`Instr ("Mississippi",'s',5,2) = 7`
	
	`Instr ( City, 'a', -2, 1 )`

Length (s)	Returns character count of string *s*. **Example:** `Length ("Pittsburgh") = 10`
Lower (s)	Returns string *s* in lower case. **Example:** `Lower ("CD-Rom") = cd-rom`

Function	Description
Ltrim (s1,s2)	Trims string *s1* from the left, up to the first character not included in string *s2*. **Example:** `Ltrim ("Mr. Jones", "Mr. ") = Jones`
Replace (s1,s2,s3)	Returns string item *s1* with all occurrences of string *s2* replaced by string *s3*. The default for *s3* deletes each occurrence of *s2*. **Example:** `Replace (customer,"Mrs.","Ms.") = replaces Mrs. with Ms. for all values of customer containing 'Mrs.'`
Rtrim (s1,s2)	Trims column string *s1* from the right, up to the first character not included in string *s2*. **Example:** `Rtrim ("Columbus, OH",", OH") = Columbus`
Substr (s,n,m)	Returns a portion of string *s*, *m* characters long, beginning at numeric position *n*. The default action for *m* includes all remaining characters. **Example:** `Substr ("312/989-9989","1","3") = "312"`
Translate (s1,s2,s3)	Returns string *s1*, with each character contained in string *s2* replaced by the corresponding characters in string *s3*. **Example:** `Translate ("12/5/97 3:48:22","/:","-|") = 12-5-97 3|48|22`
Upper (s)	Returns string *s* in upper case. **Example:** `Example: Upper ("st.") = ST.`

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.

Function	Description	Availability
None	Returns unaggregated values as stored in the database. This is the default in Query.	Query
Sum	Returns sum of underlying values. This is the default in Results and report sections.	All
Average	Returns average of underlying values.	All
Non-Null Average	Returns average of underlying values; null values excluded.	Pivot, Chart, Report
Minimum	Returns lowest of underlying values.	All
Maximum	Returns highest of underlying values	All
Count	Returns number of underlying values.	All
CountDistinct	Returns the number of distinct values in a column. This function is not supported by all database servers.	Query
Null Count	Returns number of nulls among underlying values.	Pivot, Chart, Report
Non-Null Count	Returns number of underlying values; null values excluded.	Pivot, Chart, Report
Standard Deviation	Returns standard deviation of values. This function is not supported by all database servers.	Query
Variance	Returns variance of values. This function is available through Oracle servers only.	Query
Weight	Use for computing weighted items in Pivot reports.	Query
% of Column	Returns sum of all underlying values as a percentage of their respective surface column.	Pivot
% of Row	Returns sum of underlying values as a percentage of their respective surface row.	Pivot
% of Grand	Returns sum of underlying values as a percentage of all surface values in the report.	Pivot, Chart
% of Category	Returns group total percentage of the selected value.	Report
Increase	Calculates the increase between the previous two rows or columns.	Pivot
%Increase	Calculates the percentage increase between the previous two rows or columns.	Pivot

Function	Description	Availability
Title	Returns column names	Report

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 422
- "Teradata Date Functions" on page 422
- "Teradata Miscellaneous Functions" on page 423
- "Teradata Numeric Functions" on page 423
- "Teradata String Functions" on page 423
- "Teradata System Functions" on page 424
- "Teradata Trigonometric Functions" on page 424

Teradata Conditional Functions

Function	Description	Return Type	Parameter
NullIf	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: `CASE WHEN scalar_expression_1=scalar_expression_2 THEN NULL ELSE scalar_expression_1 END`	String	String, comparison string
Coalesce	Returns the second argument if the first argument is null.	String	String 1, String 2

Teradata Date Functions

Function	Description	Return Type	Parameters
Extract	Return part from the given date	Long	Date part (Year, Month, Day, Hour, Second) Date value
Add Months	Returns the date with the months added	Date	Date, months to add

Teradata Miscellaneous Functions

Function	Description	Return Type	Parameters
BYTE	Returns the number of bytes contained in the specified byte string.	Long	Byte, varbyte or BLOB expression.
FORMAT	Returns the declared format for the named expression	String	Column expression
Title	Returns the title of an expression as it would appear in the heading for displayed or printed results.	Strong	Column expression
Type	Returns the data type defined for an expression.	String	Column expression

Teradata Numeric Functions

Function	Description	Return Type	Parameters
Logarithmic	Returns base 10 logarithm of given value.	Real	Numeric
Natural Logarithm	Returns a natural logarithm.	Real	Numeric
Null If Zero	Returns "null" if the argument is zero, otherwise, returns the argument.	Real	Numeric
Random	Returns random integer	Long	Lower limit, upper limit
Zero If Null	Returns "zero" if the argument is null, otherwise, returns the argument.	Real	Numeric

Teradata String Functions

Function	Description	Return Type	Parameter
Concat	Returns concatenated string.	String	String 1, String 2
CHAR2HEXINT	Returns the hexadecimal representation for a character string.		String expression
Index	Returns the starting position of string 2 inside string 1. Returns 0 if string 2 is not found.	String	String, Search string
Soundex	Returns a character string that represents the Soundex code for a string_expression.	String	String expression
Substring	Returns the substring of the string provided.	String	String, start position, length
Trim	Trims blanks from a given string.	String	Leading/Trailing/Both String to trim
VARGRAPHIC	Returns the VARGRAPHIC representation of the character data in a character_string_expression.	String	String expression

Teradata System Functions

Function	Description	Return Type	Parameter
Date	Returns current date.	Date	
Time	Returns current time.	Time	
User	Returns current user name.	String	

Teradata Trigonometric Functions

Function	Description	Return Type	Parameter
ACOS	Returns angle of given cosine value.	Real	Cosine value
ASIN	Returns the angle whose sine is the argument.	Real	Sine value
ATAN	Returns the angle whose tangent is the argument.	Real	Tangent value
ATAN2	Returns the arctangent of the given point.	Real	X and y coordinates
COSH	Returns the hyperbolic cosine of an argument.	Real	Angle
SINH	Returns the hyperbolic sine of an argument.	Real	Angle
TANH	Returns the hyperbolic tangent of an argument.	Real	Angle
ACOSH	Returns the inverse hyperbolic cosine of an argument.	Real	Angle
ASINH	Returns the inverse hyperbolic sine of an argument.	Real	Angle
ATANH	Returns the inverse hyperbolic tangent of an argument	Real	Angle

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

CSum (Cumulative Sum) Function

Table 122   CSum (Cumulative Sum) Function

Field	Description
Description	Accumulates a sum over an ordered set of rows, providing the current values of the SUM on each row
Syntax:	CSum(value_expression, sort_expression_list)
value_expression	A value_expression is a scalar numeric column expression for which a running sum is to be computed.
sort_expression_list	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).

MAvg (Moving Average) Function

Table 123   MAvg (Moving Average) Function

Field	Description
Description	Computes the moving average of a column using the current row and the preceding width-1 rows.
Syntax:	MAvg(value_expression, width, sort_expression_list)
value_expression	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.
width	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.
sort_expression_ list	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).

MDiff (Moving Difference) Function

Table 124   MDiff (Moving Difference) Function

Field	Description
Description	Returns the moving difference between the current row-column value and the preceding nth value. The moving difference is a common business metric used to compare activity for some variable in a current time period to the activity for the same variable in another time period at a fixed distance in the past.
Syntax:	MDiff(value_expression, width, sort_expression_list)
value_expression	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.

Field	Description
width	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.
sort_expression_ list	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).

MSum (Moving Sum) Function

Table 125   MSum (Moving Sum) Function

Field	Description
Description	Computes the moving sum of a column using the current row and the preceding *n-1* row.
Syntax:	MSum(value_expression, width, sort_expression_list)
value_expression	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.
width	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.
sort_expression_ list	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).

MLingreg (Multiple Linear Regression) Function

Table 126   MLingreg (Multiple Linear Regression) Function

Field	Description
Description	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 MLinreg function computes the regression using all the preceding rows.
Syntax:	MLingreg(value_expression, width, sort_expression_list)
value_ expression	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.
width	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.
sort_ expression_list	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.

Quantile Function

Table 127   Quantile Function

Field	Description
Description	Computes the quantile scores for the values in a column. A quantile is a generic interval of user-defined width.
Syntax:	Quantile(quantile_constant, sort_expression_list)
quantile_constant	A positive integer constant used to define the number of quantile partitions to be used.
sort_expression_list	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).
Quantile Value Range	0 through (Q-1) where Q is the number of quantile partitions specified by the quantile constant.

Rank Function

Table 128   Rank Function

Field	Description
Description	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.
Syntax:	Rank(sort_expression_list)
sort_expression_list	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).

Current_Timestamp Function

Table 129   Current Timestamp Function

Field	Description
Description	Returns the current system timestamp and current session Time Zone displacement.
Syntax:	Current_Timestamp(fractional_precision)
fractional_precision	An option precision range for the returned timestamp value. The value range is 0 through 6, inclusive. The default is 6.
Properties	Data type: TIMESTAMP WITH TIME ZONE Length: 12 Not nullable
Fields	YEAR, MONTH, DAY, HOUR, MINUTE, SECOND, TIMEZONE_HOUR, TIMEZONE_MINUTE

Qualify Function

Table 130   Qualify Function

Field	Description
Description	Filters results of a previously computed OLAP function according to user-specified conditions.
Syntax:	Qualify *search_condition*
Qualify	Represents a conditional clause in the SELECT statement.
search_condition	One or more conditional expressions that must be satisfied by the result rows. Aggregate operators with a Qualify clause can be used.
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 131   Sample Function

Field	Description
Description	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.
Syntax:	Sample(*fractional_description [count_description]*)
fractional_ description	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.
count_ description	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.
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 132   SampleID Function

Field	Description
Description	Identifies the sample to which a row belongs, distinguishing rows belonging to different samples specified in the SAMPLE clause of a SELECT statement.
Syntax:	SAMPLEID
Definition	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).
Where to Specify SAMPLEID	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.

ANSI SQL Functions

Interactive Reporting supports ANSI SQL functions, and windowing clauses for the data. Windowing or partions 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:

```
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 133   ANSI SQL Functions

Name	Description	Return Type	Parameters
CORR	Returns the Pearson product moment correlation coefficient of its argument fall all non-null data point pairs.	Real	numeric column expressions x and y
COVAR_POP	Returns the population covariance of its arguments for all non-null data point pairs.	Real	numeric column expressions x and y
COVAR_SAMP	Returns the sample covariance of its arguments for all non-null pairs.	Real	numeric column expression x and y
GROUPING	Returns a value that indicates whether a specified column in the result row was exclude	Integer	an item of a GROUP BY clause

Name	Description	Return Type	Parameters
KURTOSIS	Returns the distribution kurtosis of the column expression.	Real	numeric column expression
REGR_AVGX	Returns the mean of the *independent_column_ expression* for all non-null data pairs of the dependent and independent variable arguments.	Real	dependent and independent column expressions
REGR_AVGY	Returns the mean of the *dependent_column_ expression* for all non-null data pairs of the dependent and independent variable arguments.	Real	dependent and independent column expressions
REGR_COUNT	Returns the count of all non-null data pairs of the dependent and independent variable arguments.	Real	dependent and independent column expressions
REGR_INTERCEPT	Returns the intercept of the univariate linear regression line through all non-null data pairs of the dependent and independent variable arguments.	Real	dependent and independent column expressions
REGR_R2	Returns the coefficient of determination for all non-null data pairs of the dependent and independent variable arguments.	Real	dependent and independent column expressions
REGR_SLOPE	Returns the slope of the univariate linear regression line through all non-null data pairs of the dependent and independent variable arguments.	Real	dependent and independent column expressions
REGR_SXX	Returns the sum of the squares of the independent_ column_expression for all non-null data pairs of the dependent and independent variable arguments.	Real	dependent and independent column expressions
REGR_SXY	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.	Real	dependent and independent column expressions
REGR_SYY	Returns the sum of the squares of the dependent_ column_expression for all non-null data pairs of the dependent and independent variable arguments.	Real	dependent and independent column expressions
SKEW	Returns the skewness of the distribution of value_ expression.	Real	numeric column expression
STDDEV_SAMP	Returns the sample standard deviation for the non-null data points in value_expression.	Real	numeric column expression
VAR_POP	Returns the population variance for the data points in value_expression.	Real	numeric column expression
VAR_SAMP	Returns the sample variance for the data points in value_expression.	Real	numeric column expression
WIDTH_BUCKET	Returns the number of the partition to which value_ expression is assigned.	Integer	value expression, lower bound, upper bound, partition count
PERCENT_RANK	Returns the relative rank of rows for a value_ expression.	Real	

Name	Description	Return Type	Parameters
ROW_NUMBER	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.	Integer	
POSITION	Returns the position in string_expression_2 where string_expression_1 starts.	Integer	String expression 1, string expression 2
TRANSLATE	Converts a character string or character string expression from one server character set to another server character set.	String	String expression, conversion
CHARACTER_LENGTH	Returns the length of a string either in logical characters or in bytes.	Integer	String expression
OCTET_LENGTH	Returns the length of string_expression in octets.	Integer	String expression
CURRENT_DATE	Returns the current system date.	Date	
CURRENT_TIME	Returns the current system time.	Time	
CURRENT_TIMESTAMP	Returns the current system timestamp.	Timestamp	

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

Function						
Max	Ascii	Current Time	Month	Integer	Insert	Substr
Avg	Exp	Date	Monthname	Log10	Lcase	Ucase
Count	LTrim	Day	Quarter	Mod	Left	
Abs	RTrim	Dayname	Second	Rand	Length	
Floor	Sin	DayOfWeek	Time	Smallint	Locate	
Power	Cos	Dayofyear	Week	Trunc	Posstr	
Round	Tan	Days	Year	Char	Repeat	
Sqrt	Nullif	Hour	Ceil	Chr	Replace	

A SQL: (MSSQL) database loads these functions:

Function			
Sum	Sqrt	Grouping	Right
Min	Lower	Width_Bucket	Substring
Max	Upper	Row_Number	Str
Avg	Ascii	Current_Timestamp	User_name
Count	Exp	Rank	Suser_name
Abs	LTrim	Length	Dateadd
Floor	RTrim	Isnull	Datediff
Power	Sin	Ceiling	Datename
Round	Cos	Rand	Datepart
Sign	Tan	Reverse	Getdate

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 you 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**>,'dddd')**

   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.

	Amount Sales	Tax	Net Amount
1999	$114,266,340.20	$3,999,321.91	110,267,018.295
2000	$18,615,338.30	$651,536.84	17,963,801.464

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.

	Unit Sales	Mean Value	Median Value	Mode Value
1999	5272112	411	276	370
2000	884533	621	404	441

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.

		1999					
		Amount Sales	Entire Year	By Quarter	By Quarter and Product Line	For Q1	For Books
Q1	Books	$13,744,748.12	$8,908.96	$7,392.77	$8,816.39	7,392.769	10,745.692
	Music	$4,451,016.53	$8,908.96	$7,392.77	$5,098.53	7,392.769	10,745.692
	Videos	$2,533,558.55	$8,908.96	$7,392.77	$6,810.64	7,392.769	10,745.692
Q2	Books	$13,774,642.22	$8,908.96	$6,915.81	$8,383.84	7,392.769	10,745.692
	Music	$4,433,484.45	$8,908.96	$6,915.81	$4,721.50	7,392.769	10,745.692
	Videos	$2,608,450.82	$8,908.96	$6,915.81	$6,094.51	7,392.769	10,745.692
Q3	Books	$16,747,678.86	$8,908.96	$7,743.84	$9,092.12	7,392.769	10,745.692
	Music	$5,657,958.46	$8,908.96	$7,743.84	$5,601.94	7,392.769	10,745.692
	Videos	$3,141,276.30	$8,908.96	$7,743.84	$7,027.46	7,392.769	10,745.692
Q4	Books	$31,769,448.07	$8,908.96	$12,704.96	$15,634.57	7,392.769	10,745.692
	Music	$9,326,459.77	$8,908.96	$12,704.96	$8,060.90	7,392.769	10,745.692
	Videos	$6,077,618.05	$8,908.96	$12,704.96	$11,598.51	7,392.769	10,745.692

The following table shows how averages have been calculated in the figure above.

Table 134   Calculated Averages

Column	Numeric Function
Amount Sales	sum of Amount Sales by quarter and Product Line `= Sum (Amount_Sales)`
Entire Year	average purchase amount (Amount Sales) across all quarters and Product Lines `= Avg (Amount_Sales)`
By Quarter	average purchase amount (Amount Sales) in a specific quarter `= Avg (Amount_Sales, Quarter)`
For Quarter and Product Line	average purchase amount (Amount Sales) in a specific quarter for a Product line `= Avg data function applied to Amount_Sales column`
For Q1	average purchase size (Amount Sales) across all Product Lines for Q1 specifically `= Avg (Amount_Sales, Quarter, 'Q1')`
For Books	average purchase size (Amount Sales) across all Quarters for Books `=   Avg (Amount_Sales, Product_Line, 'Books')`

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.

	Amount Sales	Eightieth Percentile	Qualify
Redentor	365,131.171	7,258,107.987	
Robinson	1,982,942	7,258,107.987	
Saint	1,302,044.456	7,258,107.987	
Schmidt	4,067,065.456	7,258,107.987	
Schultz	808,609.729	7,258,107.987	
Shay	3,159,782.897	7,258,107.987	
Shibasaki	9,688,780.173	7,258,107.987	✔
Stuber	691,960.051	7,258,107.987	
Vilhena	731,976.471	7,258,107.987	
Wah	7,266,260.345	7,258,107.987	✔
Wayne	32,713,875.41	7,258,107.987	✔
White	7,245,879.451	7,258,107.987	
Wilson	6,353,954.035	7,258,107.987	

The second example identifies countries that make sales transactions under $10,000.

	Amount Sales	95% of Sales Amounts under the value
Argentina	$692,299.72	$12,879.02
Australia	$14,598,717.21	$33,025.46
Brazil	$1,097,107.64	$5,805.23
Canada	$5,711,431.51	$19,578.64
France	$5,976,121.79	$11,650.27
Germany	$5,978,108.14	$10,926.27
Ireland	$3,159,782.90	$16,449.54
Japan	$14,957,294.27	$22,995.99
Norway	$1,627,858.74	$25,594.52
Sweden	$2,774,068.00	$8,372.94
United Kingd	$15,053,892.94	$53,040.96
USA	$61,254,995.65	$59,133.92

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.

		Amount Sales	Rank by Country
Books	Argentina	347,544.361	12
	Australia	9,412,848.248	4
	Brazil	568,645.399	11
	Canada	3,667,768.272	7
	France	3,359,648.059	5
	Germany	3,568,789.42	6
	Ireland	2,061,186.883	8
	Japan	9,344,287.124	2
	Norway	1,178,789.796	10
	Sweden	1,594,555.432	9
	United Kingd	10,234,860.015	3
	USA	43,622,679.262	1
Music	Argentina	241,577.909	12
	Australia	2,976,666.105	4
	Brazil	363,025.444	11
	Canada	1,266,314.453	7
	France	1,760,651.286	5
	Germany	1,508,942.386	6
	Ireland	706,185.127	8
	Japan	3,949,849.182	2
	Norway	238,788.625	10
	Sweden	789,847.619	9
	United Kingd	3,017,461.916	3
	USA	10,650,603.589	1

Project Sales

This example shows a calculation for a 20% increase in sales projections for each quarter, based on Amount Sales for 1999.

20% Projected Revenue Increase

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.

`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 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.

```
Avg (Amount)
```

Table 135   Avg Example 1

State	City	Amount	Computed
AZ	Tucson	112	183.67
CA	Burbank	240	183.67
CA	Glendale	97	183.67
FL	Palmetto	70	183.67
MD	Laurel	97	183.67
MI	Detroit	208	183.67
MN	Eagan	159	183.67
NY	New York	490	183.67
NY	Rochester	180	183.67

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.

```
Avg (Amount, State)
```

Table 136   Avg Example 2

State	City	Amount	Computed
AZ	Tucson	112	112
CA	Burbank	240	168.5
CA	Glendale	97	168.5
FL	Palmetto	70	70
MD	Laurel	97	97
MI	Detroit	208	208
MN	Eagan	159	159
NY	New York	490	335
NY	Rochester	180	335

AvgNonNull

The `AvgNonNull` function returns the average (arithmetic mean) of values in a number column, excluding null values.

`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

`AvgNonNull (Amount)`

State	City	Amount	Computed
AZ	Tucson	112	194.5
CA	Burbank	240	194.5
CA	Glendale	NULL	194.5
FL	Palmetto	70	194.5
MD	Laurel	97	194.5
MI	Detroit	208	194.5
MN	Eagan	159	194.5
NY	New York	490	194.5
NY	Rochester	180	194.5

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

`AvgNonNull (Amount, State)`

Table 137   AvgNonNull Example 2

State	City	Amount	Computed
AZ	Tucson	112	112
CA	Burbank	240	240
CA	Glendale	NULL	240
FL	Palmetto	70	70
MD	Laurel	97	97
MI	Detroit	208	208
MN	Eagan	159	159
NY	New York	490	335
NY	Rochester	180	335

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 138   AvgNonNull Example 3

State	City	Amount	Computed
AZ	Tucson	112	240
CA	Burbank	240	240
CA	Glendale	NULL	240
FL	Palmetto	70	240
MD	Laurel	97	240
MI	Detroit	208	240
MN	Eagan	159	240
NY	New York	490	240
NY	Rochester	180	240

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 139   ColMax Example

State	City	Amount	Computed
AZ	Tucson	112	490
CA	Burbank	240	490
CA	Glendale	NULL	490
FL	Palmetto	70	490
MD	Laurel	97	490
MI	Detroit	208	490
MN	Eagan	159	490
NY	New York	490	490
NY	Rochester	180	490

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 140   ColMin Example 1

State	City	Amount	Computed
AZ	Tucson	112	70
CA	Burbank	240	70
CA	Glendale	NULL	70
FL	Palmetto	70	70
MD	Laurel	97	70
MI	Detroit	208	70
MN	Eagan	159	70
NY	New York	490	70
NY	Rochester	180	70

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 141   ColMin Example 2

State	City	Amount	Computed
AZ	Tucson	112	112

State	City	Amount	Computed
CA	Burbank	240	240
CA	Glendale	Null	240
FL	Palmetto	70	70
MD	Laurel	97	97
MI	Detroit	208	208
MN	Eagan	159	159
NY	New York	490	180
NY	Rochester	180	180

Count

The Count function counts the number of rows in a column.

```
Count (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.

```
Count (Amount)
```

Table 142   Count Example

State	City	Amount	Computed
AZ	Tucson	112	9
CA	Burbank	240	9
CA	Glendale	97	9

State	City	Amount	Computed
FL	Palmetto	70	9
MD	Laurel	97	9
MI	Detroit	208	9
MN	Eagan	159	9
NY	New York	490	9
NY	Rochester	180	9

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 143   CountDistinct Example 1

State	City	Amount	Computed
AZ	Tucson	112	8
CA	Burbank	240	8

State	City	Amount	Computed
CA	Glendale	240	8
FL	Palmetto	70	8
MD	Laurel	97	8
MI	Detroit	208	8
MN	Eagan	159	8
NY	New York	490	8
NY	Rochester	180	8

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 144   CountDistinct Example 2

State	City	Amount	Computed
AZ	Tucson	112	1
CA	Burbank	240	1
CA	Glendale	240	1
FL	Palmetto	70	1
MD	Laurel	97	1
MI	Detroit	208	1
MN	Eagan	159	1
NY	New York	490	1
NY	Rochester	180	1

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 145   CountNull Example

State	City	Amount	Computed
AZ	Tucson	112	1
CA	Burbank	240	1
CA	Glendale	NULL	1
FL	Palmetto	70	1
MD	Laurel	97	1
MI	Detroit	208	1
MN	Eagan	159	1
NY	New York	490	1
NY	Rochester	180	1

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.

```
CountNonNull (Amount)
```

Table 146   CountNonNull Example 1

State	City	Amount	Computed
AZ	Tucson	112	8
CA	Burbank	240	8
CA	Glendale	NULL	8
FL	Palmetto	70	8
MD	Laurel	97	8
MI	Detroit	208	8
MN	Eagan	159	8
NY	New York	490	8
NY	Rochester	180	8

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.

```
CountNonNull (Amount, State)
```

Table 147   CountNonNull Example 2

State	City	Amount	Computed
AZ	Tucson	112	1
CA	Burbank	240	1
CA	Glendale	NULL	1
FL	Palmetto	70	1
MD	Laurel	97	1

State	City	Amount	Computed
MI	Detroit	208	1
MN	Eagan	159	1
NY	New York	490	2
NY	Rochester	180	2

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 148   CountNonNull Example 3

State	City	Amount	Computed
AZ	Tucson	112	1
CA	Burbank	240	1
CA	Glendale	97	1
FL	Palmetto	70	1
MD	Laurel	97	1
MI	Detroit	208	1
MN	Eagan	159	1
NY	New York	490	1
NY	Rochester	180	1

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.

`Cume (Amount)`

Table 149   Cume Example 1

State	City	Amount	Computed
AZ	Tucson	112	112
CA	Burbank	240	352
CA	Glendale	97	449
FL	Palmetto	70	519
MD	Laurel	97	646
MI	Detroit	208	824
MN	Eagan	159	983
NY	New York	490	1473
NY	Rochester	180	1653

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.

`Cume (Amount, State)`

Table 150   Cume Example 2

State	City	Amount	Computed
AZ	Tucson	112	112
CA	Burbank	240	240
CA	Glendale	97	337
FL	Palmetto	70	70
MD	Laurel	97	97
MI	Detroit	208	208
MN	Eagan	159	159
NY	New York	490	490
NY	Rochester	180	670

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).

```
Median (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 Median function in two separate tables.

Example 1

In this example, the Median function is used on a numeric column that has an odd number of rows:

```
Median (Amount)
```

The Median function returns the number in the middle, which in this example is 30.

Table 151   Median Example 1

State	Amount	Computed
AZ	10	30
CA	20	30
FL	30	30
MD	40	30
MI	50	30

If the numbers column has an even number of rows, the Median function calculates the average of the two numbers in the middle.

Table 152   Median Example 2

State	Amount	Computed
AZ	10	25
CA	20	25
FL	30	25
MD	40	25

Mode

The Mode function returns the most frequently occurring value in columns of numbers.

```
Mode (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.

```
Mode(Amount)
```

Table 153 Mode Example

State	Amount	Computed
AZ	10	10
CA	10	10
FL	0	10
MD	40	10
MI	50	10

Percentile

The `Percentile` function returns the nth percentile of values in a column of numbers.

```
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 nth 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 `Percentile` function is used on the numeric column and the nth percentile of values (.80 in this case).

`Percentile (Units, .80)`

In the second computed column, the `Percentile` function is used on the numeric column, the nth percentile of values (.80 in this case), and the break_column.

`Percentile (Units, .80, State)`

Table 154   Percentile Example 1

State	Units	Computed 1	Computed 2
WI	50	191.8	50
AZ	70	191.8	70
CA	96	191.8	99.8
CA	98	191.8	99.8
CA	101	191.8	99.8
FL	112	191.8	112
MD	159	191.8	159
NY	241	191.8	440.2
NY	490	191.8	440.2

Example 2

In this example, two computed value columns have been calculated. In the first computed column, the `Percentile` function is used on the numeric column and the nth percentile of values (0 in this case).

`Percentile (Units, 0)`

In the second computed column, the `Percentile` function is used on the numeric column, and the nth percentile of values (.25 in this case).

`Percentile (Units, .25)`

Table 155   Percentile Example 2

State	Units	Computed 1	Computed 2
WI	50	50	96
AZ	70	50	96
CA	96	50	96

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 156   Rank Example

State	City	Amount	Computed
AZ	Tucson	112	6
CA	Burbank	241	2
CA	Glendale	97	7
FL	Palmetto	70	9
MD	Laurel	97	7
MI	Detroit	208	3

State	City	Amount	Computed
MN	Eagan	159	5
NY	New York	490	1
NY	Rochester	180	4

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 157   RankAsc Example 1

State	City	Amount	Computed
AZ	Tucson	112	4
CA	Burbank	240	8
CA	Glendale	97	2
FL	Palmetto	70	1
MD	Laurel	97	2
MI	Detroit	208	7
MN	Eagan	159	5
NY	New York	490	9

State	City	Amount	Computed
NY	Rochester	180	6

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 158   RankAsc Example 2

State	City	Amount	Computed
AZ	Tucson	112	3
CA	Burbank	240	6
CA	Glendale	97	6
FL	Palmetto	70	1
MD	Laurel	97	2
MI	Detroit	208	5
MN	Eagan	159	4
NY	New York	490	7
NY	Rochester	180	7

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 159   StdDevp Example

State	City	Amount	Computed
AZ	Tucson	112	120.79
CA	Burbank	240	120.79
CA	Glendale	97	120.79

State	City	Amount	Computed
FL	Palmetto	70	120.79
MD	Laurel	97	120.79
MI	Detroit	208	120.79
MN	Eagan	159	120.79
NY	New York	490	120.79
NY	Rochester	180	120.79

Sum

The Sum function computes the total for a column of numbers.

```
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.

```
Sum(Amount)
```

Table 160   Sum Example 1

State	City	Amount	Computed
AZ	Tucson	112	1556
CA	Burbank	240	1556
CA	Glendale	NULL	1556
FL	Palmetto	70	1556
MD	Laurel	97	1556

State	City	Amount	Computed
MI	Detroit	208	1556
MN	Eagan	159	1556
NY	New York	490	1556
NY	Rochester	180	1556

Example 2

In this example, the `Sum` function has been inserted for the numeric column and `break_column` using the following definition:

`Sum (Amount,State)`

The results are shown in the Computed column.

Table 161   Sum Example 2

State	City	Amount	Computed
AZ	Tuscon	112	112
CA	Burbank	240	240
CA	Glendale	Null	240
FL	Palmetto	70	70
MD	Laurel	97	97
MI	Detroit	208	208
MN	Eagan	159	159
NY	New York	490	670
NY	Rochester	180	670

Var

The `Var` function estimates variance based on a sample.

`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 162   Var Example 1

State	City	Amount	Computed
AZ	Tucson	112	16,413.25
CA	Burbank	240	16,413.25
CA	Glendale	97	16,413.25
FL	Palmetto	70	16,413.25
MD	Laurel	97	16,413.25
MI	Detroit	208	16,413.25
MN	Eagan	159	16,413.25
NY	New York	490	16,413.25
NY	Rochester	180	16,413.25

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 163   Var Example 2

State	City	Amount	Computed
AZ	Tucson	112	0
CA	Burbank	240	10,224.50
CA	Glendale	97	10,224.50
FL	Palmetto	70	0
MD	Laurel	97	0
MI	Detroit	208	0
MN	Eagan	159	0
NY	New York	490	48,050.00

State	City	Amount	Computed
NY	Rochester	180	48,050.00

Varp

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 164 Varp Example

State	City	Amount	Computed
AZ	Tucson	112	14,589.56
CA	Burbank	240	14,589.56
CA	Glendale	97	14,589.56
FL	Palmetto	70	14,589.56
MD	Laurel	97	14,589.56
MI	Detroit	208	14,589.56
MN	Eagan	159	14,589.56
NY	New York	490	14,589.56
NY	Rochester	180	14,589.56

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 467
- "Exponential Moving Averages" on page 469
- "Moving Diff" on page 470
- "Moving Maximum" on page 471
- "Moving Median" on page 473
- "Moving Sum" on page 474
- "Moving Minimum" on page 475
- "Direction Of Moving Function Calculation" on page 476

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:

	Month	Sales
1	Jan	10
2	Feb	15
3	Mar	17
4	Apr	20
5	May	22
6	Jun	20
7	Jul	25
8	Aug	27
9	Sep	30
10	Oct	35
11	Nov	37
12	Dec	40

The simple mean value of all of the above twelve Sales values is:

```
Sum of all twelve values divided by 12 = 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.

	Month	Sales
1	Jan	10
2	Feb	15
3	Mar	17
4	Apr	20
5	May	22
6	Jun	20
7	Jul	25
8	Aug	27
9	Sep	30
10	Oct	35
11	Nov	37
12	Dec	40

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:

```
(10 + 15 + 17) / 3 = 42 / 3 = 14
```

Next, consider the mean of the sum of the second, third and fourth original values i.e.:

```
(15 + 17 + 20) / 3 = 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 165 Simple Moving Average

Month	Original Sales Values	3-Month Simple Moving Average
Jan	10	
Feb	15	
Mar	17	14
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

Month	Original Sales Values	3-Month Simple Moving Average
Nov	37	34
Dec	40	37.33

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 166 Centered and Trailing Averages

Month	Centered Averages	Trailing Averages
Jan		
Feb	14	
Mar		14

"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 167 3-Month Simple Moving Averages

Month	Original Sales Values	3-Month Simple Moving Average
Jan	10	
Feb	15	
Mar	17	14

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:

	Month	Sales
1	Jan	10
2	Feb	15
3	Mar	17
4	Apr	20
5	May	22
6	Jun	20
7	Jul	25
8	Aug	27
9	Sep	30
10	Oct	35
11	Nov	37
12	Dec	40

➤ 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:

```
{(Dec Sales value * 3) + (Nov Sales value * 2) + (Oct Sales value * 1)} divided by
(3 + 2 + 1)
= {(40 * 3) + (37 * 2) + (35 * 1)} / 6
= (120 + 74 + 35) / 6
= 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 168   3-month Weighted Moving Average

Month	Original Sales Values	3-Month Simple Moving Average
Jan	10	
Feb	15	
Mar	17	15.12
Apr	20	18.17
May	22	20.5
Jun	20	20.67
Jul	25	22.83
Aug	27	25.17
Sep	30	28.17
Oct	35	32
Nov	37	35.17
Dec	40	38.17

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 * (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 = Smoothing Constant = 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:

	Month	Sales
1	Jan	10
2	Feb	15
3	Mar	17
4	Apr	20
5	May	22
6	Jun	20
7	Jul	25
8	Aug	27
9	Sep	30
10	Oct	35
11	Nov	37
12	Dec	40

If we calculated the Exponential Moving Average in a similar fashion to the 3-Month Simple Moving Average, we would perform the following steps:

➤ To calculate the exponential moving average:

1 Calculate the Smoothing Constant according to the 2 / (1 + n) formula.

N = window of values = 3, therefore the Smoothing Constant is:

```
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 * (C - P)) + P as indicated below:

Table 169   X = (K * (C - P)) + P Example

Month	Sales	Calculation (K * (C - P)) + P	Exponential Moving Average
Jan	10	= Original value	10
Feb	15	(0.5 * (15 - 10)) + 10	12.5
Mar	17	(0.5 * (17 - 12.5)) + 12.5	14.75
Apr	20	(0.5 * (20 - 14.75)) + 14.75	17.375
May	22	(0.5 * (22 - 17.375)) + 17.375	19.688
Jun	20	(0.5 * (20 - 19.6875)) + 19.6875	19.844
Jul	25	(0.5 * (25 - 19.844)) + 19.844	22.422
Aug	27	(0.5 * (27 - 22.422)) + 22.422	24.711
Sep	30	(0.5 * (30 - 24.711)) + 24.711	27.355
Oct	35	(0.5 * (35 - 27.355)) + 27.355	31.178
Nov	37	(0.5 * (37 - 31.178)) + 31.178	34.089
Dec	40	(0.5 * (40 - 34.089)) + 34.089	37.044

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 170   Moving Difference Example

Month	Original Sales Values	3-Month MovingDiff Average
Jan	10	
Feb	15	
Mar	17	7
Apr	20	5
May	22	5
Jun	20	0
Jul	25	3
Aug	27	7
Sep	30	5
Oct	35	5
Nov	37	7
Dec	40	5

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

is similar to the Simple Moving Average, No "Weighted" or "Exponential" can be calculated for 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 171 MovingMax Function Example

Month	Original Sales Values	3-Month MovingMax
Jan	10	
Feb	15	
Mar	17	17
Apr	20	20
May	22	22
Jun	20	22
Jul	25	25
Aug	27	27
Sep	30	30
Oct	35	35
Nov	37	37
Dec	40	40

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 interger 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 172   MovingMed Function Example

Month	Original Sales Values	3-Month MovingMed Average
Jan	10	
Feb	15	
Mar	17	15
Apr	20	17
May	22	20
Jun	20	20
Jul	25	22
Aug	27	25
Sep	30	27
Oct	35	30

Month	Original Sales Values	3-Month MovingMed Average
Nov	37	35
Dec	40	37

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 interger 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 173   Moving Sum Example

Month	Original Sales Values	3-Month MovingSum Average
Jan	10	
Feb	15	
Mar	17	42
Apr	20	52
May	22	59
Jun	20	62

Month	Original Sales Values	3-Month MovingSum Average
Jul	25	72
Aug	27	72
Sep	30	82
Oct	35	92
Nov	37	102
Dec	40	112

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 interger 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 174   Moving Minimum Example

Month	Original Sales Values	3-Month Moving Minimum
Jan	10	
Feb	15	
Mar	17	10
Apr	20	15
May	22	17
Jun	20	20
Jul	25	20
Aug	27	20
Sep	30	25
Oct	35	27
Nov	37	30
Dec	40	35

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 175   Moving Function Example

Month	Sales
Jan	10
Feb	15
Mar	17
Apr	20
May	22

Month	Sales
Jun	20
Jul	25
Aug	27
Sep	30
Oct	35
Nov	37
Dec	40

Adding a new Computed Item column to display a 3-Month Moving Difference would yield:

Table 176   3-Month Moving Difference Example

Month	Sales	3-Month Moving Difference
Jan	10	
Feb	15	
Mar	17	7
Apr	20	5
May	22	5
Jun	20	0
Jul	25	3
Aug	27	7
Sep	30	5
Oct	35	5
Nov	37	7

Consider, however, if the "Month" and "Sales" data were oriented as follows (as could be done in a Pivot section):

	Month											
	Jan	**Feb**	**Mar**	**Apr**	**May**	**Jun**	**Jul**	**Aug**	**Sep**	**Oct**	**Nov**	**Dec**
Sales	10	15	17	20	22	20	25	27	30	35	37	40

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).

	Month											
	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Sales	10	15	17	20	22	20	25	27	30	35	37	40
3-Month Moving Difference												

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:

	Month											
	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Sales	10	15	17	20	22	20	25	27	30	35	37	40
3-Month Moving Difference			7	5	5	0	3	7	5	5	7	5

Applying Sorts

15

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.

Year	State	Item Type	Amounts	Units
1997	NY	The Standard	8,958.6	70
1996	NY	EZ Fax Modem-I	7,203.9	55
1996	MI	The Turbo	7,986.6	45
1996	MI	1024 MB Drive	10,438	40
1996	NY	The Turbo	11,536.2	65
1995	NY	1024 MB Drive	9,133.25	35
1996	CA	EZ Fax Modem-I	8,513.7	65
1995	CA	1024 MB Drive	10,438	40

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.

Year	State	Item Type	Amounts	Units
1995	CA	1024 MB Drive	7,828.5	30
1995	CA	1024 MB Drive	10,438	40
1995	CA	1024 MB Drive	10,438	40
1995	NY	1024 MB Drive	9,133.25	35
1996	CA	1024 MB Drive	9,133.25	35
1996	CA	560 MB Drive	8,428.8	60
1996	CA	EZ Fax Modem-I	7,858.8	60
1996	CA	EZ Fax Modem-I	8,513.7	65

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 479 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, 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.

16

Understanding Connectivity in Interactive Reporting

This section describes how to connect to a relational database and a *multidimensional database* using Interactive Reporting database connection files, including how to set up these files, define connection preferences, and how to manage connections.

About Interactive Reporting Database Connection Files

Connectivity is generally one of the most difficult aspects of querying for end users to master. Client/server database applications rely on a complicated web of listeners, network addresses, and preferences that are difficult for anyone but a database administrator to troubleshoot.

Interactive Reporting users can sidestep these potential difficulties by using an Interactive Reporting database connection file (.oce files) to define the terms, conditions, and methods for connecting to data sources.

With a database administrator's assistance, Interactive Reporting database connection files enable a stable connection to be set up once, and then distributed and reused. End users supply a database user name and password each time they log on to query a database.

Interactive Reporting database connection files retain all the information necessary to log on to a specific configuration of database and connection API software. In addition, Interactive Reporting database connection files retain DBMS-specific connection preferences as well as specifications for automatic access to metadata (see "Using Metatopics and Metadata" on page 515).

Interactive Reporting database connections store complete sets of connection parameters about:

- Connection software

- Database software
- Database server hosts
- Database user names (optional)

> **Note:** For security reasons, user passwords are not saved with Interactive Reportings database connection files.

Interactive Reporting database connections have significant advantages in network environments. One connection can be created for each database in the environment and shared with each end user.

Interactive Reporting database connection files simplify the connection process for company personnel by transparently handling host and configuration information. Each user can substitute their own database user name when using an Interactive Reporting database connection file (.oce), which enforces security measures and privileges that are centralized at the database server.

Because passwords are not saved, there is no danger that distribution provides unauthorized access to any user who receives the wrong Interactive Reporting database connection file or acquires it from other sources.

By default, no explicit Interactive Reporting database connection file is required to process Interactive Reporting document files or job outputs using the EPM Workspace or Interactive Reporting Web Client. That is, an user is not required to have specific access privileges to process an Interactive Reporting document file. However, a control setting of an Interactive Reporting document file or job access can be defined to require explicit access. For more information, see the *EPM Workspace Administrator's Guide* and the *EPM Workspace User's Guide*.

> **Note:** It is to your advantage to create and distribute Interactive Reporting database connection files to facilitate the logon process when storing Interactive Reporting Studio data models.

Working with Interactive Reporting Database Connections

Interactive Reporting Studio provides a Database Connection Wizard to help you create new Interactive Reporting database connections. Before you create a new Interactive Reporting database connection files, collect and verify the following connection information:

- Connection API software and version (for example, Essbase, SQL*Net for Windows NT, and so on)
- Database software and version (for example, MetaCube 4, Oracle 8, and so on)
- IP address, database alias, or ODBC data source name of database server
- Database user name

Creating Interactive Reporting Database Connections (.oces)

The Database Connection Wizard steps you through the Interactive Reporting database connection file creation process and captures the connection parameters in a file that enables you to connect to a data source. Interactive Reporting Studio saves the cInteractive Reporting database connection file in the default Interactive Reporting database connection directory.

With an advanced user's assistance, an Interactive Reporting database connection file can be set up once, distributed, and reused. You only have to supply a database user name and password each time you log in to query a database.

Note: There is a correlation between the connectivity software and database choices. For example if you select Essbase for the connection software, only .Essbase is available as a database choice.

➤ To create an Interactive Reporting database connection:

1 Select **Tools**, then **Connection**, and then **Create**.

   The Database Connection Wizard is displayed.

2 Select the connection software used for connecting to the database server from the pull down list in the **What connection software do you want to use?** field.

3 Select the database server in the **What type of database do you want to connect to?** field.

4 To configure metadata settings, select **Show Meta Connection Wizard**.

5 To configure advanced connection preferences, select **Show advanced options**.

6 Click **Next**.

7 Depending on the database, enter your user name in the User Name field, password in the Password field, the IP address, ODBC database source or server alias name in the Host field and click **Next**.

   If you selected to work with meta data settings, the Meta Data Connection Wizard launches. See the "Accessing the Open Metadata Interpreter" on page 519 for more information.

8 The wizard prompts you to save the Interactive Reporting database connection file.

9 To save the Interactive Reporting database connection file, click **Yes**.

   The Save dialog box is displayed. Interactive Reporting Studio saves theInteractive Reporting database connection file in the default Interactive Reporting connection directory.

10 To save the Interactive Reporting database connection file in a different directory, navigate to the desired directory and click **Save**.

Table 177   Database Connection Configuration Wizard options

Option	Description
What connection software do you want to use?	Select the connection software to use when connecting to the database. Depending on the connection software, additional fields may be displayed. These fields enable you to customize the Interactive Reporting database connection file; show metadata settings; and select ODBC logon dialogs.

Option	Description
What type of database do you want to connect to?	Select the type of database to which to connect from the pull-down list.
Show Metadata Connection Wizard?	To view and edit meta data settings, select this field. The Metadata Definitions dialog box is configured with specific SQL statements to read meta data on multiple databases.
Show advanced options?	To select advanced preferences for the Interactive Reporting database connection file, select this field. Connection preferences enable you to select what instructions and protocols the Interactive Reporting database connection file should observe. The preferences are saved with the Interactive Reporting database connection file and applied each time you use the connection. For example, you can use connection preferences to filter extraneous tables from the Table Catalog or specify how the connection software should manage SQL statements. Connection preferences vary depending on the connection software and database.
Prompt for database name	To select the specific database name on the server, select this field.
Use ODBC Logon Dialogs?	If you select ODBC connection software, use the ODBC logon dialog boxes instead of the Interactive Reporting Studio dialog boxes, select this field. To use the Interactive Reporting Studio connection dialog boxes, leave this field unchecked.
User Name	Enter the sign on name to the database.
Password	Enter the password for the database
Host	Enter the IP address, database alias, or ODBC data source name.

Note: In Release 9.3, Hyperion supports connections to Informatica MX2 repositories, specifically connections to the Informatica PowerCenter 7.1 repositories using the Informatica Metadata Exchange SDK 7.1 client software. The Connection Wizard has been changed to prompt for the repository server host name and port number instead of the ODBC connection string, user name and password. The Informatica Metadata Exchange SDK 7.1 client software must installed to the server or the machine that has DAS running to work in the EPM Workspace and in Interactive Reporting Web Client. Connections to Informatica MX2 repositories are supported on Windows platforms only.

Setting Connection Preferences

Connection preferences specify the way certain aspects of the Interactive Reporting database connection file are managed. The preferences are saved with an Interactive Reporting database connection file and are applied each time you use the connection.

For example, use connection preferences to filter extraneous tables from the Table Catalog or change the way the connection software handles SQL transaction statements.

Connection preferences differ depending on the Interactive Reporting Studio edition, connection API, and DBMS.

Connection preferences are accessed by selecting the Show Advanced Options check box in the Database Connection Wizard. Table 178 list all of possible options that are available in the Wizard: the options available to you depend on the connection configuration.

Table 178   Database Connection Configuration Wizard options

Option	Description
ALLOW SQL-92 Advanced Set Operations	Enables support for the Intersection and Difference operators in the Append Query option.
Apply Filters to restrict the tables that are displayed in the table catalog	Enables specification of table filter conditions for limiting or customizing the list of tables in the table catalog.
Exclude Hyperion Repository Tables	Excludes all repository tables from the table catalog. "Filter by" and "metadata" definitions override this preference.
Allow Non-Joined Queries	Prohibits processing when topics are not joined in the Query Contents frame.
Use SQL to get Table Catalog	Specifies use of SQL to retrieve tables, instead of using SQL Server sp_tables and sp_columns stored procedures. This option enables table filtering, but may be slower than stored procedures. (Sybase and MS SQL Server)
Choose the Data Retrieval Method	Specifies how the server returns data. In most cases, "Retrieve data as Binary" is the most appropriate, and fastest method. Select "Retrieve data as Strings" if the connection API does not support native datatype retrieval, or if queries return incorrect or unreadable data.
Time Limit ____ Minutes	Establishes an automatic disconnect from the database after the specified period of inactivity.
Auto Commit After Select	Sends a commit statement to the database server with each Interactive Reporting Studio SQL statement to unlock tables after they have been used. Use this feature if tables are locked after use or users experience long waits for tables.
Save Interactive Reporting Database connection Without User Name	Enables general distribution of an Interactive Reporting database connection file by saving it generically, without a user name. Instead, any user can log on by typing their own user name.
Use Quoted Identifiers	Specifies that internal keywords or table and column, or owner names with special characters sent to the server be enclosed in quotation marks. For example, SELECT SUM("AMOUNT"), "STORE_ID" FROM "HYPERION"."PCS_SALES"GROUP BY"STORE_ID" The default value for new connections is off.
Allow Change Database at Logon	Adds Database field to logon dialog box enabling the user to select a specific database when logging on to the DBMS. (Sybase and MS SQL Server)
Use large buffer query mode	Specifies a binding process to retrieve more records per fetch call. If the ODBC driver supports binding, use this option for faster retrieval. (ODBC only). If this feature is turned on, the ODBC Extended Fetch call requests data at 32k at a time.

Option	Description
Packet Size Setting. 512*	Enables Sybase's DB-Lib users to set up a large buffer retrieval from the database so that more data can be transferred at one time.
	If this feature is selected, you can specify a multiple of 512 bytes for the number of bytes to transfer at one time.
	Before you specify a multiple of 512 bytes, the server must have enough memory to allocate for the transmission of the selected packet size.
	To check which packet size the Sybase server supports, run the isql command:
	`sp_configure`
	and type go.
	A list of parameters is returned. Find the parameter showing the "Maximum Network Packet Size." If the packet size you entered exceeds the maximum packet size, you have to enter again a smaller packet size.
	To change the packet size, issue the following command in isql:
	`Sp_configure "maximum network packet size". <new value>`
	(where <new value> is the new size).
Oracle Buffer Size	Determines the default buffer size when retrieving rows of data from an Oracle connection. The default size is 8000 bytes. A user can change this value to retrieve more rows per buffer, which may result in a performance improvement, but at the expense of additional memory requirements. The minimum size is 8000. If a user specifies a smaller value, nor error is returned, but 8000 bytes is used. There is no hard coded maximum size value for this field.
Disable Asynchronous Processing	Turns off the ability to make simultaneous requests to the database server.
	This feature is available in Interactive Reporting Studio only.
	Note:
Retain Data Formats	Interactive Reporting Studio uses the default formats specified by the database server when handling date, time, and timestamp values. If the default formats of the server have been changed, you can retain or preserve these adjusted preferences to ensure Interactive Reporting Studio interprets date/time values correctly.
Server Dates	Enables alteration of internal Interactive Reporting Studio date handling to match server default settings in case of a discrepancy. For more information on this feature, see "Modifying Server Date Formats" on page 493.
Disable Transaction Mode	On upload to the repository, Interactive Reporting Studio brackets SQL Insert statements with transaction statements. Disable Transaction Mode if the RDBMS does not support transactions.
	This feature is only available in Interactive Reporting Studio.
Do you want to save your OCE	Enables you to save theInteractive Reporting database connection file so that it can be reused at a later time.
Use outer join operator on limits	Inserts an outer join operator (+) in the SQL on limits applied to the "inner" table for Oracle Net connection software to an Oracle database. By default this feature is enabled and is recommended; it is provided to work around Oracle restrictions when using outer joins with certain limit conditions, such as when an OR expression is needed. An outer join operator enables Interactive Reporting Studio to retrieve all rows from the "left" or "right" table matching joined column values if found or retrieves nulls for non-matching values. If this feature is disabled, then nulls for non-matching values are not retrieved. Use the Join Properties dialog box to assist in determining which is the "left" and "right" table. Oracle does not support full (left AND right) outer joins with the (+) operator. When an ODBC driver is used, this feature is greyed out.

Option	Description
Use ODBC outer join syntax on limits	When a limit has been applied to an inner table of an outer join, this feature enables the limit to be placed on the On clause of the SQL statement instead of the Where clause. The default setting for this feature is unchecked.
Use ODBC outer join escape syntax	Inserts ODBC outer join escape syntax in the SQL statement. **Note:** Disable this option if nested outer joins are needed.

Filtering Tables

For databases with many tables, it can help to filter out tables you do not need from the Table catalog. The table filter enables you to specify filter conditions based on table name, owner name, or table type (table or virtual views).

Note: The table filter works with all database server connections except ODBC. If you are working with a Sybase or Microsoft SQL Server database, modify the connection and specify that Interactive Reporting Studio use SQL statements to retrieve the Table catalog before filtering tables.

Typically, you filter tables when creating an Interactive Reporting database connection file, although you can modify an existing Interactive Reporting database connection file later to filter tables.

➤ To filter tables from the Table Catalog when creating an Interactive Reporting database connection file:

1 Select **Tools**, then **Connection**, and then **Create**.

The Database Connection Wizard is displayed.

2 Select the connection software used to connect to the database server from the pull down list in the **What connection software do you want to use?** field.

3 Select the database server in the **What type of database do you want to connect to?** field.

4 Select **Show Advanced Options** and click **Next**.

5 Connect to the data source and click **Next**.

The dialog box varies according to the connection software you are using. In most cases, you need to specify a user name, password and host name. Click *Next*.

6 Click **Define** next to a table name, table owner, or table type filter check box.

7 Select a comparison operator from the drop-down box. The filter constraints determine which tables are included in the Table catalog.

Complete a filter definition by doing one of the following:

● Enter constraining values in the edit field and select the check mark.

- Click **Show Values** to display a list of potential database values and select values from the list.

- If you are comfortable writing your own SQL statements, click **Custom SQL** to directly code table filters that have greater flexibility and detail.

8 Click **OK**.

Interactive Reporting Studio prompts you to save the filter settings. Once saved, a check mark displays in the appropriate filter check box, which you can use to toggle the filter on and off.

Note: After you complete the Data Connection Wizard, verify that the filter conditions screen out the correct tables. In the Catalog frame, select **Refresh** on the shortcut menu.

➤ To filter tables from the Table Catalog when modifying an Interactive Reporting database connection file:

1 To filter tables for the current connection, select **Tools**, then **Connection**, and then **Modify**.

The Meta Connections Wizard dialog box is displayed.

2 To filter tables for another connection, select **Tools**, then **Connections Manager**, and then **Modify**.

The Connections Manager dialog box is displayed. In the Document Connections frame, select the Interactive Reporting database connection file and click *Modify*.

The Meta Connections Wizard dialog box is displayed.

3 Configure the first Wizard as necessary, and then click **Next** to go to the second Meta Connections Wizard dialog box.

4 Configure the second Wizard as necessary, and then click **Next** to go to the third Meta Connection Wizard dialog box.

5 On the third Meta Connection Wizard dialog box, click **Define** next to a owner, table or type filter check box.

A Filter dialog box is displayed. The Filter dialog boxes resemble and operate using the same principles as the Limit dialog box.

6 Select a comparison operator from the drop-down box. The filter constraints determine which tables are included in the Table Catalog.

7 Complete a filter definition by doing one of the following:

Enter constraining values in the edit field and select the check mark.

Click *Show Values* to display a list of potential database values and select values in the frame.

If you are comfortable writing your own SQL statements, click *Custom SQL* to code table filters directly with greater flexibility and detail. For example, you can write a SQL filter, which enables only tables beginning with "Sales" to be displayed in the table catalog. As new "Sales" tables are added to the database, they automatically are displayed in the Table Catalog.

8   Select any other customizing options to apply, and click **OK**.

You are prompted to save the filter settings. Once saved, a check mark is displayed in the appropriate filter check box, which you can use to toggle the filter on and off.

9   Click **Next** to continue through each dialog box, selecting any preferences for theInteractive Reporting database connection file.

10 Click **Finish**.

11 In the Hyperion dialog box, click **Yes** to save the Interactive Reporting database connection file.

12 In the Save Open Catalog dialog box, browse to a directory, enter the new connection name in the **File Name** field, and then click **Save**.

13 In the Table Catalog of the Query section, select **Refresh** on the shortcut menu to verify that the filter conditions screen out the correct tables.

Modifying Server Date Formats

Interactive Reporting Studio uses the default formats specified by the database server when handling date, time, and timestamp values. If the default formats of the server have been changed, you can adjust preferences to ensure that Interactive Reporting Studio interprets date/time value

➤ To modify server date formats:

1   Select **Tools**, and then **Connection**, and then **Create**.

The Database Connection Wizard is displayed.

2   Select **Show Advanced Options**, then **Next**.

3   Click **Server Dates**.

The Server Date Formats dialog box is displayed.

● **To Server Formats**—Date and time formats submitted to the server (such as limit values for a date or time field).

● **From Server Formats**—Formats Interactive Reporting Studio expects for date/time values retrieved from the server.

● The default values displayed in the To and From areas are usually identical.

4   If the server defaults have changed, select the date, time, and timestamp formats that match the new server defaults from the To and From format drop-down boxes.

If desired, click *Default* to restore all values to the server defaults stored in the Interactive Reporting database connection file.

5   If you cannot find a format that matches the database format, click **Custom**.

The Custom Format dialog box is displayed.

6   Select a data type from the Type drop-down box.

7   Select a format from the Format drop-down box or type a custom format in the Format field.

8   Click **OK**.

The new format is displayed as a menu choice in the Server Date Formats dialog box.

Setting up an Excel Spreadsheet as a Data Source

➤ To set up an Excel spreadsheet as a data source:

1 Open the Excel spreadsheet , and select the entire section (range of cells) of the Excel spreadsheet that is to be used as a "topic" (table) in Interactive Reporting.

   Note that the Name Box in the top left corner of the Excel window displays the name of the first cell selected (such as A1).

2 Click in this Name Box.

   Make sure that the entire range of cells is still selected in the main document.

3 Enter the name you want to use for this "table" when you access the Table Catalog in your data source from Interactive Reporting.

4 Confirm that the name is referencing all of the cells that you need.

   You can do this by deselecting the cells, then selecting the name just assigned from the Name Box. Verify that this highlights the correct range of cells.

5 Save the spreadsheet.

   You can repeat steps 1–5 in the spreadsheet to create multiple tables.

6 Open the ODBC control panel application and create a new Data Source using an Excel driver.

7 Point this data source to the spreadsheet file that you just manipulated.

8 Click **OK** to save the data source.

9 Open Interactive Reporting and create an Interactive Reporting document connection file (OCE) to point to this new Excel ODBC data source.

   This source should be created with ODBC for both the connection software and the Server.

10 Click **OK**.

11 On the ODBC control panel, assign a name to the data source and click **OK**.

12 On the Catalog Definition Window, click **OK**.

13 Save the Interactive Reporting document connection file (OCE) and give it a meaningful name.

Creating an OLAP Connection File

➤ To create an OLAP connection file:

1 Select **Tools**, then **Connection**, and then **Create**.

   The Database Connection Wizard is displayed.

2 Select the connection software used to connect to the *OLAP database* server from the drop-down box.

3 Select the OLAP database server from the drop-down box and click **Next**.

Depending on the database you select in this field, you may have to specify a password to connect to the database. Enter your name, password, and host address information.

The sequence of dialog boxes that are displayed depend on the multidimensional database server to which you are connecting. The following sections provide connection information for these multidimensional databases:

- Connecting to Essbase or DB2 OLAP
- Connecting to an OLE DB Provider
- Connecting to Essbase (CubeQuery only)

Connecting to Essbase or DB2 OLAP

➤ To connect to an Essbase or a DB2 OLAP database:

1 Follow the instructions for "Creating an OLAP Connection File" on page 494.

2 Select the application/database to which to connect and click **Next**.

This is the *cube* from which to retrieve values.

3 Select the *measures* dimension for the cube in the Dimension Name field and click **Next**.

This is the specific measure group from which to retrieve values.

4 Click **Finish** to save the Interactive Reporting database connection file.

Note: For information on connecting to an Essbase 7.x or 9.x server, see Connecting to Essbase (CubeQuery only)

Connecting to an OLE DB Provider

➤ To connect to an OLE DB provider:

1 Follow the instructions in "Creating an OLAP Connection File" on page 494.

2 Select the database to connect.

NT domain authentication is performed for OLAP cube files (.cub). If the user name and password provided (when attempting to process or retrieve dimensions) are not valid, an access error is returned and the user cannot access the file. To access the file, provide a valid NT domain user name and password. To specify a domain, enter it in the user name field in the form *DOMAIN\jdoe.*

Note: By default, Interactive Reporting Web Client users are prompted to enter their Windows credentials (user ID, password, and optionally Windows domain, which can be specified in the login user ID prompt field, preceding the user ID text and delimited by a backslash; for example, if domain is HyperionDomain and user ID is user1, HyperionDomain\user1 can be specified in the user ID field when logging on to Microsoft OLAP databases. These changes are enforced to provide more secure

access to these databases. If prompted, the user must enter credentials that can be successfully authenticated by the Windows operating system at the database server. Failure to provide credentials that can be successfully authenticated by Windows results in an error message being returned to the user and login to the database being denied. If the user's credentials are successfully authenticated, the database login proceeds and any role-based security on cube data granted at the database level for the specified user ID is invoked and honored. If no role-based security is implemented at the database level (the database cubes and their data are available to all users), the database administrator can choose to publish an Interactive Reporting database connection file or the database with a pre-assigned system-administrator-level user ID and password. If users access the database using this Interactive Reporting database connection file, they are not prompted to enter any login credentials. They have *passed through* to the database, where access to all cube data is allowed. Note that these statements also apply to Interactive Reporting Web Client users who access local cube files created from Microsoft OLAP or other OLE DB for OLAP databases (such as the sample cube files that are presented with the sample files provided with the installation).

3 If the OLE DB for OLAP database provides the ability to retrieve dimension properties, select **Enable Retrieval Of Dimension Properties** and click **Next**.

4 Select the name of the Provider from the drop-down box and click **Next**.

For more information about the remaining dialog boxes, consult the database documentation of the provider.

OBI Server

You can query OBI Server Presentation Layer topics by connecting to the Oracle BI Server and using it as a "proxy" data source. OBI Server refers to the Oracle Business Intelligence Server component where the residing semantic layer uses ODBC for external communication. OBI Server includes multiple sources and optimized SQL generation processes regardless of the schema. Although OBI Server exposes itself as an ODBC data source, and uses ODBC/ODBC connectivity functions properly, there are restrictions including:

- OBI Server is not a database.

- OBI Server cannot handle complex SQL constructs. Derived Queries and correlated sub-queries are not supported.

- Stored procedures are not supported.

- Process Result to Table is not supported.

- Retrieving the drill path from Oracle BI Server and applying it to pivots/charts is not supported.

- OBI Server masks physical schema from all clients, and only exposes presentation layer topics. As a result, joins between Interactive Reporting and OBI presentation data model topics are mostly non-functional. By default the 'Allow non joined Queries' option in Advanced options is enabled, and the prompt "This query is not properly joined. Do you wish to continue?" is not launched for queries that connect to the OBI Server. This behavior

is an exception specific to use of Oracle BI Server since Interactive Reporting assumes non-joined queries sent to Oracle BI Server produce a normal results sets.

● Certain database functions may not be supported.

The first example shows the available presentation level tables:

The second example shows the available presentation level topics:

Queries to OBI Server generate SQL statements in the same way as standard queries to other data sources.

Connecting to OBI Server

➤ To select OBI Server as a data source:

1   Launch **ODBC Data Source Administrator** and select **System DSN.**

    The ODBC Data Source Administrator dialog box is displayed.

2   Select the **System DSN** tab.

    The System DSN dialog box is displayed.

3   Select **Oracle BI Server**

    The Oracle BI Server DSN Configuration Wizard is displayed.

4   In the **Name** field, enter the name of the DSN, and select **Next.**

5   In the **Login ID** and **Password** field, enter your ORACLE BI Server login ID and password, and select **Next**

6   From the **Database** list, select a database.

    Optional: Enable the *Change the default catalog to* field, and select a default catalog.

7   Select **Finish.**

    The ODBC Data Source Administrator dialog box is displayed.

8   Select **OK.**

OBI Server Integration with Interactive Reporting

Each folder in the Oracle BI Server Semantic Layer Presentation Area appears as a database to Interactive Reporting. By default all such folders are displayed in the catalog tree of Interactive Reporting.

Presentation layer topics are available for you to choose from as data model topics including the:

● Presentation Layer (first column)

● Business Model Layer (second column)

● Physical Layer (third column)

Within Interactive Reporting, topics from the OBI Server presentation layer are exposed.

Tables are added to the data model as 'non-joined' tables. The query results set displays without prompting "This query is not properly joined. Do you wish to continue?"

Connecting to Essbase (CubeQuery only)

This section shows how to create a new Interactive Reporting connection file. When you query against an Essbase cube, by default the Interactive Reporting document is created in the CubeQuery section. To use these instructions you must have access to an Essbase 7.x or 9.x server used as a data source. In addition Interactive Reporting requires an Essbase 9.3 or greater Runtime client installation.

Note: Be sure the option to use the pre 9.3.1 OLAPQuery section has been disabled. Select Tools, then Options, then Program Options , then OLAP, and disable the Use Old OLAP Query section option.

➤ To connect to Analytic Server:

1 Select **Tools**, then **Connection**, and then **Create**.

   The first Database Connection Wizard is displayed.

2 From the **What connection software do you want to use?** drop-down, select **Essbase**.

3 From the **What type of database do you want to connect to?** drop-down, select **Essbase**.

4 Click **Next**.

   The second Database Connection Wizard is displayed.

5 Enter a user name in the **User Name** field.

6   Enter a password in the **Password** field.

7   Enter the host (server) name in the **Host** field.

8   Click **Next**.

The third Database Connection Wizard is displayed.

9   From the **Application/Database** drop-down, select a cube and click **Next**.

The fourth Database Connection Wizard is displayed.

10   From the **Dimension Name** drop-down, select the measure dimension and click **Next**.

Measures are numeric values in a database cube available for analysis. Measures could be margin, cost of goods sold, unit sales, budget amount, and so on.

The fifth Database Connection Wizard is displayed.

11   Click **Finish**.

12   When you are prompted to save the OCE (Interactive Reporting database connection file), click **yes**.

The Save Open Catalog dialog is displayed.

13   Enter a file name and click **OK**.

A new Query section is created in a blank Interactive Reporting document file.

Modifying Interactive Reporting Database Connection Files

When you create an Interactive Reporting database connection file, you establish a working database connection file for data modeling and querying. You may need to modify an Interactive Reporting database connection file to reflect changes in the network or hardware configuration, or to manage other connection information.

Note: Changes to basic connection configuration, such as new database or host name, require you to log off and rebuild the Interactive Reporting database connection file.

➤ To modify an Interactive Reporting database connection file:

1   Close any open Interactive Reporting document files.

2   Select **Tools**, then **Connection**, and then **Modify**.

The Modify Connection dialog box is displayed.

3   Select the connection file to modify and click **Open**.

The Database Connection Wizard is displayed showing the information for the Interactive Reporting database connection file you selected.

4   Make any desired changes and then save the Interactive Reporting database connection file when prompted.

Connecting to Databases

In Interactive Reporting Studio, you use an Interactive Reporting database connection file (.oce) to perform tasks that require you to connect to a database, such as:

- Downloading a Data Model

- Processing a query to retrieve a data set

- Showing values for a server limit

- Using server functions to create computed items

- Scheduling an Interactive Reporting document file.

The way you select an Interactive Reporting database connection file depends on which edition of Interactive Reporting Studio you are using and the data model orInteractive Reporting database connection file with which you are working.

If a data model is present in the Query section, Interactive Reporting Studio automatically prompts you with the correct Interactive Reporting database connection file when your actions require a database connection.

When you open Interactive Reporting Studio to begin a work session (for example, by downloading a data model from an Interactive Reporting Studio repository, or creating a data model from scratch) you must select the correct Interactive Reporting database connection file for the targeted database.

Monitoring Connections

Before you attempt to connect to a database, make sure you are not already connected.

You can monitor the current connection status by observing the connection icon, , on the lower right side of the Status bar. An X over the icon, , indicates there is no current database connection.

To check the connection information, position the cursor over the connection icon. The Interactive Reporting database connection file in use and database name is displayed on the left side of the Status bar.

Connecting with a Data Model

Once a *data model* is downloaded to or created in theInteractive Reporting document file, theInteractive Reporting document file is associated with theInteractive Reporting database connection file used to create the data model. Interactive Reporting document files store a reference that calls the associated Interactive Reporting database connection file whenever you need to log on to the database to build or process a query.

➤ To log on to a database from an existing Interactive Reporting document file:

1   Select **Tools**, then **Connection**, and then **Logon** or double-click the connection icon on the Status bar.

The Interactive Reporting database connection dialog box is displayed with the Interactive Reporting database connection file name in the title bar.

2   Enter the user name and password and click **OK**.

Connecting Without a Data Model

Interactive Reporting Studio users have the option of creating new data models in an empty Interactive Reporting document file. Other users download prebuilt data models from the repository.

In either situation, you select anInteractive Reporting database connection file and connect to a database before you proceed. The database you select contains either the source tables for the data model you plan to create, or the repository that contains the data models you need to download.

➤   To select an Interactive Reporting database connection file when you create a new Interactive Reporting database connection file:

1   Select **File**, then **New** to display the New File dialog box.

2   Select the **Recent Database Connection Files** radio button and select an Interactive Reporting connection file from the list, then click **OK**.

If the Interactive Reporting database connection file is not displayed, click *Browse* to display the Select Connection dialog box. Navigate to the Interactive Reporting database connection file and click *Open*.

Interactive Reporting Studio prompts you for a user name and password.

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

If you do not have the rightInteractive Reporting database connection file for a particular database, ask the database administrator to provide one or help you create an Interactive Reporting database connection file.

Note: You can create new blank Interactive Reporting document file s without connecting to a database. Blank Interactive Reporting document files are useful for importing data files such as Excel spreadsheets; for creating Dashboard master Interactive Reporting document files and for performing tasks you do not necessarily associate with a database.

Setting a Default Interactive Reporting Database Connection

If you log on to one database more frequently than others, you should set the Interactive Reporting database connection file for that particular database as the default connection. Whenever you log on to create a new data model, the default Interactive Reporting database connection file loads automatically.

If you frequently use different databases in the work, you may not want to set a default Interactive Reporting database connection. If you leave the default Interactive Reporting database

connection file preference blank, Interactive Reporting Studio prompts you to select an Interactive Reporting database connection file each time you log on.

➤ To set a default Interactive Reporting database connection file:

1 Select **Tools**, then **Options**, and then **Program Options**.

   The Interactive Reporting Studio Options dialog box is displayed.

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

3 In **Connections Direction**, enter the default connection directory that contains the Interactive Reporting database connection files you use to connect to different databases and click **OK**.

4 In **Default Connection**, enter the full path and file name of the Interactive Reporting database connection file used as the default connection.

   The next time you log on (and create a new Interactive Reporting document file), the default Interactive Reporting database connection file is automatically used.

   Store your default Interactive Reporting database connection file in your connections directory so that Interactive Reporting Studio can find them when you or users of your distributed Interactive Reporting document files attempt to log on.

Logging On Automatically

Interactive Reporting provides an Auto Logon feature that maintains the current Interactive Reporting database connection file. Auto Logon is enabled by default.

➤ To toggle Auto Logon:

1 Select **Tools**, then **Options**, and then **Program Options**.

   The Interactive Reporting Studio Options dialog box is displayed.

2 Click the **General** tab to display the General tab.

3 Select the **Auto Logon** check box and click **OK**.

➤ To use Auto Logon when creating a new Interactive Reporting document file:

1 Select the connection icon on the Status bar to verify that Interactive Reporting Studio is connected to the database.

2 Select **File**, then **New**.

   The Auto Logon dialog box is displayed.

3 Click **Yes** to accept the existing connection.

   Interactive Reporting Studio opens the new Interactive Reporting document file. If Auto Logon was accepted, you are connected to the database server automatically. Otherwise, you can select a different Interactive Reporting database connection file.

Using the Connections Manager

The Connections Manager enables you to view the status of all Interactive Reporting database connection files in all open Interactive Reporting document files. Use the Connections Manager to check or change the status of an Interactive Reporting database connection file, to modify connection preferences, or to change database passwords.

The Document Connections frame of the Connections Manager lists each open Interactive Reporting document file and its associated Interactive Reporting database connection file:

- **Connection**—Name of the selected Interactive Reporting database connection file

- **Status**—Connection status (connected or disconnected)

- **Used By**—Name of the Interactive Reporting document section that accesses the database

Use the plus (+) and minus (−) signs to navigate through the tree structure.

Logging On to a Database

➤ To log on to a database:

1 Select **Tools**, then **Connections Manager** or press **F11**.

   The Connections Manager dialog box is displayed.

2 Select an Interactive Reporting database connection file and click **Logon**.

   The Database Password dialog box is displayed.

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

   Once connected, the *X* is removed from the connection icon on the tree.

Logging Off of a Database

➤ To log off of a database:

1 Select **Tools**, then **Connections Manager** or press **F11**.

   The Connections Manager dialog box is displayed.

2 Select the Interactive Reporting database connection and click **Logoff**.

Modifying an Interactive Reporting Database Connection File Using the Connections Manager

You can use the Connections Manager to change your Interactive Reporting database connection file preferences, depending on the database and connection software.

Note: If you are not familiar with the preferences and their effects, ask the database administrator for assistance before changing the default settings.

➤ To modify an Interactive Reporting database connection file using the Connections Manager:

1 Select **Tools**, then **Connections Manager** or press **F11**.

   The Connections Manager dialog box is displayed.

2 Select the Interactive Reporting database connection file and click **Modify**.

   The Database Connection Wizard is displayed showing the information for the Interactive Reporting database connection file.

3 Make any desired changes and save the Interactive Reporting database connection file when prompted.

Changing Data Passwords

You can change the database password if you are connected to any of these database servers in Interactive Reporting Studio: Essbase, Teradata, Oracle, Red Brick Warehouse, Microsoft SQL Server, or Sybase Server.

➤ To change the password:

1 Select **Tools**, then **Connections Manager** or press **F11**.

   The Connections Manager dialog box is displayed.

2 Select the Interactive Reporting database connection file associated with the database to change and click **Change Database Password**.

   The Change Password dialog is displayed.

3 Enter a user name in the **Username** field.

4 Enter the current password in the **Current Password** field.

5 Enter the new password in the **New Password** field.

6 Enter the new password in the **Verify New Password** and click **OK**.

Note: Some database servers support case-sensitive passwords and/or require a minimum password length. For more information, see the documentation for the database server.

Changing Database Passwords

You can change the database password if you are connected to any of these database servers in Interactive Reporting Web Client: Oracle Essbase, Oracle, Red Brick Warehouse, Microsoft SQL Server, or Sybase Server.

➤ To change the password:

1 Select **Tools**, then **Change Database Password.**

The Change Password dialog box is displayed.

2 Enter the user name in the **Username** field.

3 Enter the current password in the **Current Password** field.

4 Enter the new password in the **New Password** field.

5 Enter the new password again in the **Verify New Password** field.

> **Note:** Some database servers support case-sensitive passwords and/or require a minimum password length. For more information, see the documentation for the database server.

Working with an Interactive Reporting Document and Connecting to a Database

Interactive Reporting document files consolidate instructions and specifications for querying, limiting, sorting, and computing data stored on the database server. An Interactive Reporting document file is centered on a Data Model and queries. You can build Data Models and queries yourself using Interactive Reporting Studio, or download shared Data Models from a document repository into an Interactive Reporting document file.

Interactive Reporting document files are associated with a database server by way of a separate Interactive Reporting database connection file. You use a separate Interactive Reporting database connection file for each database server. Each Interactive Reporting database connection file retains routines, instructions, protocols and parameters in a small file, also called an Open Catalog Extension (.oce). TheInteractive Reporting database connection file also preserves DBMS-specific connection preferences and specifications for automatic access to metadata.

For Interactive Reporting Studio users, the process of creating a new Interactive Reporting document file and logging onto a database is simple. You select an Interactive Reporting database connection file for the database server to use and enter the database password. You can select either a new or an existing Interactive Reporting database connection file.

➤ To create a new Interactive Reporting document using an existing Interactive Reporting database connection file:

1 Select **File**, then **New.**

The New File dialog box is displayed.

2 From **Recent Database Connection Files**, select an Interactive Reporting database connection file from the list.

3 If the connection file is not displayed, click **Browse** to display **Select Connection.** Navigate to the Interactive Reporting database connection file and click **Open.**

The Connection Password dialog box is displayed.

4 Type your user name in the **Host User** field, password in the **Host Password** field, and click **OK**.

If you do not have the right Interactive Reporting database connection file to connect to a particular database, ask your administrator to provide or help you create a connection file.

➤ To create a new Interactive Reporting document file using a new Interactive Reporting database connection file:

1 Select **File**, then **New**.

The New File dialog box is displayed.

2 Select **A New Database Connection File** field, and click **OK**.

The Database Connection Wizard is launched.

3 Follow the instructions provided by the Database Connection Wizard.

➤ To create a blank Interactive Reporting document file with no Interactive Reporting database connection file:

1 Select the **Other** check box.

2 Select the Blank Document field and click **OK**.

Blank Interactive Reporting document files are useful for importing data files such as Excel spreadsheets; for creating a Dashboard master Interactive Reporting document; and performing tasks you don't normally associate with a database.

➤ To open a Interactive Reporting document file, select **Open an Existing Document** and select an Interactive Reporting document file from the **Recent Database Connection Files** list.

If the Interactive Reporting document file is not displayed, click *Locate File*. When the browse box is activated, click the Browse button to display the Open File dialog box. Navigate to the Interactive Reporting document file and click *Open*. Interactive Reporting document files are saved with a .bqy extension on Windows.

➤ To select an Interactive Reporting database connection files from the document repository:

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

The Select Connection dialog box is displayed.

2 Navigate to the Interactive Reporting database connection file and click **Open**.

When querying the database, you first select the data items that interest you from a Data Models, Standard Query or Standard Query with Reports. You can find a repository object to start with by selecting one from the Repository Catalog and downloading it to the desktop. When you download the object to the Contents frame, the object becomes the basis of a new Interactive Reporting document file.

3 If you are not connected, log on to the database containing the document repository by selecting an Interactive Reporting database connection file from the Select Connection dialog box and entering your database user name and password.

The Open from Repository dialog box is displayed.

The Open from Repository dialog box shows the Repository Catalog in the left frame and description information in the right frame. The Repository Catalog is in directory tree format, which enables you to navigate through the repository structure.

Repositories are organized into subdivisions, which depending on the database may have subdivisions called databases, and most likely have subdivisions called owners. Databases and owners can be departmental headings, people in your organization, or other criteria established by the administrator.

You cannot access versions 4.0 and older of the repository.

4   Under each owner name in the repository, there are user groups.

User groups are established by an advanced user to categorize and store repository objects by content and access privileges. You have been granted access to only the items you see in the Repository Catalog.

5   Select the document icons in the directory tree to display profiles in the Model Info and Description Areas to the right.

6   When you have navigated to the correct repository owner and user group and found the repository object, select the object in the directory tree and click **Open**.

Interactive Reporting downloads the repository object to the appropriate section.

Connecting to Interactive Reporting Web Client

Connections made through Interactive Reporting Web Clients and the EPM Workspace can be made immediately to a database, or deferred until a query is actually processed.

Note: A locally saved Interactive Reporting document file does not prompt to connect to the EPM Workspace when opened by dragging the Interactive Reporting document file into a Web browser. A message is displayed stating that the Interactive Reporting document file is opening in offline mode. This is part of Windows XP SP2's new pop-up blocker feature. The workaround is to disable the pop-up blocker. In Microsoft Internet Explorer, select Tools, then Pop-Up Blocker, then Turn Off Pop-up Blocker.

Note: Firefox does not invoke Interactive Reporting Web Client if: the Interactive Reporting document file name contains double-byte characters or characters from a code page different from the system code page in its file name. In addition, multilingual file names are not recognized by Firefox.

➤ To select a Web client connection method:

1   Select **Tools**, then **Connect** or press **F11**.

The Connect drop-down box is displayed.

2   Select **Web Clients**.

The Web Clients dialog box is displayed.

3 **Select the connection method for the web client:**

- **Immediately connect to database**—Select this method to immediately connect to a database using genuine database authentication. You are prompted for the logon credentials to the database being accessed. The value set here for the Interactive Reporting document file in Interactive Reporting Studio cannot be changed in Interactive Reporting Web Client. This connection method is the preferred method for Interactive Reporting document files created in Interactive Reporting version 8.2 and later.

- **Defer connection to database until used to process SQL**—Select this method to defer making a connection to a database until the query is processed. You are prompted for logon credentials to the database without using genuine database authentication. That is, no actual Interactive Reporting database connection file is attempted until the query is processed.

Deferring Interactive Reporting Web Client Connections

When an Interactive Reporting document file is opened in Interactive Reporting Web Client, a connection is made to the database immediately using the criteria set in the Interactive Reporting database connection file. This has been the default behavior for the Interactive Reporting Studio and Interactive Reporting Web Client since Release 8.2. Interactive Reporting Web Client optionally enables you to defer the connection to the database until the query is processed. That is, the credentials for the user are obtained following the rules established by the EPM Workspace and by the publishers of the Interactive Reporting document file and document content. For example, if the Interactive Reporting database connection file associated with the query is set to prompt the user when the Interactive Reporting document file was published, the user is prompted; if at publishing time the credentials were supplied (Specify Now), then those credentials are used, and so on. If you defer the connection to the database, the deferment is not saved with theInteractive Reporting document file. When it is reopened, an immediate attempt to connection to the database is made.

Note: Deferring the connection to the database in Interactive Reporting Web Client, can only be set in the Interactive Reporting Studio. The setting only affects Interactive Reporting document file opened in Interactive Reporting Web Client, and not in the Interactive Reporting Studio

➤ To defer the connection to the database for an Interactive Reporting document file in Interactive Reporting Web Client:

1 Select **Tools** , then **Connection**, and then **Web Client**.

The Web Client dialog box is displayed.

2 To defer the connection to the database until the SQL is processed, select **Defer connection to database until used to process SQL. This was the default behavior before 8.2** and click **OK**.

3 **Optional**. To reset the connection option to the default setting, select **Immediately connect to database. (This is the default behavior for version 8.2 and later and is recommended** and click **OK**.

Connecting to EPM Workspace

Use the Connect to Server dialog box to specify the Data Access Servlet URL required to launch the EPM Workspace. The EPM Workspace consists of services, applications, and tools for those users who need to find and view Interactive Reporting document files and for users who need to import files, schedule jobs, and distribute the output.

➤ To connect to the EPM Workspace:

1 Select **Tools**, then **Connect to Interactive Reporting - System 9**.

The Connect to Server dialog box is displayed.

2 Specify the Data Access Servlet URL required to launch the EPM Workspace in the **Server Address** field.

17

Bidirectional Text

About Bidirectional Text

Bidirectional text is text containing both right-to-left and left-to-right directional runs. Bidirectional text support is available on all platforms, in the user interface and in all exported document files:

● 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 file is opened. Due to this behavior, text rendering may differ from what is presented in the Interactive Reporting Studio or EPM Workspace.

● The system-defined bidirectional behavior of the following user interface controls is not changed: edit, drop-down, and list-box. This is true both for clients (Interactive Reporting Studio and Interactive Reporting Web Client) and the EPM Workspace.

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.

הרב שלמה אבינר נולד בשנת 1943 בעיר ליון בצרפת.

You read this sentence starting from the right side beginning with:

הרב שלמה אבינר נולד בשנת

next reading number 1943, then reading:

בעיר ליון בצרפת.

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.

18

Using Metatopics and Metadata

About Metatopics and Metadata

Metatopics and metadata enable advanced users to mask the more confusing technical aspects of databases for non-technical users. While data models are already simplified views of the database, they sometimes still present a challenge to novice users. This is especially true when confusing database names and complicated strategies are visible in the data model.

For most end users, the confusing aspects of query building stem from two sources:

● Data model topic and join structures

● Database naming conventions

Interactive Reporting Studio provides two solutions to deal with each of these problems. These complementary solutions can be integrated to shield company personnel from the technical aspects of the query process and make end-user querying completely intuitive:

● **Metatopics**—Topics created from items in other topics. Metatopics are higher level topics, or "virtual topics" that simplify the data model structure and make joins transparent. A metatopic looks and behaves like any other topic and can accept modifications and metadata.

● **Metadata**—Data about data. Typically stored in database tables, and often associated with data warehousing, metadata describes the history, content, and function of database tables, columns, and joins in understandable business terms.

  Metadata is useful for overcoming the awkward names or ambiguous abbreviations often used in a database. For example, for a database table named CUST_OLD, metadata can

substitute a description business name for the table, such as "Inactive Customers," when it is viewed by the end user. Metadata may also include longer comments.

Because most business maintain their metadata on a database server, it is a potentially useful guide to the contents of the database, if it can be synchronized and used in conjunction with the data it describes.

Data Modeling with Metatopics

As noted earlier, metatopics allow you to create higher level topics that can greatly simplify the appearance of a data model.

Unlike other topics, metatopics are independent of actual database tables. You can use metatopics to make the column and join structure of an underlying database transparent. You can substitute instead streamlined and intuitive topics adapted to the way users conceptualize information.

For example, you can replace a data model of joined topics with a single metatopics that contains only the items business personnel need in their queries. The joins are completely transparent.

Tip: Metatopics do not support detail view.

Creating Metatopics

You can create a new, empty metatopic or copy an existing topic to use as the basis for a metatopic.

➤ To create a new, empty metatopic:

1   Select **DataModel**, then **Add Metatopic.**

2   Type the name of the new topic in the Topic Properties dialog box and click **OK.**

➤ To create a metatopic from an existing topic:

1   Select a topic in the Content frame.

2   Select **DataModel**, then **Promote to Metatopic.**

A new metatopic is displayed in the Content frame with the default name: Meta_TopicName. The new topic contains the same items defined in the source topic.

Copying Topic Items to a Metatopic

After you create a metatopic, you can rebuild its structure by copying topic items from other topics. Once the topic items are in place, you can view the data model solely at the metatopic level, excluding the original topics in favor of a single metatopic or multiple unjoined metatopics.

➤ To copy items from other topics to a metatopic, select the item that you want to add from an existing topic and drag it to the metatopic.

➤ To select and drag multiple topic items from the same topic, press and hold down the modifier key **Alt** for Windows, [**Option**] for MAC or **Ctrl+Alt** for Motif while using the mouse.

Note: You can select items from only one topic at a time.

Caution! If a metatopic contains items copied from an original source topic, do not remove the original topic or use the icon view. Because metatopic items model data through the original source topics, removing the original source topics or using an icon view also removes the copied topic items from the metatopic.

Creating Computed Metatopic Items

You can customize metatopics by adding *computed items* that do not exist in the database. Computed metatopic items provide end users with access to information they need without storing the data in the database or forcing them to master complicated computations.

Computed Metatopic items can be calculated by either the database server or locally. Locally computed metatopic items are restricted to reference items drawn from the metatopic where the item is placed. Server computed items can reference any items in the original topics or metatopics of the data model.

➤ To create a computed metatopic item:

1 Select the metatopic for which you want to create a computed metatopic item.

2 Select **Data Model**, then **Add Metatopic Item**, and then **Server** or **Local**.

   The server or local version of the Modify Item dialog box for computed items is displayed.

3 Enter a descriptive item name in the Name field.

4 Type or use the following buttons to create computed item expression:

● Functions button—Applies scalar functions to data items.

● Reference button—Adds Request Items to the expression.

● Options button—Specifies a data type.

● Operator buttons—Adds logical and arithmetic operators to the expression.

Customizing or Removing Metatopics and Metatopic Items

You can apply the same customization options that you use to make original topics and items more intuitive to metatopics and metatopic items.

➤ To remove a metatopic or metatopic item, use one of these options:

- Select the metatopic or topic item that you want to remove and select **Remove** on the shortcut menu.

- Press **Del.**

- Press the Delete button.

> **Caution!** If you remove a metatopic item, it cannot be restored to the metatopic. You must copy the item back to the metatopic or recreate it.

Viewing Metatopics

There are a number of ways to view a data model. By default, database-derived source topics and any metatopics you have created are displayed together in the Content frame in Combined view.

➤ To change the data model view, select **DataModel**, then **Data Model View**, and then **Option**.

Options include:

- **Combined**—Displays both original (database-derived) and metatopics in the Content frame.

- **Original**—Displays only database-derived topics in the Content frame.

- **Meta**—Displays only metatopics in the Content frame.

> **Caution!** If an original topic contains items that have been copied to a metatopic, do not iconize or remove the original topic from the Content frame in Combined view. Metatopic items are based on original items and remain linked to them. If an original topic is iconized or removed, any metatopic items based on its contents become inaccessible.

MetaData in Interactive Reporting Studio

Interactive Reporting Studio utilizes available metadata to simplify data models. By applying metadata naming conventions and descriptive information, metadata makes the information locked away in database tables and columns more accessible.

Metadata can be applied in several ways in Interactive Reporting Studio. If you have a source of metadata stored on a database server, Interactive Reporting Studio users can use the Open Metadata Interpreter to link it to data models and automatically apply the metadata information.

The data modeling features of Interactive Reporting Studio provide ways to add the benefits of metadata if you don't have a centralized metadata source. Interactive Reporting Studio automatically makes topic and item names more intelligible, and enables you to customize and change the appearance of these entities.

Using the Open Metadata Interpreter

The Open Metadata Interpreter is a powerful tool you can use to link Interactive Reporting Studio to metadata, or information about the database. By modifying the SQL that Interactive Reporting Studio sends to the database server, you can dictate where Interactive Reporting Studio finds the information it uses to create a data model from database tables.

The Open Metadata Interpreter enables Interactive Reporting Studio users to draw this information from an enterprise source of business metadata.

The Open Metadata Interpreter reads metadata from tables on a database and applies it to data models through a live database connection. The specifications for reading these tables are stores in Interactive Reporting database connection files. Once configured, metadata definitions are available to anyone who uses theInteractive Reporting database connection files.

Accessing the Open Metadata Interpreter

The Open Metadata Interpreter (OMI) is a feature of Interactive Reporting database connection files—files that enables Interactive Reporting Studio to manage database connectivity. OMI is implemented using the Metadata Definition dialog box of the Database Connection Wizard.

➤ To open the Metadata Definition dialog box:

1 If Interactive Reporting Studio is not connected to a database, select the Interactive Reporting database connection files that you want to direct to the metadata source and log on.

2 Select **Tools**, then **Connection**, and then **Modify**.

The Database Connection Wizard is launched with the Meta Connection Wizard displayed.

3 Select whether to run the Meta Connection Wizard on the current connection or on a different connection.

If you select a different connection, the Select Metadata Interactive Reporting database connection field becomes active.

a. Enter the full path and file name of the Interactive Reporting database connection file that you want to use. You can also click *Browse* to navigate to the location of the Interactive Reporting database connection file.

b. Click **Next**.

The Password dialog box is displayed.

c. Enter the database name in the **Host Name** field and the database password in the **Host Password** field and click **OK**.

d. Select the current database name and password to make the metadata connection or to specify an alternate name and password.

If you specify an alternate user name and password, enter the name and password that you want to use for the metadata connection.

4 Click **Next**.

5 Select the metadata schema where the meta settings are stored from the drop-down box.

Metadata schema are provided by third party vendors and saved in the `bqmeta0.ini` file. When you select a metadata schema, the predefined schema populates the fields in the Metadata Definition dialog box and is saved to the Interactive Reporting database connection file. If you select another schema, the metadata definitions are overwritten in the Interactive Reporting database connection file.

If you want to customize the metadata settings, select **Custom** from the drop-down box and click **Edit**. The Metadata Definition dialog box is displayed, which contains tabs for tables, columns, joins, lookup, and remarks.

For detailed explanations of the metadata definitions, see "Configuring the Open Metadata Interpreter" on page 520.

6 Enter the schema name or owner of the metadata repository table (for custom settings) or click **Next** to complete the Meta Connection Wizard and return to the Data Connection Wizard.

Configuring the Open Metadata Interpreter

The Open Metadata Interpreter is implemented using the Metadata Definitions dialog box. You add metadata definitions in the Metadata Definition dialog box, which contains five tabbed pages.

The pages can be independently configured and are designed to assist you in creating SQL Select statements to extract and apply metadata from predefined source tables or provided by third party vendors.

Radio buttons at the top of the certain pages enable you to specify naming based on actual default table and column names, or a custom metadata source. When the custom option is selected, the SQL entry fields on the tab are activated, and you can enter SQL statements into the separate metadata definition areas.

Metadata Definition: SQL Entry Fields

Each Metadata Definition tab has up to three Metadata Table Definition SQL entry fields:

● **Select**—Generates SQL Select statements, and is divided into distinct fields which specify the columns that store the metadata. The columns are located in the database table described in the From field. If necessary, you can use aliases in the Select fields to distinguish between multiple tables.

- **From**—Generates an SQL From clause, and specify the table(s) that contains metadata that applies to the database item described by the tab. You can also enter SQL to access system tables when necessary. If you need to reference more than one table in the From field, you can use table aliases in the SQL.

- **Where**—Generates SQL Where clauses and is used on the Columns and Joins pages to indicate which topic needs to be populated with item names or joined to another topic. It can also be used to establish relationships between multiple tables or filter tables.

Notes on Entering SQL:

- Entries are required in all From entry fields, and in all fields marked with an asterisk (*).

- Under default settings, Metadata Definition fields specify the system-managed directory tables (except when using ODBC). You cannot modify field values when the Default radio button is selected.

- :TABLE and :Column should not show quotes in the metadata set up. SQL adds the quotes automatically.

- Clicking Reset at any time when defining a custom source populates the entry fields with the database default values. It may be helpful to start with the defaults when setting up metadata definitions.

- You may sometimes use database variables when entering a Where clause. Interactive Reporting Studio provides :OWNER, :TABLE, :COLUMN, :LOOKUPID, :TABALIAS, and :COLALIAS variables which temporarily store a database owner, table, column, or domain ID number and aliases of the active topic or item. Each variable must be entered in all caps with a leading colon.

Metadata Definition: Tables

Extracting and applying metadata to topics is the simplest metadata configuration. When metadata is defined for database tables, they display in the Table catalog with the names supplied in an alternate "table of tables," and topics drawn from the tables are renamed to reflect the metadata as well.

Once the Table tab is configured, all data models using the connection apply metadata names instead of the default server name to topics in the Content frame.

➤ To apply metadata names to data model topics:

1 On the **Tables** tab, select **Custom Definition**.

   The SQL entry fields activate and the system-managed information clears. Click *Reset* if you want to use the database default as a starting point.

2 In the **Select** fields, enter the appropriate column names as they are displayed in the alternate table of tables.

   - **Owner Name**—Name of the owner column in the alternate table of tables

- **Physical Table Name**—Name of the column of physical table names in the alternate table of tables

- **Table Alias**—Name of the column of metadata table aliases in the alternate table of tables

- **Table Type**—Name of the column of physical table descriptions in the alternate table of tables

3 In the **From** field, enter the physical name of the alternate table of tables.

4 Use the Where fields to filter selected topics (for example, to limit the metadata mapping to include only certain owners).

> **Note:** If multiple folders exist in the repository, the following modifications are necessary to the Interactive Reporting Studio `bqmeta0.ini` file in order to filter the list of tables by folder:

➤ To filter Informatica tables:

1 Under the heading labeled [Informatica], change the TableWhere property as follows (do not include brackets): `TableWhere=SUBJECT_AREA='<folder name>'`

2 Change the ColumnWhere property as follows (do not include brackets):
`ColumnWhere=table_name = ':TABLE' and SUBJECT_AREA='<folder name>'`

Metadata Definition: Columns

On the Columns tab, you need to specify the topics in which items should display. You may also need to refer to the system-managed table of columns (in addition to the alternate table of columns) for some specific column information. Once you configure the Columns tab, all data models using the connection apply metadata to topic items in the Content frame instead of using default server names.

➤ To apply metadata names to data model topic items:

1 On the Columns tab, select **Custom Definition**.

The SQL entry fields activate and the system-managed information clears. Click *Reset* if you want to use the database defaults as a starting point.

2 In the **Select** fields, enter the appropriate column names as they are displayed in the alternate table of columns and/or system-managed table of columns.

- **Physical Column Name**—Name of the column of physical column names in the alternate table of columns

- **Column Alias**—Name of the column of metadata column aliases in the alternate table of columns

- **Column Type**—Name of the column of column data types

- **Byte Length**—Name of the column of column data lengths

- **Fraction**—Name of the column of column data scales

- **Total Digits**—Name of the column of column precision values

- **Null Values**—Name of the column of column null indicators

   If you use more than one table in the From field, enter the full column name preceded by a table name in the Select field.

   ```
   table_name.column_name
   ```

3 In the **From** field, enter the physical names of the alternate table of columns (and system-managed table of tables, if necessary).

   If you are using both tables in the From field, you can simplify SQL entry by using table aliases.

4 Use the **Where** field to relate columns in the alternate and system-managed tables of tables to ensure metadata is applied to the correct columns.

   Use the following syntax in the Where field (do not include brackets):

   ```
   <table of columns>.<tables column>=':TABLE' and <table of columns>.<owners
   column>=':OWNER'.
   ```

   Interactive Reporting Studio automatically populates a topic added to the Content frame with the metadata item names when it finds rows in the alternate table of columns that match the names temporarily stored in :TABLE and :OWNER. Use also the variables :TABALIAS and :COLALIAS to specify table and column aliases in SQL.

   Note: The database variables must be entered in upper case and preceded with a colon.

Metadata Definition: Joins

You can use the auto-join feature to automatically join topics based not only on the best guess of Interactive Reporting Studio, but also on primary and foreign key information stored in an alternative table of joins. Join strategies include:

- **Best Guess**—Joins columns of similar name and data type.

- **Custom**—Selects joins defined in a custom metadata source.

- **Server-Defined**—Uses joins that have been established on the database server.

The Joins tab uses SQL instructions to employ a custom join strategy stored in metadata. Once Interactive Reporting Studio is directed to the metadata source, all data models using the connection apply specified join logic between topics.

➤ To automatically join topics using metadata join information:

1 On the Joins tab, select **Custom**.

   The SQL entry fields activate. (There are no system defaults for the Joins tab.) Click *Clear* to clear the entry fields if you make a mistake and want to start over

2 In the **Select** fields, enter the appropriate column names as they are displayed in the alternate table of joins. Interactive Reporting Studio requires data in the Primary Table and Primary Column fields to find the primary keys.

- **Primary Database Name**—Sets the name of the column of databases for primary key tables in the alternate table of joins.

- **Primary Owner**—Sets the name of the column of owners belonging to primary key tables in the table of joins.

- **Primary Table**—Sets the name of the column of primary key tables in the table of joins.

- **Primary Column**—Sets the name of the column of primary key items in the table of joins.

- **Foreign Database Name**—Sets the name of the column of databases for foreign key tables in the alternate table of joins.

- **Foreign Owner**—Sets the name of the column of owners belonging to foreign key tables in the table of joins.

- **Foreign Table**—Sets the name of the column of foreign key tables in the table of joins.

- **Foreign Column**—Sets the name of the column of foreign key items in the table of joins.

  If you use more than one table in the From field, enter the full column name preceded by a table name in the Select fields.

  ```
  table_name.column_name
  ```

3 In the **From** field, enter the physical name of the alternate table of joins.

4 Use the Where field to tell Interactive Reporting Studio which topics to auto-join.

Use the following syntax in the Where field (do not include brackets):

```
<owners column>=':OWNER' and <tables column>=':TABLE'
```

If Auto-Join is enabled, Interactive Reporting Studio automatically joins topics added to the Content frame when it finds rows in the alternate table of joins that match the names temporarily stored in :TABLE and :OWNER. You can also use the variables :TABALIAS and :COLALIAS to specify table and column aliases in the SQL.

Note: The database variables must be entered in upper case and preceded with a colon.

Metadata Definition: Lookup

Lookups apply metadata to values that are queried by the Show Values command in the limit dialog box. If the database tracks data by codes, abbreviations, or ID numbers, lookup values can help users effectively limit queries.

For example, the product table may track sales by product ID number. When the user attempts to limit the Product ID column in a query, a Show Values call to the database yields only ambiguous product ID numbers. It can be hard to tell where to apply the limit.

Using the Lookup tab, you can map the product ID values to a column of descriptive product names elsewhere in the database. When the user clicks Show Values, he or she selects among descriptive product names to set the limit on the underlying product ID numbers.

Note: To use this feature, you need a table of descriptive lookup values in the database, and an additional mapping table to verify which items are supported by lookup values and where the corresponding lookup values are stored.

➤ To apply metadata to limit lookup values:

1 On the Lookup tab, select **Use SQL Definition**.

The SQL entry fields activate. Click *Clear* to clear the entry fields if you make a mistake and want to start over.

2 In the **Select** fields, enter the appropriate column names as they displayed in the domain registry table. The Lookup Table, Lookup Value Column, Lookup Description Column, and Lookup Domain ID Column are required for Interactive Reporting Studio to locate lookup values.

   - **Lookup Database**—Name of the column of databases in the domain registry table.

   - **Lookup Owner**—Name of the column of owners in the domain registry table.

   - **Lookup Table**—Name of the column of tables containing lookup domain description values in the domain registry table.

   - **Lookup Description Column**—Name of the column of columns containing descriptive lookup values in the domain registry table.

   - **Lookup Value Column**—Name of the column of columns of original column values in the domain registry table.

   - **Lookup Domain ID Column**—Name of the column of domain IDs in the domain registry table.

3 In the **From** field, enter the physical name of the domain registry table.

Interactive Reporting Studio first sends SQL to the domain registry table to see if

Lookup values are available for a given item.

4 Use the **Where** field to identify which items have lookup values.

Use the following format (do not include brackets):

```
<tables column>=':TABLE' and <columns column>=':COLUMN'
```

When you limit an item and show values, Interactive Reporting Studio stores the physical table and column names of the item in the variables, :TABLE and :COLUMN.

Interactive Reporting Studio searches the domain registry table for a row that matches the values temporarily stored in :TABLE and :COLUMN. When it finds a row that matches, it pulls lookup values from the specified columns in the domain descriptions table. You can also use the :LOOKUPID variable to store the lookup domain ID value.

Note: The database variables must be entered in upper case and preceded with a colon.

5 Use the **Lookup Where** field to synchronize the values in the domain registry and domain description tables.

Metadata Definition: Remarks

If database remarks already exist for the database, you can configure the Interactive Reporting database connection to retrieve and display them as part of the data model.

Database remarks function like context-sensitive help by providing detailed contextual information about a table or column, and can be very helpful to users when navigating through a large data model.

The Remarks tab uses SQL instructions to direct Interactive Reporting Studio toward the unified server source of remarks for tables and columns. Once the Remarks tab is configured, all data models using the connection have access to remarks (**Query** > **Show Remarks**).

➤ To add remarks from stored metadata:

1 On the **Remarks** tab, select **Table Remarks** to set up remarks for tables, or select **Column Remarks** to set up remarks for columns.

   Click *Clear* to clear the entry fields if you make a mistake and want to start over.

2 In the **Tab Name** field, type the name of the tab that you want to be displayed in the Show Remarks dialog box.

3 In the **Select** field, enter the name of the column of table or column remarks.

4 In the **From** field, enter the physical name of the table containing table or column remarks.

5 Use **Where** to link the selected topic to its corresponding remark.

   Use the following syntax in the Where field:

   ```
   Name of the Remarks Table =:TABLE
   ```

   and

   ```
   Name of the Remarks Column=:COLUMN
   ```

   The dynamic variable automatically inserts the physical name of the object from which the user is requesting data in the application. Interactive Reporting Studio displays remarks when it finds rows in the remarks tables which match the names temporarily stored in :TABLE and :COLUMN. You can also use the variables :TABALIAS (displays name of a table) and :COLALIAS (displays name of a column) to specify table and column aliases in the SQL.

   Note: The database variables must be entered in upper case and preceded with a colon.

6 Click **Add** to add the tab to the Remarks Tabs list.

   The Remarks Tabs list shows all of the tabs you entered in the order in which you entered them. The first tab in the lists is the default or first tab to be displayed in the Show Remarks dialog box.

   Use the following buttons to reorder the appearance of Remarks tabs:

   ● **Up**—Moves a tab up one position (toward the front of the Show Remarks dialog box).

   ● **Down**—Moves a tab down one position (toward the back of the Show Remarks dialog box).

➤ To update a Remarks tab:

1 On the **Remarks tab**, select the tab from the Remarks tabs list.

   The information for the selected tab is displayed in Remarks SQL fields.

2 Enter the desired changes in the **Select**, **From**, and **Where** fields, and then click **Update**.

➤ To delete a Remarks tab, select the tab from the Remarks tabs list and click **Delete**.

19

Data Modeling

About Data Models

When you use Interactive Reporting Studio to query a relational database and retrieve information, you work with a *data model*: a focused visual representation of the actual database tables

Interactive Reporting Studio users can create data models, selectively viewing and packaging the contents of a database for querying or distribution. Distributed or shared data models are beneficial for several reasons:

● They substitute descriptive names for arcane database table and column names, enabling users to concentrate on the information, rather than the data retrieval.

● They are customized for users' needs. Some kinds of data models include prebuilt queries that are ready to process, and may even include reports that are formatted and ready to use. Other data models may automatically deliver data to users computers.

● They are standardized and up-to-date. A data model stored in the document repository can be used throughout the company and is easily updated by the database administrator to reflect changes in the database structure.

> **Note:** You can only add create and modify Data Models if you have the "Data Model, Query, and Analyze" adaptive state.

A Data Model displays database tables as topics in the Contents frame. Topics are visually joined together like database tables and contain related items used to build a query.

Multiple queries can be constructed against a single Data Model in the same Interactive Reporting document file. If you modify the Data Model, any changes are automatically propagated to the corresponding queries.

In addition to standard Data Models derived from database tables, you can create metatopics—virtual views independent of the actual database. You use metatopics to standardize complex calculations and simplify views of the underlying data with intuitive topics customized for business needs.

If you want to preserve a Data Model for future queries, you can promote it to a master data model and lock its basic property design. This feature enables you to generate future queries without having to recreate the Data Model. An Interactive Reporting document file can contain any number of master data models from which any numbers of queries can be generated.

Building a Data Model

Data models are the building blocks of queries. In a data model, database tables are represented by topics. A *topic* is a list of items, each corresponding to a column in the database tables.

Adding Topics to a Data Model

You create data models by choosing database tables from the *Table Catalog* and assembling them as topics in the Content frame. The Table catalog is a listing of the tables available in the database. Once connected to a database, you can display the Table catalog and drag the topics that you want to include in the data model to the Content frame.

➤ To add a topic to a data model:

1   In the Query section, select **DataModel**, then **Table Catalog**, or press **F9**.

    If you are not connected to the database, you are prompted to log on. Once connected, the Table catalog is displayed listing the available database tables.

    Note: Users can filter tables from the display as part of the database connection.

2   Drag tables from the Table catalog to the Content frame.

    Each database table you place in the Content frame is converted to a topic in a data model.

Removing Topics from a Data Model

➤ To remove a topic from a data model, select the topic and select **Remove** on the pop-up menu or press **Del.**

Understanding Joins

Tables in relational databases share information through a conceptual link, or *join*, between related columns in different tables. These relationships are displayed in the data model through visual join lines between topic items.

Joins enable you to connect or link records in two tables by way of a shared data field. Once a data field is shared, other data contained in the joined tables can be accessed. In this way, each record can share data with another record, but does not store and duplicate the same kind of information.

Joins can be automatically created joins for you, or you can manually join topics.

Suppose you queried only the Customers table to determine the number of customers. You would retrieve 32 records with the names of the stores that purchase products since 32 is the exact amount of stores that have made a purchase.

But suppose you made the same query with the Customers table and Sales table joined. This time you would retrieve 1,000 records, because each store made multiple purchases. Figure 1 shows the intersection of all records in the Sales table that mention stores listed in the Customers table.

Figure 1   Result of Join Between Two Tables

In other words, a database query returns the records at the intersection of joined tables. If one table mentions stores 1-32 and the other table mentions those same stores repeatedly, each of these records will be returned.

If you join still a third table, such as items, records are returned from the intersection of all three. Figure 2 shows the intersection of all records in the Sales table that have stores in the Customers table and items in the Items table.

Figure 2   Result of Join Between Three Tables

The following sections discuss the types of joins available and how to use them:

Simple Joins

A simple join between topic items, shown in Figure 3, retrieves rows where the values in joined columns match.

Figure 3   Simple Join Between Identical Store Key Fields in Two Topics

Joins need to occur between items containing the same data. Often, the item names between two topics are identical, which sometimes indicates which items join. When selecting items to join, recognize that two items may share the same name, but refer to completely different data. For example, an item called "Name" in a Customer table and an item called "Name" in a Product table are probably unrelated.

Cross Joins

If topics are not joined, a database cannot correlate the information between the tables in the data mode. This leads to invalid datasets and run-away queries. In this case, a database creates a cross join between non-joined tables, where every row in one table is joined to every row in another table.

Joining Topics

The Auto Join Tables option automatically joins database tables as they are added to the Content frame using one of three different join strategies. If Auto Join Tables is not selected, you can manually create joins between topics in the Content frame.

➤ To automatically join topics as they are added to the Content frame:

1 Select **DataModel**, then **Data Model Options**.

   The Data Model Options dialog box is displayed.

2 Select the **General** tab.

3 Select the **Auto Join Tables** check box and then click **OK**.

   When you add tables from the Table catalog to the Content frame, joins automatically display between topics.

   Clear the Auto Join Tables check box to turn off this feature and manually create joins yourself.

   Note: Joins are not added for topics that are in the Content frame before you select the Auto Join Tables option.

Specifying an Automatic Join Strategy

You can instruct Interactive Reporting Studio to use one of three different strategies when automatically joining topics. The strategy chosen is employed with a particular connection and saved with the Interactive Reporting database connection file.

➤ To select an automatic join strategy for a database connection:

1 **If you are not currently connected to the database, select an Interactive Reporting database connection file and log on.**

2   Select **Tools**, then **Connections**, and then **Modify**.

The Meta Connection Wizard is displayed with the On The Current Connection option selected.

> **Note:**   For information on metatopics and metadata, see Creating Metatopics.

3   Click **Next**.

The Meta Connection Wizard displays the repository where the meta settings are stored.

4   Click **Edit**.

The Metadata Definition dialog box is displayed.

5   Select the **Joins** tab.

6   Select a join strategy. Join strategy options are:

- **Best Guess**—Joins topics through two items that share the same name and data type
- **Custom**—Joins topics according to specified schema coded in SQL in the Metadata Join Definitions area
- **Server-Defined**—Joins topics based on primary and foreign keys established in the underlying relational database

7   When you have completed the selection, click **OK**.

Manually Joining Topics

You can create relationships between topics by manually joining topic items in the Content frame (see Figure 4).

Figure 4   Manually Created Join Between Two Related Data Items in Two Topics

➤   To manually join two topics, select a topic item, drag it over a topic item in another topic, and release.

A join line is displayed, connecting the items in the different topics.

Showing Icon Joins

When a topic is iconized, you can toggle the display of joins to other topics in the Content frame.

➤ To show icon joins:

1 Select **DataModel**, then **Data Model Options**.

   The Data Model Options dialog box is displayed.

2 Select the **General** tab to display the General tab.

3 Select the **Show Icon Joins** check box and click **OK**.

   Clear the Show Icon Joins check box to turn off this feature and hide joins of iconized topics.

Specifying Join Types

Join types determine how data is retrieved from a database.

➤ To specify a join type:

1 Select a join line and select **View**, then **Properties** or click the Properties icon.

   The Join Properties dialog box is displayed.

2 Select a join type and click **OK**.

   Four types of joins are supported:

   - **Simple join** (=, >,<, >=, <=+)—A simple (linear) join retrieves the records in both tables that have an identical data in the joined columns.

     You can change the default join setting for simple joins by choosing an operator from the drop-down box. The default setting, Equal, is preferred in most situations.

   - **Use ANSI inner join syntax**— An inner join is logically the same as a WHERE clause simple join. The INNER join syntax is ANSI92 compliant, specifying the join condition as part of the FROM clause. Like a WHERE clause simple join, the INNER join returns only those rows from the joined tables where the join condition is met. This option is specific to tables in a database and is applied when SQL is generated; it can not be chosen for local joins. It is only available with ODBC connections. The option can not be selected for individual OUTER joins; for those, use settings in the Advanced portion of the OCE wizard. The default behavior is the simple join (WHERE clause syntax). Select this option to optimize a particular query when recommended by your database administrator or database documentation. Example of SQL with this option: SELECT AL1.ACCOUNT_DESC, AL2.ACCOUNT_STATUS_CODE FROM SYSTEM.ACCOUNT AL1 INNER JOIN SYSTEM.ACCOUNTIFO AL2 ON (AL1.ACCOUNT_CODE=AL2.ACCOUNT_CODE) This same statement, using the default join logic, would have been generated as: SELECT AL1.ACCOUNT_DESC, AL2.ACCOUNT_STATUS_CODE FROM SYSTEM.ACCOUNT AL1, SYSTEM.ACCOUNTINFO AL2 WHERE (AL1.ACCOUNT_CODE=AL2.ACCOUNT_CODE)

- **Left outer join** (+=)—A left join retrieves all rows from the topic on the left and any rows from the topic on the right that have matching values in the join column.

- **Right outer join** (=+)—A right join retrieves all rows from the topic on the right and any rows from the topic on the left that have matching values in the join column.

- **Outer or full outer join** (+ = +)—An outer join combines the impact of a left and right join. An outer join retrieves all rows from both tables matching joined column values, if found, or retrieves nulls for non-matching values. Every row represented in both topics is displayed at least once.

Note: A fifth join type, Local Join, is available for use with local Results sets. See "Using Local Joins as Filters" on page 538 for more information.

Caution! Not all database servers support all join types. If a join type is not available for the database to which you are connected, it is unavailable for selection in the Join Properties dialog box.

Removing Joins

You can remove unwanted joins from the data model. Removing a join has no effect on the underlying database tables or any server-defined joins between them. A deleted join is removed from consideration only within the data model.

➤ To remove a join from a data model, select the join and select **Remove** on the shortcut menu.

Using Defined Join Paths

Defined Join Paths are customized join preferences that enables you to include or exclude appropriate tables based on the items referenced on the Request and Limit lines. "Bridge tables," which are not explicitly referenced in the query, are transparently added to the SQL From clause. The net effect limits the query to all referenced tables based on available table groupings, which generate the most efficient SQL for queries off the data model.

➤ To use defined join paths:

1 Select **DataModel**, then **Data Model Options**.

   The Data Model Options dialog box is displayed.

2 Select the **Joins** tab to display the Joins tab.

3 Select the **Use Defined Join Paths** option and click **Configure**.

   The Define Join Paths dialog box is displayed.

4 In the Define Join Paths dialog box, click **New Join Path** to name and add a join path.

   The New Join Path dialog box is displayed.

5 In the **New Join Path** dialog box, enter a descriptive name for the join path and click **OK**.

The join path name is highlighted in the Defined Join Paths dialog box.

6 Select a topic in the Available topics list and use the move right (-->) button to move it to the Topics In Join Path list.

7 To remove join paths from the Topics in Join Path list, select the move left (<--) button, to remove join paths from the Topics In Join Path list.

8 When join paths are completely defined for the data model, click **OK**.

> **Tip:** Join paths are not additive; Interactive Reporting Studio cannot determine which tables are common among several paths and link them on that basis. Join paths are not linear, and if selected, the simplest join between all tables in the path is included when processing a query.

Using Local Joins

You can add the results of one query to the results of another query in an Interactive Reporting document file by using *local joins*. Rows from the data sources are joined in the Results section.

> **Note:** No aggregation can be applied to local result tables and the local results data set cannot be processed to a table.

For example, you might want to see budget figures drawn from MS SQL server and sales figure drawn from an Oracle database combined in one Results set.

> **Caution!** Local joins are memory and CPU intensive operations. When using this feature, please limit the local joins by using a moderate number of rows.

The following sections explain how to work with local joins:

- "Creating Local Joins" on page 537
- "Using Local Joins as Filters" on page 538
- "Limitations of Local Results and Local Joins" on page 540

Creating Local Joins

➤ To create a local join:

1 Select **Insert**, then **Insert New Query** to create the first query in the Interactive Reporting document file (BQY):

The Insert Query dialog is displayed.

2 Complete the steps for inserting a new query (described in the **Insert Query** topic).

3   In the Query section, verify item data types and associated data values in source documents so that you know how to join them in the Interactive Reporting document file (BQY).

4   Build the *Request Line*, and add server and *local filters*, data functions, and computations to the query as needed.

5   Select **Process**.

> **Tip:** For consistent results, queries that use local joins should be placed after queries that generate the needed results.

6   Select **Insert**, then **Insert New Query** to create the second query.

(See step 1 for information on inserting a new query.)

Add topics from the Table catalog to the Content pane, and build the Request line.

7   In the Table catalog, select **Local Results** on the shortcut menu.

8   In the Table catalog of the second query, select **Local Results** on the shortcut menu.

A Local Results icon, displays in Elements.

9   Expand the Local Results icon to display the Results table icon, .

10   Double-click a **Results** set or drag it to the Content pane.

The Results set from the first query is displayed as a topic in the Content pane.

11   In the Content pane, manually create a join between the Results set and another topic.

12   Build the Request line and select **Process**.

Local joins are processed on the client machine.

You can use Process All to process the queries, in which case the queries are processed in the order in which they are displayed in the Section catalog.

For example, in a Interactive Reporting document file with three queries, Query1, Query2, and Query3, the queries are executed in the order shown. If Query1 is a local join of the results of Query2 and Query3, it will still be processed first. If Query2 and Query3 have existing Results sets, then the local join in Query1 will occur first, before processing Query2 or Query3. If the Results sets for either Query2 or Query3 are not available, one or both of those queries are processed first in order to get the required results.

Using Local Joins as Filters

A filter local join is a variation of a local join. Instead of independently running two queries then locally joining the data on the desktop, a filter local join runs the first query to retrieve a list of values, then uses those values to filter a column in the second query.

For example, a query may be run from an inventory table in an Oracle database to retrieve a list of part numbers that are out of stock. The resulting part number list may be used as a filter join to define the list of values retrieved from a work_in_process table in another database to determine the status of the stock replenishment.

Note: The second query could potentially be a very long SQL statement since using filter local joins generates an `SQL Having` clause for each item.

➤ To use the values retrieved from one query as filter values for another query:

1 Build the first query to include as a filter in the second query by selecting **Insert**, then **Insert New Query** .

The Insert Query dialog is displayed.

2 Complete the steps for inserting a new query (described in the Insert Query topic).

3 Verify item data types and associated data values in source documents so you will know how to join them in the second query.

4 Build the Request line, and add server filter, data functions and computations to the query as needed.

5 Select **Process**.

6

7 Select **Insert**, then **Insert New Query**.

8 Build the second query.

  a. Verify item data types and associated data values in source documents so you will know how to join them to the first query.

  b. Build the Request line, and add server and local limits, data functions, and computations to the query as needed.

9 In the Table catalog of the second query, select **Local Results** from the shortcut menu.

A Local Results icon, ⬚ is displayed in the Catalog frame.

10 Expand the Local Results icon to display the Results table icon, ▦.

11 Double-click the **Results** icon or drag it to the Content frame.

The Results set from the first query that you built is displayed as a topic in the Content pane.

Note: The purpose of embedding the Results is to obtain a list of values. Do not include and Results set topic items on the Request line. Also, do not place any limits on topic items in this Results set. must not include any fields from the embedded Results section. If you do add a topic item from or set a limit on this Results set, you will not be able to set a Limit Local join.

12 In the Content pane, manually join the Results set to a another topic in the second query.

A join line is displayed, connecting the different topics.

13 Double-click the join line that was created by joining the Results set and other topic, or click the Properties icon.

The Join Properties dialog box is displayed.

14 Select **Filter Local Join** and click **OK**.

Note: If the Filter Local Join option does not display in the Join Properties dialog box, make sure that no Results set topic items are included in the Request line and that no Filter have been placed on any Results set topic item.

15 Click **Process** to build the query and apply the filter constraint.

Limitations of Local Results and Local Joins

The following limitations apply to local results and local joins:

1. You cannot use any governors with local results topics as part of the query. The following are governors accessed from the Query Options dialog box:

   - Returning Unique Rows

   - Row limit

   - Time limit

   - Auto-Process

   - Custom Group by

2. You cannot have more than one local join per local results topic. When setting up a query using a local results topic, you cannot have more than one local join between the local results topic and another topic/local results topic.

3. You cannot set query limits on local results topic items. Limits must be set in the query/ result sections of the query that produces the local results. Attempting to set a query limit on a local results topic item invokes the following error message: "Unable to retrieve value list for a computed or aggregate request item".

4. You cannot aggregate local results tables.

5. You cannot process local results data to a table.

6. You cannot have more than one limit local join. A limit local join involves two topics, one of which is a local results topics. A local results item is used as a limit to the other topic. Attempting to define more than one limit local join invokes the following error message: "This query contains a local results object involved in a join limit. It is not possible to have other local results objects when you have a local join limit".

7. You cannot combine limit local joins with local joins. Attempting to combine a limit local join and local join invokes the following error message: "This query contains a local results object involved in a join limit. It is not possible to have other local results objects when you have a local join limit".

8. You should expect compromised performance when a query is associated with large local results sets. This is expected behavior since Interactive Reporting Studio is not a database.

9. You cannot use metatopics with local results. You cannot promote a local results topic to a metatopic or add a local results topic item as a metatopic item. The Promote To Meta Topic and Add Meta Topic Item DataModel menu options are not available for local results topics and topic items.

10. You cannot access or change properties for local results topic items. Properties include remarks, number formatting, aggregate/date/string functions, data types, and name.

11. You cannot have query request line computed columns from local results topic items. The Add Computed Item menu option is not available for local results topic items.

12. You cannot use Append Query features of unions or intersections with local results topic items. The Append Query menu option is not available when a local result topic is part of a query.

Working with Topics

There are several features that enable you to customize the appearance of topics to make them easier for end users to work with. The following sections describe how to work with topics:

- "Changing Topic Views" on page 541
- "Modifying Topic Properties" on page 542
- "Modifying Topic Item Properties" on page 543
- "Restricting Topic Views" on page 544

Changing Topic Views

You can change how you view topics in the Content frame. There are three ways to *view* topics, as shown in Figure 5:

Figure 5   Structure (1), Detail (2), and Icon (3) Topic Views

- **Structure view**—Displays a topic as a simple list of component data items. This is the default setting.

Structure view enables you to view and select individual data items to include in a query. This is the easiest view to use if you are familiar with the information that a data model, topics, and topic items represent.

- **Detail View**—Presents a topic in actual database view with a sample of the underlying data. When you change to Detail view, a small query is executed and a selection of data is loaded from the database server. The topic is displayed as a database table with each topic item displayed as a database column field.

    Detail view is useful when you are unfamiliar with a topic. You can browse the first few rows of data to see exactly what is available before adding a topic item to the query.

    Note: Detail view is not available for special items such as metatopics or computed data items.

- **Icon View**—Deactivates a topic and reduces it to an icon in the Content frame. When a topic is displayed in Icon View, associated items are removed from the Request and Limit lines. The topic is not recognized as being joined to other topics, and is temporarily removed from the data model and the SQL statement.

    If no items from a topic are needed for a particular query and the topic does not link together other topics which are in use, reduce the topic temporarily to Icon view to make large queries run faster and to consume fewer database resources.

➤ To change a topic view:

1   Select a **topic** in the Content frame.

2   Select **DataModel**, then **Topic View**, and then *View*.

    The topic is displayed in the chosen view.

    In Icon view, you can restore the topic view by double-clicking the topic icon.

Modifying Topic Properties

Use the Topic Properties dialog box to customize the way a topic and associated items are displayed in the data model. By default, items are displayed in the order in which they are defined in the underlying table, or the order in which they are added to a *metatopic*. You can change the way items are ordered or restrict their display of items within a topic.

➤ To modify topic properties:

1   In the Catalog pane, select the topic and select **View**, then **Properties**, or click the Properties icon.

    The Topic Properties dialog box is displayed.

2   Change the properties to the desired setting and click **OK**.

    Available options include:

    - **Topic Name**—Name of the topic that is shown in Elements. You can change this field to display a more user-friendly name in the Content pane.

- **Physical Name**—Full name of the underlying database table
- **Items To Display**—Topic items available for the selected topic.
  - ❍ **Hide/Show All**—Hides or actively displays all topic items.
  - ❍ **Up/Down**—Moves selected item up or down one space in the topic display.
  - ❍ **Sort**—Sorts listed items alphabetically.
- **Set As Dimension**—Defines the drill-down path or *hierarchy* for dimensional analysis as shown in the data model. This feature is used in conjunction with the Set As Fact field in the Topic Item Properties dialog box.
- **Allow Icon View**—Enables the *icon view* option for the topic
- **Allow Detail View**—Enables the *detail view* option for the topic.
- **Cause Reload**—Specifies automatic reloading of server values the next time Detail View is activated.
- **Rows to Load**—Specifies the number of rows to be loaded and displayed in Detail View.

Modifying Topic Item Properties

Topic items are discrete informational attributes of topics, such as Customer ID, Street Address, or Sales Revenue, and are the basic building blocks of a query. Topic items are organized within topics and represent the columns of data in database tables.

You can modify the names of topic items to make them easier for users to understand and set drill-down path information.

➤ To modify a topic item:

1 Select the topic item and select **View**, the **Properties** or click the Properties icon.

The Topic Item Properties dialog box is displayed, showing information about the source of the topic column in the database.

2 Change the topic item properties to the desired setting and click **OK**.

Available options include:

- **Item Name**—Displays the name of the item.
- **Set As Fact**—Eliminates items with integer or real values from a drill-down path. This feature is used in conjunction with the Set As Dimension field in the Topic Properties dialog box.
- **Information**—Additional column information from the database. Information about keys is displayed only when server-defined joins are enabled.
- **Length**—Enables you to change the string length of columns.

Restricting Topic Views

Individual topics within a data model can be restricted to control the availability of the Icon view and Detail view, or to limit the number of rows retrieved (which can consume network and server resources) for Detail view.

➤ To set access to Icon or Detail views:

1 Double-click a topic to be view-restricted.

   The Topic Properties dialog box is displayed with the view options shown toward the bottom of the dialog box. The dialog box also contains options for customizing topics.

2 Select the **Allow Icon View** or **Allow Detail View** check boxes to toggle the availability of either view.

3 If necessary, **Cause Reload** to specify loading from the server when Detail View is selected.

   New data is retrieved the next time Detail View is activated for the topic, after which Cause Reload will be toggled off automatically.

4 If desired in Detail View, enter the number of rows to be returned from the server for Detail View, and click **OK**.

   By default, the first ten rows of a table are retrieved for preview in Detail View.

Working with Data Models

You can customize data models in a number of ways. You can change how data models are displayed in the Content frame. You also can define other data model options, such as user access, feature availability, and query governors. Review the following sections for information on:

- "Changing Data Model Views" on page 544
- "Setting Data Model Options" on page 545
- "Automatically Processing Queries" on page 550
- "Promoting a Query to a Master Data Model" on page 550
- "Synchronizing a Data Model" on page 550

Changing Data Model Views

There are a number of ways to view a data model. By default, database-derived source topics and any metatopics you have created are displayed together in the Content frame in Combined view.

➤ To change the data model view, select **DataModel**, then **Data Model View**, and then **Option**

Options include:

- **Combined**—Displays both original (database-derived) and metatopics in the Content frame.

- **Original**—Displays only database-derived topics in the Content frame.

- **Meta**—Displays only metatopics in the Content frame.

Caution!   If an original topic contains items that have been copied to a metatopic, do not iconize or remove the original topic from the Content frame in Combined view. Metatopic items are based on original items and remain linked to them. If an original topic is iconized or removed, any metatopic items based on its contents become inaccessible.

Setting Data Model Options

➤ To set data model options:

1   Select **DataModel** , then **Data Model Options**.

The Data Model Options dialog box is displayed.

2   Set the desired options for the data model and click **OK**.

Note: All users have access to the join preferences, but not to the limit, query governor, or auditing features, which are designed to customize data models stored for distribution.

Before applying any new features, be aware that:

- One of the first three limit options (Show Values, Custom Values, or Custom SQL) must be enabled in order for users to apply limits in the Query section.

- Changing join usage usually changes the number of rows retrieved from the database. It also introduces the possibility that novice users may create improperly joined queries.

- If query governors are set as part of a data model, and end users set query governors on a query built from the data model, the more restrictive governor takes precedence.

The following sections provide additional information about data model options:

- "Saving Data Model Options as User Preferences" on page 545

- "Saving Data Model Options as Profiles" on page 546

- "Data Model Options: General" on page 546

- "Data Model Options: Filters" on page 547

- "Data Model Options: Auditing" on page 549

Saving Data Model Options as User Preferences

You can save the data model options you specify as default user preferences by clicking the Save as Defaults button on any of the tabs in the Data Model options dialog box.

➤ To change the defaults without affecting any existing data models (including the current one), click **Save as Defaults**, then **Cancel**.

➤ To change the defaults and apply them to the current data model, click **Save as Defaults**, then **OK**.

Note: The following data model options apply to the current data model only and cannot be saved as defaults: Topic Priority information and the Use Defined Join Paths option on the General tab.

Saving Data Model Options as Profiles

When you save data model options as default user preferences and apply them to a data model, you can save the Interactive Reporting document file for use as a *profile*. Over time, you can build a set of *profile* Interactive Reporting document files. By opening a *profile* Interactive Reporting document file and saving the options from the data model of the profile Interactive Reporting document file as defaults, users can switch between proven data model options appropriate to the task at hand.

A first time *profile* Interactive Reporting document file, created from a blank data model before saving any changes to the default settings, can be used to restore the data model options to the default settings. A more complete *profile*Interactive Reporting document file, appropriately populated with topics, can be used to promulgate data model options for the Use Defined Join Path feature.

Data Model Options: General

Use the General tab to select the following design options for the tables and the governors for the data model:

- **Design Options**
  - ❍ **Auto Alias Tables**—Enables the product to replace underscores with spaces and display item names in mixed upper/lower case when a table is added to the Content frame from the Table catalog.
  - ❍ **Auto Join Tables**—Instructs the product to automatically *join* database tables based on one of three different join strategies as they are added to the Content frame if their names and data types are identical. If Auto Join Tables is not selected, you must manually create joins between topics in the Content frame.
  - ❍ **Show Icon Joins**—Shows topic joins when a topic is in icon view (minimized). It is recommended that you activate this feature.
  - ❍ **Allow Drill Anywhere**—Activates the *drill anywhere* menu item on the menus within the Pivot and Chart sections. This option enables users to drill to any field.
  - ❍ **Allow Drill To Detail**—Activates the *drill to detail* menu item on the menus within the Pivot and Chart sections. This option enables users to query the database again once

they have reached the lowest *level* of detail; it only works if the Allow Drill Anywhere option is selected.

Note: If you want to use certain drilling options, they must be predefined in the data model itself. The ability to use pre-defined drill paths, drill-to-detail and incremental drilling all depend on the data model used by the query. They are not controlled in the individual chart or pivot. The ability to turn off drilling altogether is also found in the data model definition.

- **Governors (Interactive Reporting Studio Only)**

  ❍ **Return First _____ Rows**—Specifies a cap on the number of rows retrieved by a query against the data model, regardless of the size of the potential Results set.

    Note: All users can also set query governors, but data model options automatically override governors set at the query level. If row limits are also set at the query level, the lower number is enforced.

  ❍ **Time Limit _____ Minutes**—Specifies a cap on the total processing time of a query against the data model. Seconds are entered as a decimal number. Available for asynchronous connection API software (for example, Open Client) that support this feature.

Data Model Options: Filters

Use the Filters tab to specify filter browse level preferences and to select global filter options.

When you use Show Values to set filters, you may sometimes need to sift through a lot of data to find the particular values you need. Filter preferences enable you to dictate the way existing filters reduce the values available through the Show Values command.

For example, you want to retrieve customer information only from selected cities in Ohio. However, the database table of customer addresses is very large. Because Interactive Reporting applies a default filter preference, once you place the initial filter on State, the Show Values set returned for City is automatically narrowed to those cities located in Ohio. This saves you from returning thousands of customers, states, and from all sales regions.

You can adjust this preference so that the initial filter selection has no effect on the potential values returned for the second filter (all cities are returned regardless of state).

- **Filter Options**

  ❍ **Show Minimum Value Set**—Displays only values that are applicable given all existing filters. This preference takes into account filters on all tables and related through all joins in the data model (which could be potentially a very large and long running query).

  ❍ **Show Values Within Topic**—Displays values applicable given existing filters in the same topic. This preference does not take into account filters associated by joins in the data model.

  ❍ **Show All Values**—Displays all values associated with an item, regardless of any established filters.

Tip: When setting these preferences for metatopics, be sure to display the data model in Original view.

- **Global Filter Options** (Designer only)

  - **Show Values**—Globally restricts use of the Show Values command in the Filter dialog box, which is used to retrieve values from the server.

  - **Custom Values**—Globally restricts use of the Custom Values command in the Filter dialog box, which is used to access a custom values list saved with the Interactive Reporting document file or in a flat file.

  - **Custom SQL**—Enables the user to code a filter directly using SQL.

    Note: The Topic Priority dialog box is displayed only if you first select join in the data model.

    Note: Since most data models do not have the same set of topics, you cannot save changes to the topic priority as default user preferences. (For more information on default user preferences, see "Saving Data Model Options as User Preferences" on page 545.)

Data Model Options: Joins

Use the Joins tab to select join usage preferences.

- **Use All Joined Topics**—Use all joined (non-iconized) topics in the data model.

- **Use The Minimum Number Of Topics**—Use only of topics represented by items on the Request Line.

- **Use All Referenced Topics**—Use only of topics represented by items on the Request or Limit lines. Changing join usage usually changes the number of rows retrieved from the database. It also introduces the possibility that novice users may create improperly joined queries.

- **Use Defined Join Paths**—Use user predefined join path that groups the joins necessary to query from the data model. Click **Configure** to create a custom join path.

- Note that since most data models do not have the same predefined join paths, you cannot save the Use Defined Join Paths option as a default user preference. (For more information on default user preferences, see "Saving Data Model Options as User Preferences" on page 545.)

- **Use Automatic Join Path Generation**—Instructs Interactive Reporting to dynamically generate joins based on the context of user selections on the Request and Filter lines.

Data Model Options: Topic Priority

Use the Topic Priority tab to define the order that tables are included in the Interactive Reporting SQL statement. Defining a topic priority can significantly speed up large queries.

When defining topic priorities, remember that the centralized fact topic in your data model is the largest and receives the most use during a query. By prioritizing this topic first, followed by the remaining topics in descending order of magnitude, the database server can more efficiently use the internal join logic between tables.

➤ To set topic priorities in a data model:

1   Choose **Data Model**, then **Data Model Options**.

    The Data Model Option dialog boxes is displayed.

2   Select **Topic Priority**.

    Topics in the data model appear listed in the Tables list in the order they were placed in the Content pane.

3   Rank the topics in the desired order. Click the arrow to move selected topics up or down in the list.

4   Click **Auto-Order** to automatically detect the magnitude of each topic and rank them accordingly in descending order.

5   When the topics appear in the desired order, click **OK**.

    To save the current topic priority order as the default, select *Save as Default* before clicking *OK*.

Note: **Explorer Users:** The Topic Priority dialog box appears only if you select a join in the data model.

Note: Since most data models do not have the same set of topics, you cannot save changes to the topic priority as default user preferences. (For more information on default user preferences, see "Saving Data Model Options as User Preferences" on page 545.)

Data Model Options: Auditing

Use the Auditing tab to monitor user events within a managed query environment.

By attaching SQL statements to specificInteractive Reporting document file events, an advanced user can record how Interactive Reporting Studio, a database server, and network resources are being used. When triggered, the SQL statements update an audit log table, which the administrator can query independently to track and analyze usage data. For detailed information about auditing, see "Auditing with Interactive Reporting" on page 565.

Note: Although you can save the definitions of specific audit events as default user preferences, you cannot save the enabled/disabled state of the audit events as defaults. (For more information on default user preferences, see "Saving Data Model Options as User Preferences" on page 545.)

Automatically Processing Queries

Use the *auto-process* feature to have a standard query processed automatically when it is downloaded from the repository.

➤ To set Auto-Process:

1   Display a standard query Interactive Reporting document file open in the Content pane.

2   Select **Query**, then **Query Options**.

    The Query Properties dialog box is displayed.

3   Select the **Auto-Process** check box, and then click **OK**.

4   Select **File**, then **Save To Repository** to upload the Interactive Reporting document to the repository.

    The query automatically processes when a user opens the Interactive Reporting document file from the repository.

Promoting a Query to a Master Data Model

A query may be promoted to a *master data model*. This essentially separates the data model from the query, enables multiple queries to be based on a single master data model, and creates a new data model-only section in the Interactive Reporting document file. Master data models do not contain Request lines.

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. Each time a new query is inserted into a Interactive Reporting document file that contains a master data model, you are prompted to link the new query to the master data model. When a query is promoted to a master data model, it is added to the Section pane as a new section. Once you promote a query to a master data model, you cannot undo it.

➤ To promote a query to a master data model:

1   In the Query section, select or build a data model.

2   Select **DataModel**, then **Promote to Master Data Model**.

    Data models in Query sections that are linked to the master data model are locked and cannot be changed. They display with a gray background and the message: "Locked Data Model."

    Note: To keep the master data model locked for web users, deploy the Interactive Reporting document file in "Query and Analyze" mode and not in "Data Model" mode.

Synchronizing a Data Model

If data models are distributed to company personnel, it is important to keep them updated to reflect system changes. Data models provide visual understanding of the database; if they are corrupted, users can become lost and frustrated.

For example, consider the situation when a database administrator structurally alters a database table by adding columns, modifying data, or renaming a field. If the changes are not registered to data models, then Dashboard sections, metatopics, or intranet-distributed reports become obsolete.

Advanced users can ensure data model integrity using the Sync With Database command, a one-step integrity check and update. Sync With Database detects inconsistencies with the database, updates the data model, and provides an itemized list of the changes made. The list can use be used to update metatopics and report sections quickly and without interrupting workflow.

➤ To synchronize a data model:

1   Open the **data model** and log on to the database.

2   Select **DataModel** , then **Sync With Database**.

Interactive Reporting Studio compares original topics with their corresponding database tables. If the structure of the tables has changed, Interactive Reporting Studio modifies data model topics to reflect the changes. The Data Model Synchronization dialog box is displayed, describing changes to the database. Select the Show Detail Information check box for an itemized list.

Tip: Because metatopics are a separate logical layer constructed from original topics, they are not automatically updated. The Sync With Database feature removes any altered items from metatopics, but preserves the remaining structure so that repairs are minor. Sync With Database works transparently with most other customized attributes of a data model.

Data Model Menu Command Reference

The following table provides a quick reference to the commands available on the Data Model menu and lists any related shortcuts.

Table 179   Data Model Menu Commands

Command	Description	Keyboard Shortcut	Shortcut Menu
Table Catalog	Expands the Table catalog in the Catalog frame.	F9	
Data Model View	Enables you to select combined, original (database-derived), or metaviews of topics.		
Topic View	Enables you to select structure, detail, or icon views of topics.		*
Promote to Metatopic	Creates a metatopic from an existing topic.		*
Add Metatopic	Adds a metatopic to the data model.		
Add Metatopic Item	Enables you to add either a server or local metatopic item.		

Command	Description	Keyboard Shortcut	Shortcut Menu
Sync With Database	Detects inconsistencies with the database, updates the data model, and provides an itemized list of the changes.		
Promote To Master Data Model	Promotes the current query to a master data model.		
Data Model Options	Enables you to specify options for General, Limits, Joins, Topic Priority, and Auditing.		

20 Managing the Interactive Reporting Document Repository

About the Document Repository

The document repository provides an efficient way to manage and distribute data model objects for end-user querying. By storing standardized objects in a document repository located on the database server, advanced uses can provide version-controlled data models for entire workgroups to access as needed.

Note: The document repository described in this section is a place for a Interactive Reporting Studio user to store and maintain data models and standard reports.

Objects that can be stored in the document repository are:

- **Data model**—A basic data model that is a group of related topics designed as a starting point for building a query. A basic data model opens in the Content frame of the Query section in which a group of joined topics is displayed.

- **Standard query**—A data model with a query already assembled. After the query is downloaded, you simply process the query to retrieve data. Standard queries are ideal for users who use the same data on a regular basis; for example, to get inventory updates that fluctuate from day to day. A standard query opens in the Results section.

  If a standard query has the *auto-process* feature enabled, the query automatically runs when it is downloaded and populates the Results and report sections with data.

- **Standard query with reports**—A standard query that includes preformed reports which enable you to process the query and view the data using customized report sections. A formatted standard query with reports is displayed in the Pivot, Chart, Dashboard, or Report sections.

Administering a Document Repository

Use the Administer Repository dialog box as an access point to create and maintain repositories and the objects stored inside the repositories. You can use this dialog box to inventory the contents of all repositories on a database server, and update the descriptions of the stored contents.

Note: Repository administration is the province of the Interactive Reporting Studio advanced user. The data model contents of a document repository are available to end users, but only an advanced user can store and manage shared repository objects.

The following sections describe the tasks associated with administering a document repository:

- "Creating Repository Tables" on page 554
- "Confirming Repository Table Creation" on page 555
- "Managing Repository Inventory" on page 556
- "Managing Repository Groups" on page 557

Creating Repository Tables

A repository is a central place in which an aggregation of data is kept and maintained in an organized way. A *document repository* is a group of specialized database tables used to store different kinds of data models.

A document repository can be located on any database in the network environment, and can even store data models associated with any other database in the environment.

➤ To create repository tables:

1 Select **Tools**, then **Administer Repository**.

2 Choose **Select** to open the Select Connection dialog box and select theInteractive Reporting database connection file for the database on which you want to create repository table, or select the Interactive Reporting database connection file for the active Interactive Reporting document file.

  The Administer Repository dialog box is displayed.

3 Click **Create** to open the Create Repository Tables dialog box.

4 Change the default configuration.

- **Owner Name**—Enter the database and owner names (if applicable) under which you want to create the tables. If both database and owner are specified, separate them with a period (for example, Sales.GKL).

- **Grant Tables to Public**—Check Grant Tables to Public to grant general access to the repository tables at the database level.

You must grant access to the repository tables in order for users to download data models; otherwise, you will need to manually grant access to all authorized users using a database administration tool.

Do not grant tables to public if you need to maintain tight database security and upload privileges are only permitted for a small group of users.

- **Data Type Fields**—Change default data types for column fields to match data types of the database server. If the DBMS and middle ware support a large binary data type, use it for VarData columns. If not, use the largest character data type.

5   Click **Create All** to create the repository tables under the specified user.

The All Tables Created dialog box is displayed.

Note: If table creation fails, make sure the database logon ID of the server has been granted Table Create privileges.

6   Click **OK**, and then click **Close** to close the Create All dialog box.

Confirming Repository Table Creation

Repository tables are hidden in the Table Catalog by default. To confirm that the repository tables were created (or if you would prefer to display the tables), you can modify the connection preferences of the Interactive Reporting database connection file, and include the repository tables in the Table catalog.

➤ To include repository tables in the Table catalog:

1   Select **Tools**, then **Connection**, and then **Modify**.

The Database Connection Wizard is displayed.

2   Select the **Show Advanced Options** check box, and then click **Next**.

3   Enter a user name and password to connect to the data source, and then click **Next**.

4   Clear the Exclude Hyperion Repository Tables check box and click **Next**.

5   Click **Next** through the rest of the wizard dialog boxes, and then click **Finish**.

6   Save theInteractive Reporting database connection file.

7   Select **DataModel** , then **Table Catalog** or press F9 to view the Table catalog including the document repository tables:

The following document repository tables should be displayed:

- BRIOCAT2
- BRIOGRP2
- BRIOBRG2
- BRIOOBJ2

For detailed information on the document repository tables, see " Document Repository Table Definitions" on page 561.

Managing Repository Inventory

Use the Administer Repository dialog box to create and maintain document repositories and the objects stored inside the repositories. You can use this dialog box to inventory the contents of all repositories on a database server and update the descriptions of the stored contents.

➤ To update a repository object description:

1 Select **Tools**, then **Administer Repository**, and select the Interactive Reporting database connection file associated with the repository object with which you want to work.

   The Administer Repository dialog box is displayed.

2 Click the **Inventory** tab.

3 Select a model type from the Model Type drop-down box.

   The Model Type drop-down box shows the model type folders that contain the repository objects. Interactive Reporting Studio supports three types of repository objects: Data Model, Standard Query and Standard Query with Reports. When you select a model type, the description for that model type becomes active.

4 Edit the description in the Description frame of the BRIOCAT2 area.

   The BRIOCAT2 area shows the following catalog details for the selected model:

   ● **Unique Name**—Name of the object as it is displayed in the repository.

   ● **Creator**—Name of the person who created the object.

   ● **Created**—Date on which the object was saved to the repository defaults.

   ● **Version**—Version number of the object.

5 Click **Update**.

To modify the attributes of a document object itself, download the object, alter the Interactive Reporting document file and upload it to the repository. For more information, see "Modifying Repository Objects" on page 559.

➤ To delete a repository object:

1 Select **Tools**, then **Administer Repository**, and then select the Interactive Reporting database connection file associated with the repository object with which you want to work.

   The Administer Repository dialog box is displayed.

2 On the **Inventory** tab, select the model type to be deleted from the Model Type drop-down box.

3 Select a repository object from the Model List and click **Delete**.

   The object is deleted from the repository.

Managing Repository Groups

The repository group feature enables you to classify stored objects by their availability to distinct workgroups that you define. Users can download repository objects provided that they have access privileges in an authorized workgroup.

This feature complements the open repository by adding a security layer which enables you to consolidate objects into a single repository while selectively restricting access to certain objects as needed.

For example, you are the database administrator at a software firm. Ellen needs access to sales and marketing data models to complete a customer survey presentation. Gavin, a product manager, uses these and product management data models to complete his competitive analyses. Jason, a salesperson, needs access only to the standard query with reports for sales.

The solution is to create groups: Product Management, Marketing, and Sales, and give each group access to the objects that they need. Then assign users to appropriate groups: Ellen would have access to both sales and marketing, Jason to sales, and Gavin to all three.

➤ To set up a repository group:

1   Select **Tools**, then **Administer Repository**, and then select the Interactive Reporting database connection file associated with the repository group with which you want to work.

    The Administer Repository dialog box is displayed.

2   Select the **Groups Setup** tab to display the Groups Setup tab.

3   Enter the group name that you want to add the repository structure in the Groups field and click **Add**.

    Tip: If you enabled *Grant Tables To Public* when creating the repository, the default group, Public, is in the Groups list.

4   Select the group for which you want to associate a user name or names.

5   Enter the user name(s) in the Users field, and click **Add** to add the names to the group.

    Add multiple users by delimiting with commas in the edit field, for example: user1, user2, and user3.

6   All users with access to the repository, regardless of other grouping affiliations, have default access to Interactive Reporting document files placed in the Public group.

➤ To remove a user group or user, select the user name in the Users list and click **Remove**.

Working with Repository Objects

The following section discusses how to create and modify repository objects, and how to use the automatic distributed refresh (ADR), a sophisticated version control feature, to control document versions:

● "Uploading Interactive Reporting Documents to the Repository" on page 558

Uploading Interactive Reporting Documents to the Repository

After you have created a document repository, you can upload repository objects (data models, standard queries, and standard queries with reports) for version-controlled distribution to networked Interactive Reporting users.

Note: When you store objects in the document repository for user access, make the Interactive Reporting database connection file available to users as well.

➤ To upload an object to the repository:

1 With the repository object you want to upload open in the Content frame, select **File**, then **Save To Repository** and select theInteractive Reporting database connection file to associate with the object.

   If necessary, click **Select** to launch the Select Connection File dialog box, navigate to the Interactive Reporting database connection file, and click **OK**.

   The Save To Repository dialog box is displayed showing the Model tab.

2 In the Model Type area, select the type of object you are saving to the repository.

   Select between Data Model, Standard Query, and Standard Query with Reports.

3 In the Model Info area, enter information about the repository object.

   - **Unique Name**—Name of the object that you want to show for the object in the repository
   - **Creator**—Name of the person who created the object. This information is useful in tracing the document source for updates and so on
   - **Created**—Date on which the object was saved to the repository defaults
   - **Locked/Linked Object** (**Required For ADR**)—Toggles repository object locking. Previously, repository models were locked to maintain versions (see "Controlling Document Versions (Automatic Distributed Refresh" on page 560), and could not be modified by the end user. Unlocked data models can be downloaded as usual and the query modified. However, once saved outside the repository, the unlocked model loses its automatic version-control.
   - **Prompt For Sync On Download**—Prompts users with the request: "A newer version of the object exists in the repository, downloading the changes may overwrite changes you have made to the local file. Would you like to make a copy of the current document before proceeding?"

     If the user selects *Yes*, a copy of the locally saved object is made, Automatic Distributed Refresh is disabled for the copy, and the object is synchronized with the newer version of the object
   - **Description**—Enter a description of the repository object and what it can be used for. The maximum character length that you can add is 255 characters.

4 Select the **Groups** tab.

Groups associated with the owned repository are displayed in the Groups list. The PUBLIC group is included by default.

5 Use the arrow buttons to grant access to repository groups by adding them from the Available Groups list to the Selected Groups List.

- **Available Groups**—Available user groups from which access can be granted.

- **Selected Groups**—Groups added to the granted access list for the stored object.

> **Tip:** You must move the PUBLIC group to the Selected Groups list if you want to provide general, unrestricted access to the repository object.

6 Click **OK** to save the object to the repository.

7 Distribute theInteractive Reporting database connection file to end users as needed to access both the object source database, and if necessary, the document repository used to store the object.

Modifying Repository Objects

You can make modifications to document objects stored in the repository by downloading, modifying, and uploading them again. You can save the object under a new name, but if the object is not significantly altered, it is best to retain consistency by reloading the document under the same name. This ensures that linked Interactive Reporting document files are automatically updated.

> **Caution!** Modifications made to repository objects propagate throughout the user environment via Automatic Distributed Refresh (ADR), which track objects by unique ID and version number. Each time the object is uploaded to the repository, it is also assigned a new version number. For ADR to work properly, you must upload a modified repository object with the same name as the original.

➤ To modify a repository object:

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

The Select Connection dialog box is displayed.

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

2 Select the Interactive Reporting database connection file, 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.

4 Navigate through the repository tree and select the repository object that you want to use

The Open From Repository dialog box 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

5 Click **Open.**

The repository object is downloaded to the appropriate section.

6 Make the desired changes to the object, and then select **File**, then **Save To Repository.**

7 Select the correct Interactive Reporting database connection for the repository object, and enter the user name and password if prompted.

The Save To Repository dialog box is displayed.

8 Select the **Model** tab and verify the correct document type in the **Model Type** field.

If the Model type is grayed out, the object has not been modified and it cannot be saved to the repository at this time.

9 Add any object information in the Model Info area and then click **OK.**

You are asked if you want to enter a unique name for the object. Click *No* to replace the current object with the object you just modified. Click *Yes* to save the modified object under a different name. For Automatic Distributed Refresh to work properly, you must save a modified object with the original object name and model type, and save it in the same user-owned repository.

10 If you assigned another name to the object, you are prompted to associate the modified object with a group. Click **OK.**

The Group tab is displayed automatically so that you can associate the object with a group.

11 Use the arrow buttons to grant access to repository groups by adding them from the Available Groups list to the Selected Groups List.

12 Click **OK.**

Controlling Document Versions (Automatic Distributed Refresh

Automatic Distributed Refresh (ADR) is a sophisticated version control feature that transparently updatesInteractive Reporting when the data model or standard query is changed in the document repository. ADR operates completely in the background without any user interaction

ADR assumes that:

- Each object in the BRIOOBJ table has a unique ID number.

- Each object is assigned an iterative version number each time it is altered and uploaded.

Data model objects are typically downloaded from the document repository intoInteractive Reporting document files that are used to analyze data through pivots, charts, and other reports. When a user saves work to an Interactive Reporting document files on disk (either a local hard disk or *file server*), Interactive Reporting Studio stores both a link to the source object (which was downloaded from the document repository) and the connection information needed to reconnect to the repository.

Document Versions in Interactive Reporting Studio

Document Repository Table Definitions

When the Interactive Reporting document file is reopened, Interactive Reporting Studio reads the link information, connects to the repository, checks to see if the object exists, and checks if it has the same version number stored in the document. If the object in the repository has been modified, it will have a new version number, which indicates that Interactive Reporting Studio should update the old version saved in the Interactive Reporting document. For ADR to work properly, you must save a modified object with the original object name and model type, and save it in the same user-owned repository. Data models and standard queries (with or without reports) are synchronized using ADR.

The document repository tables are detailed in the following sections:

> **Note:** The following tables, which were created in Interactive Reporting version 6.6 and prior, are no longer used nor will they be referenced by any aspect of Interactive Reporting Studio: BRIOOCE2, BRIODMQ2, BRIOUSR2, BRIOSVR2, and BRIOUNIQ.

BRIOCAT2 Document Repository Table

This table lists the details of the BRIOCAT2 table, which records a description of the repository objects and local documents loaded in the document repository

Table 180   BRIOCAT2 Table

Column	Datatype	Description
UNIQUE_ID	NUM	Unique identifier for a stored Repository object.
OWNER	CHAR	Creator of the object.
APP_VERSION	CHAR	Version used to upload the object.
CREATE_DATE	DATE	Most recent date of upload for the object.

Column	Datatype	Description
ROW_SIZE	NUM	Number of rows occupied by the stored object in the BRIOOBJ2 table.
READY	CHAR	Indicates whether previous upload of the stored object was completed successfully.
FILE_NAME	CHAR	Descriptive name of the stored object.
FILE_TYPE	CHAR	File type of the stored object, such as data model, locked query, locked report, LAN-based, folder.
DESCRIPTION	CHAR	Description of the object.
VERSION	CHAR	Latest version number of the object, used for ADR.
TOTAL_SIZE	NUM	Total size of the stored object in bytes.

BRIOOBJ2 Document Repository Table

This table lists the details of the BRIOOBJ2 table, which stores the actual objects loaded in the document repository.

Table 181 BRIOOBJ2 Table

Column	Datatype	Description
UNIQUE_ID	NUM	Unique identifier for a stored repository object.
ROW_NUM	NUM	Sequence ID for segment of the object.
VAR_DATA	BLOB or LONG RAW	Data Model object in binary chunk format.

BRIOBRG2 Document Repository Table

This table lists the details of details of the BRIOBRG2 table, which stores the associations between registered documents and repository groups.

Table 182 BRIOBRG2 Table

Column	Datatype	Description
UNIQUE_ID	NUM	Unique identifier for a repository document.
GROUP_NAME	CHAR	Name of a repository group.

BRIOGRP2 Document Repository Table

This table lists the details of the BRIOGRP2 table, which maintains the list of repository groups and their associated users and privileges.

Table 183 BRIOGRP2 Table

Column	Datatype	Description
GROUP_NAME	CHAR	Name of a repository group.

Column	Datatype	Description
USER_NAME	CHAR	Name of a document repository user assigned to the group.

Document Versions in Interactive Reporting Web Client

Automatic Document Refresh (ADR) in Interactive Reporting Web Client enables the end-user to merge and synchronize locally saved Interactive Reporting documents (BQY) as soon as they connect to the repository with the latest version of the repository document. This feature is only for documents opened for the first time.

When a locally saved document is opened, a connection dialog for the Interactive Reporting database connection file (.oce) in the repository prompts the user to connect to the repository. The version information in the Interactive Reporting document is compared with the current version information for the document in the repository. If ADR has been enabled for the document, you are prompted to update to the latest version. The user can proceed with the refresh or not. After the document has been refreshed, the refreshed data cannot be undone. If the flag *ADR* is disabled for the document, the document refresh is not available and the locally saved document is opened as is.

The ADR synchronizing procedure is controlled at the system level and at the document level. Interactive Reporting Web Client ADR always refreshes the whole document. All the documents are published and stored in the repository with some ADR control flags enabled or disabled. Unlike in Interactive Reporting Studio, in Interactive Reporting Web Client there is no concept of Model Type.

ADR Control Flags

Control flags determine if an Interactive Reporting document is eligible for ADR. These flags include:

- **ADR Global Flag**—This flag controls the availability of the ADR feature. For a new installation of Interactive Reporting, this flag defaults to enabled. For an upgrade installation, this flag is disabled. You system administration can enable or disable this feature as needed.

- **ADR BQY Metadata**—This flag is enabled or disabled when an Interactive Reporting document is published to the repository. If the flag is enabled, then only this particular document is allowed for ADR. For simple ADR, this flag is always enabled. ADR for job output defaults to a disabled flag when an Interactive Reporting document is published by a job action. In this case, a user can enable this flag by modifying the properties of the Interactive Reporting document. This flag is always disabled for a job output collection.

ADR Behavior

The following table shows how ADR behaves with documents in different scenarios.

Table 184   Simple ADR Behavior

Interactive Reporting document section in Repository version	Interactive Reporting section in local Interactive Reporting document	Action in Merged Interactive document
Section does not exist.	Section does not exist.	No action
Section does not exist.	Section exists.	Add from local document.
Section exists.	Section does not exist.	Add from Repository document.
Section exists.	Section exists.	Write Repository version.

Auditing with Interactive Reporting

About Auditing

Auditing enables information to be collected about data models downloaded from the repository. You can use auditing features to track how long queries take to process, which tables and columns are used most often, and even record the full SQL statement that is sent to the database.

Audit information can help the database administrator monitor not only the effectiveness of each distributed data model, but also the weaknesses and stress points within a database. The results are useful for performing impact analysis to better plan changes to the database.

Auditing requires minimal additional setup and can be implemented entirely within the studio version of Interactive Reporting. The steps required for auditing data models are:

● Create a document repository with an inventory of distributed data models.

● Create a database table in which to log audit events.

● Use data model options to define events that you want to audit for each data model.

● Save the audited data models to the document repository.

● Use Interactive Reporting to query the audit table and to analyze the data it contains.

Special Considerations

● The Audit log may fill up. Monitor it regularly and delete any entries that are no longer used.

● Before uploading your audited data model to the document repository, log in as a user and test each auditing event to verify that your SQL statements are not generating any errors.

● Auditing is not supported for the Process Results To Database Table feature, nor for data models. However, scheduled Interactive Reporting documents containing linked data models are audited normally.

Creating an Audit Table

Before you enable auditing of data models, you need to identify the events that you want to track and create a database table to record the information.

Use an SQL editor to create an audit table. Since the query accesses only one database, the audit table needs to reside where the query is processed. Create columns that reflect the types of information that you want to record.

Table 185 provides a sample structure for the table named BQAUDIT. You can customize your audit table and columns to store information related to any events that you can define.

Table 185   Sample Structure for the BQAUDIT Table

Column	Data Source	Explanation/Example

EVENT_TYPE	Text	Events which occur within the context of a query session, such as:
		
		    'Logon'
		    'Logoff'
		    'Post Process'

USERNAME	SQL function	Database user information returned by a database SQL function, such as:
		
		    user (Oracle)
		    user_name (Sybase)
		    CURRENT_USER (Red Brick)

DAY_EXECUTED	SQL function	Date, time, and duration information returned by a database SQL function, such as:
		
		    sysdate (Oracle)
		    getdate (Sybase)
		    CURRENT_TIMESTAMP (Red Brick)

SQL_STMT	Interactive Reporting keyword	SQL statements generated by the user and captured from the Interactive Reporting SQL log, and returned by the keyword variable :QUERYSQL
DATAMODEL	Interactive Reporting keyword	Data models accessed by the user and returned by the keyword variable :REPOSITORYNAME
NUM_ROWS	Interactive Reporting keyword	Query information returned by the keyword variable :ROWSRETRIEVED

Defining Audit Events

After you create the audit table on the database, define the events that track each data model.

Note: To log audit data, provide Interactive Reporting users the database authority to execute each SQL statement you define for auditing events. For example, all users must have insert or update authority to the Audit table that you create.

➤ To define audit events:

1 Download an existing data model that you want to track from the document repository, or create a new data model in the Content frame using the Table catalog.

For more information about creating a new data model, see "Building a Data Model" on page 530.

2 In the Query section, select **DataModel** , then **Data Model Options**.

The Data Model Options dialog box is displayed.

3 Select the **Auditing** tab to display the Auditing tab.

The Auditing tab displays the events you can audit.

4 Click **Define** to define the way in which an event is audited.

The SQL For *Event* dialog box is displayed.

5 Enter one or more SQL statements to update the audit table when the event occurs, and click **OK**.

A check mark is displayed next to the event on the Auditing tab in the Data Model Options dialog box. You can use the check box to enable or disable the event definition without entering the SQL statement again. You can also click *Define* again at any time to modify the SQL statement.

6 Select **File**, then **Save to Repository** to save the audited data model to the document repository.

The SQL statement is sent to the database whenever a user triggers the event while using the data model.

Auditing Keyword Variables

Interactive Reporting Studio provides keyword variables (see Table 186) that can be used to help define audit events. The keywords can be inserted into audit event SQL statements to return specific data each time the event is triggered.

Tip: When entering an auditing keyword variable, always precede it with a colon (:) and enter all keyword text in uppercase. Other items in the SQL statement may also be case sensitive, depending on your database software.

Table 186   Auditing Keyword

Keyword Variable	Description
:ROWSRETRIEVED	Number of rows retrieved by the most recently executed query.
:REPOSITORYNAME	Name of the repository object in use (data model or standard query with reports).
:QUERYSQL	(Pre Process, Limit: Show Values, and Detail View only) Complete SQL text of the most recently executed query statement. **Tip:** Consider the maximum column length when using :QUERYSQL. You may want to use a substring function to limit the length of the SQL being logged. For example: SUBSTR(:QUERYSQL,200)

Keyword Variable	Description
:SILENT	Restricts display of the audit-generated SQL statement within the user's SQL Log. When the :SILENT keyword variable is included in the audit statement, the SQL Log output reads "Silent SQL sent to server..." instead of the SQL statement.
	This keyword variable provides a security feature when the triggered SQL statement is sensitive or should remain undetected.

Sample Audit Events

Table 187 provides examples of audit events. Most examples include ORACLE SQL database functions.

Table 187 Sample Audit Events

Audit Event	Description
Logon	Executed each time a successful logon occurs.
| | ```
insert into <owner>.bqaudit (username, day_executed, event_type) values (user, sysdate, 'Logon')
``` |
| | **Note:** The logon audit event fires for each action when used with the Data Access Service. Because of connection pooling, it is not possible, at the Data Model level, to determine when an actual logon event is required. |
| **Logoff** | Executed each time a successful logoff occurs. |
| | ```
insert into <owner>.bqaudit (username, day_executed, event_type) values (user, sysdate, 'Logoff')
``` |
| | **Note:** The logoff audit events fires for each action when used with the Data Access Service. A logoff event does not happen until the connection reaches the configured idle time. |
| **Pre Process** | Executed after Process is selected, but before the query is processed. It is useful to track the date and time of both Pre Process and Post Process in order to determine how long a query takes to process. |
| | ```
insert into <owner>.bqaudit (username, day_executed, event_type) values (user, sysdate, 'Pre Process')
``` |
| **Post Process** | Executed after the final row in the result set is retrieved at the user's workstation. It is useful to track the date and time of both Pre Process and Post Process in order to determine how long a query takes to process. |
| | ```
insert into <owner>.bqaudit (username, day_executed, event_type, num_rows, sql_stmt) values (user, sysdate, 'Post Process', :ROWSRETRIEVED, SUBSTR(:QUERYSQL, 1, 200))
``` |
| **Limit:Show Values** | Executed after selecting the Show Values button when setting a Limit. |
| | ```
insert into <owner>.bqaudit (username, day_executed, event_type, datamodel, sql_stmt) values (user, sysdate, 'Show Values', :REPOSITORYNAME, :QUERYSQL)
``` |
| **Detail View** | This statement is executed when a user toggles a topic to Detail View and a sampling of data from the database is loaded. Remember that values are only loaded when you first toggle to Detail View, or when Cause Reload is selected in the Topic Properties dialog box. |

Audit Event	Description
New Data Model	This statement is executed when the Data Model is downloaded from the document repository into a Interactive Reporting document. `insert into <owner>.bqaudit (username, day)_executed, event_type, datamodel) values (user, sysdate, 'New Data Model', :REPOSITORYNAME)`
Data Model Refresh	This statement is executed after a Data Model is refreshed through ADR. `insert into <owner>.bqaudit (username, day_executed, event_type, datamodel) values (user, sysdate, 'Data Model Refresh', :REPOSITORYNAME)`

22 IBM Information Catalog and Interactive Reporting Studio

About the IBM Information Catalog

IBM's Visual Warehouse (VW) is a family of products that design, load, manage, and retrieve information from data warehouses. Interactive Reporting Studio is a component for the IBM VW solution, and is resold by IBM as part of VW bundles. To further extend the capabilities of the solution, register, administer, and distribute Interactive Reporting documents (BQY file extension). in the VW Information Catalog. The Information Catalog is a repository of document information with pointers to the physical objects.

Another feature of the Catalog is that it enables you to categorize content stored in documents by specific subject area. A full-search engine in the repository enables you to search for information stored in the documents. For example, you could search on all documents associated with "sales." In this case, the search results could yield Word files, Excel files, and Interactive Reporting documents. When you find a document that you want to work with, the IBM Information Catalog launches the appropriate application and opens the document.

Registering Documents to the IBM Information Catalog

This section explains how to register documents to the Catalog. It includes the following sections:

- "Defining Properties" on page 572
- "Selecting Subject Areas" on page 572

Visual Warehouse must already be installed before you can register or administer this feature. Also, the client document object types must already exist before completing the following steps. For more information see "Creating Object Type Properties" on page 573.

➤ To register an Interactive Reporting documents:

1 Display the repository object that you want to upload open in the Content pan.

2 Select **File**, then **Register to IBM Information Catalog**.

The Save File dialog box is displayed.

3   Type the name of Interactive Reporting document in the **File Name** field.

4   In the **Save As Type** field, leave the default `.bqy` file type and click **Save**.

The Connect To Information Catalog Repository dialog box is displayed.

5   Type your user identification in the **User** field.

6   Type your password in the **Password** field.

7   Type the ODBC data source name in the **Database Alias** field if it is different than the default database alias value.

The Register To Information Catalog dialog box is displayed, showing the Properties and Subject Area tabs. Use these corresponding pages to describe the properties and subject matter of the Interactive Reporting document.

8   Click the **Properties** tab to display the Properties tab.

9   In the **Available Properties** list, select a property of the Interactive Reporting document to which you want to add a value.

10 In the **Enter Value for Selected Property** edit box, type a value for the property.

11 Repeat Step 8 through Step 9 for all properties.

12 Click the **Subject Areas** tab.

13 In the **Specify The Subject Area** list, use the plus (+) and minus (–) signs to navigate through the Subject area structure (Grouping Category) and select the subject area folder to which you want to add the Interactive Reporting document.

The Subject Area displays a tree view of eligible subject area folders in which you can add the Interactive Reporting document.

14 Click **Add** to add the Interactive Reporting document or instance to the Subject Area specified in Step 12.

15 Click **OK**.

Defining Properties

You can define the values of selected properties for a document when registering to the catalog. Use the Properties tab to show and edit properties, data types, and lengths:

- **Available Properties**—Displays a list of available properties that you can specify.

- **Enter Value**—Edit any available value by typing the information in this edit box. For a description of eligible values for the properties, see the Description field.

Selecting Subject Areas

Use the Subject Area tab to display and select a subject area for the document that you are registering. By including the document in a Subject Area folder, you can later search for the document by topic.

- **Specify The Subject Area**—Displays a tree view of eligible subject area folders in which you can add the document. Use the plus (+) and minus (−) signs to navigate through the folders. To add a document to folder, select the subject area folder and click **Add**.

- **Subject Areas Containing**—Displays the subject area folder to which the document has been added.

Administering the IBM Information Catalog

This section explains how to administer the IBM information catalog, including:

Creating Object Type Properties

Use the Setup Objects Types under Administer IBM Information Catalog to create an object type and specify its properties. An *object type* shows a category of business information, for example, a Interactive Reporting document or an image. An *object type property* describes an attribute of the object type, for example, its name or data type.

Once an object type has been created, you cannot modify its existing properties or add new properties. You can, however, delete the entire object type, but not the individual properties of a selected object type.

➤ To set up the BQY object type and properties:

1   Choose **File**, then **Administer IBM Information Catalog**.

   The Connect To Information Catalog Repository dialog box is displayed.

2   Type your user identification in the **User** field.

3   Type your password in the **Password** field.

4   Type the ODBC data source name in the **Database Alias** field if it is different than the default database alias value.

5   Click **OK**.

   The Administer Information Catalog dialog box is displayed.

6   Click the **Setup Object Types** tab.

7   In the Object Type drop-down box select Interactive Reporting document.

8   In the **Name** field, type the name of the property that you want to associate with the object type.

9   In the **Short Name** field, type an abbreviated version of the property name.

10 In the Datatype list, select the data type classification of the property (for example, character-based) from the drop-down list box.

11 In the **Length** field, type the maximum length character of the property.

12 To require that the property be completed when a user registers a document, click the **Entry Required** check box.

13 To add the object type property to the Properties for Object Type list box, click **Set**.

14 Repeat Step 8 through Step 12 for each property that you want to associate with the selected object type.

15 To create the object type, click **Create Object Type**.

Deleting Object Types and Properties

You can delete the entire object type, but not the individual properties of a selected object type once an object type has been created.

➤ To delete an BQY object type and properties:

1 Choose **File** , then **Administer IBM Information Catalog**.

The Connect To Information Catalog Repository dialog box is displayed.

2 Type your user identification in the **User** field.

3 Type your password in the **Password** field.

4 Type the ODBC data source name in the **Database Alias** field if it is different than the default database alias value.

5 Click **OK**.

The Administer Information Catalog dialog box is displayed.

6 Click the **Setup Object Types** tab.

7 In the Object Type drop-down list box, select Interactive Reporting document.

8 Click **Delete Object Type**.

Administering Documents

Use the Administer Documents tab to search for a specific document based on an object type, property, and other selected criteria (see Table 188). After the document has been located, you can either delete or edit the associated properties.

Table 188   Object Type Search Criteria

Field	Description
Object Type	Interactive Reporting document object type.

Field	Description
Select Property	Property by which you want to search on the document from the pull-down list. Complete the search condition by selecting a value in the Search Criterion field below. For example, if you specify a Name property, type the name of the document in the Search Criterion field below.
Search Criterion	Use this field in conjunction with the Select Property field above. Once you have selected a property complete the search conditions by specifying the value of the property. For example, if you selected the Order Type property, you might type Interactive Reporting document in this field.
Case-sensitive Search	If you want the search engine to distinguish between uppercase and lowercase letters when determining which documents to retrieve, click this field.
Wildcard Search	A wildcard is a special symbol that represents one or more characters and expands the range of your searching capabilities. You can use the % wildcard symbol to match any value of zero or more characters. For example, to documents whose properties contains "1997 sales", type: `1997 Sales %` in the Search Criterion field.
Search	Retrieves the search results.
Clear Search	Clears the results of the current search.
Search Results	Results of the search.
Delete	Deletes a selected document from the repository.
Edit	Enables you to edit the value properties of a document through the Properties tab of the Register To IBM Information Catalog option.

Setting Up Object Types

Use the Set Up Object Types tab to set up object types and their properties (see Table 189) An *object type* shows a category of business information, for example, a document or an image. An *object type property* describes an attribute of the object type, for example, its name or data type.

Note: You can create and delete only the Interactive Reporting document object types and properties through the Interactive Reporting Setup Object Types features. For more information, see "Creating Object Type Properties" on page 573.

Table 189 Object Types and Properties

Field	Description
Object Type	Interactive Reporting document object types.
Name	Name of the property that you want to associate with the object type.
Short Name	Short name of the property that you want to associate with the object type.

Field	Description
Datatype	Data type of the property.
Length	Length of the property.
Entry Required	Requires a user to select a property when registering a document to the DataGuide repository.
Set	Adds a new object type property to the Properties for Object Type list. If an object type has already been created, this button is unavailable.
Remove	Removes a new object type property from the Properties For Object Type list. If an object type has already been created, this button is unavailable. Once an object type has been created, you cannot remove its properties; the entire object type must be deleted.
Properties For Object Type	Properties defined for the object type. To show the entire definition for a property, click a property in the list.
Create Object Type	Creates an Interactive Reporting document (BQY) object type. Once an object type has been created, you cannot modify its existing properties or add new properties.
Delete Object Type	Deletes an Interactive Reporting document (BQY) object type. You cannot delete the individual properties of a selected object type.
Clear	Clears the definition fields of a property.

23

Row-Level Security in Interactive Reporting Documents

In This Chapter

About Row-Level Security

Properly implemented, row-level security gives individuals in an organization access to only the information they need to make informed decisions. For example, managers need payroll information on their direct reports. Managers do not need to know payroll information for other departments within the organization. Row-level security allows this level of granularity.

The Row-Level Security Paradigm

Most database administrators understand the concept of row-level security. Returning to the payroll data example, all detailed compensation data on the employees of an organization is stored in the same table(s) within the database. Typically, some column within this table can be used as a limit, either directly or by a join to another table with its own set of limits, to restrict access to the data within the table based on the identification of the user accessing the data. Following the payroll example, an employee ID often identifies the sensitive compensation data. A join to a separate employee information table, which contains non-compensation related information such as home address and title, would include a department number. A manager would be limited to details on the employees for her/his particular department.

Row-level security, implemented at the database level, is often done by means of a view. To an application, accessing a view is no different than accessing a table. However, the view is instantiated based on the appropriate limits. Coupled with the GRANT and corresponding REVOKE data definition statements available with the prevalent Relational Database Management Systems (RDBMS), the base tables can be made inaccessible to most users, and the views on that data, filtered based on user identification, made accessible instead. Multiple views may sometimes be required to fully implement a security scheme, depending on how the tables are defined and how the information contained therein must be shared. For instance, a different view for managers versus those in human resources might be required.

Column level security, a companion concept to row-level security, can be similarly enforced. Views can easily hide a piece of information. For example, in the payroll example, the salary column can be left out of the view, but other types of information about an employee can still be accessible to those who need it. This type of security can impart a special problem for standardized reports, available throughout the organization but for audiences with different access permissions. The reporting software might have difficulty dealing with the missing information, consequently requiring different implementations of these otherwise similar reports.

Reporting and Analysis and Row-Level Security

The Reporting and Analysis approach to data security is server based. Both the Interactive Reporting Web Client and EPM Workspace are designed to fully implement a secure data access platform. The non-server based clients do not participate in this security mechanism. Users of Interactive Reporting Studio need access beyond that of most users to effectively create the dashboards, and analytic reports required by the majority of the data consumers. In addition, the security information can be placed in a centralized location for the servers (the repository). For the desktop clients, it would in some cases need to be dispersed to multiple databases and maintained separately.

To effectively control access, the servers key off the user's identification when connecting to it. This is the user's logon name, used to establish a session with the Reporting and Analysis services.

Beyond this user name, the servers make no assumptions about the user's place within the organization. A security system can be built entirely independent of any existing grouping of users. New groupings can be defined in lieu of existing ones. This is especially important where groups were not defined with data security as a primary goal. Row-level security can also take full advantage of the existing structures where data security was built into the user and group structure. In many cases, row-level security will work within existing database role definitions and third-party software security systems.

Column Level Security

Interactive Reporting servers can easily handle the challenge of column level security. Such hidden information is replaced with an administrator-supplied value (possibly NULL, zero, blanks, or a similar placeholder value), and thus it will not cause existing reports to fail when encountering this type of a security constraint.

Performance Issues

The system is designed to not impose any significant performance penalty. The security information is collected at the time the user opens an Interactive Reporting document from the server's repository, and only then if the server knows the security controls are enabled. When a user opens a locally saved Interactive Reporting document from a previous session with the Reporting and Analysis services, the security information is recollected when reconnecting to the server in case it has changed.

Publishing in a Secure Environment

A powerful feature of Interactive Reporting is the ability to take data "on the road." Once data has been extracted from the database, which is where the row-level security restrictions are enforced, that data can be saved with the Interactive Reporting document for offline analysis and reporting. Users who publish should be aware of the implications of their audience when publishing data and reports.

If the publication of the data is difficult to control in the current configuration of users and groups known to the server, consider the following options:

● Publish without the detailed results of the queries, leaving only the summary charts and Pivots for the "general" audience. If they need to drill into the summary data, they will need to rerun the queries, at which time their particular security restrictions will be applied. (Even some charts and Pivots can reveal too much, so there is still a need for prudence when publishing these Interactive Reporting documents).

● Create theInteractive Reporting documents with `OnStartup` scripts to reprocess queries as the Interactive Reporting document is opened. This option provides the user only the data to which they are entitled.

All users should take similar precautions when sharing information generated from Interactive Reporting. This includes exchanging the Interactive Reporting documents (`BQY` extensions) themselves by e-mail or shared network directories, exporting the data as HTML files and publishing them to a web site, posting the data on FTP servers as the result of a job action, and creating PDF files from the reports.

Securing the Security Information

The row-level security feature is implemented by means of database tables. The servers read this data and never update it. it is recommended that these tables actually be defined somewhere other than the repository schema, and that read-only access be granted to only the select few that should be able to update the security information.

As additional protection, the actual tables can be hidden via a view, and a WHERE clause can be added to each view definition so that only the server's user identification, by which it connects to the database to read the row-level security tables, can read the content, if the database supports it. Table 190 shows examples of Where clauses if the repository connection is made as user *brioserver*.

Table 190   Repository Connection Made as *brioserver*

Database	Sample Where Clause on CREATE VIEW
DB2	WHERE USER = 'BRIOSERVER'
Oracle	WHERE USER = 'BRIOSERVER'
SQL Server	WHERE USER = 'brioserver'

Note: Be aware of case sensitivity with the user name and allow that, for SQL Server, the user might be 'dbo'.

Each view has the same name as its underlying table, and all available columns from that table would be selected.

Row-Level Security Tables

Three tables implement the row-level security features for Interactive Reporting Web Client and EPM Workspace. Because the tables can be populated in a manner appropriate to the site's requirements, no predefined user interface to maintain the tables is provided as part of the Interactive Reporting product. However, the `row_level_security.bqy` can be modified to suit an ad hoc implementation of the row-level security feature, and as a tutorial or test tool when setting up a production system. The only requirement is that the basic set of column definitions be retained. The sample Interactive Reporting document can be used in all cases as a reporting tool for the row-level security data as the servers see it.

Implementing a secure data access environment using row-level security requires an understanding of SQL. First, knowing how the database relationships are defined is critical. Second, specifying the restrictions is directly translated into the SQL ultimately processed at the database.

Creating the Row-Level Security Tables

When the Interactive Reporting components are installed using the *custom* option, the installer prompts for whether or not to create the row-level security tables and in what database.

If the Interactive Reporting components are installed using the *express* option, or it is elected not to create the row-level security tables during a custom install, then the tables must be created manually. The Interactive Reporting document mentioned above can create the tables, the necessary SQL DDL can be created based on the table definitions found in this documentation, or the script that creates them during the install can be run. To locate these scripts, look on the install CD under the DATA directory, and then under the appropriate database brand or vendor. The script of interest is named `CreateRLS.sql`.

When creating the tables post-install, use the web-based Administration module to tell the system where the tables are located. The information required includes the data source name, the database type, the API used to access the database, and the database credentials needed to access the row-level security tables.

There is also a setting to enable or disable the row-level security feature. This setting is intended to enhance performance in systems where the feature is not needed. When disabled, no attempt is made to access these tables. The feature should always be enabled if data security is to be enforced.

The BRIOSECG Table

The BRIOSECG table defines the users and groups that are subject to row-level security restrictions. There are two columns, BUSER and BGROUP, both of varying character length (VARCHAR(n)). The maximum length is not fixed by the server; set it to a practical value.

A user name is defined as the server authentication name (`ODSUsername` is the property of the ActiveDocument object in the Interactive Reporting Object Model). For jobs, it is the user who scheduled the job.

Group names are arbitrary. The data security administrator is free to define these as required. When both columns of a row are populated with non-null values, the user name defined in the BUSER column is a member of the group name defined in BGROUP.

As maintained by the sample Interactive Reporting document `row_level_security.bqy`, when a user is added, a row is added to the table with a NULL value in the BGROUP column. When a group is added, a NULL value is stored in the BUSER column. This is a device used by the Interactive Reporting documents to maintain the table and is recommended practice, but it is not a requirement for correct operation of row-level security.

This table is theoretically optional. Without it, however, all users exist as single individuals; they cannot be grouped to apply a single set of restrictions to all members. For example, Vidhya and Chi are members of the PAYROLL group. If this relationship is not defined in BRIOSECG, then any restrictions that apply to Vidhya that should also apply to Chi have to be defined twice. By defining the PAYROLL group and its members, Vidhya and Chi, the restrictions can be defined only once and applied to PAYROLL group.

A group name cannot be used in BUSER; that is, groups cannot be members of other groups. Users, of course, can be members of multiple groups, and this can effectively set up a group/subgroup hierarchy. For example, a PAYROLL group might contain users Sally, Michael, Kathy, David, Bill, Paul, and Dan. Sally, Dan, and Michael are managers, and so they can be made members of a PAYROLL MANAGER group. Certain restrictions on the PAYROLL group can be overridden by the PAYROLL MANAGER group, and Dan, to whom Sally and Michael report, can have specific overrides to those restrictions placed explicitly on the PAYROLL MANAGER group.

Where the database supports it, and if the user's authentication name in Reporting and Analysis corresponds, this table can be a view created from the roles this user has in the database. For example, in Oracle:

```
CREATE VIEW BRIOSECG (BGROUP, BUSER) AS
   SELECT GRANTED_ROLE, GRANTEE FROM DBA_ROLE_PRIVS
```

DBA_ROLE_PRIVS is a restricted table. Since the server reads the view using a configured database logon, it would not be appropriate to use USER_ROLE_PRIVS instead of DBA_ROLE_PRIVS, because that user view will reflect only the server's roles, not the user on whose behalf the server is operating. Again, this is an Oracle example; other RDBMS may or may not provide a similar mechanism. In some cases, depending on the database, a stored procedure could collect the role information for the users and populate a BRIOSECG table if a simple SELECT is inadequate to collect the information. This would require some means to invoke the procedure each time role definitions were changed.

When using the database's catalog or some other means to populate BRIOSECG, the sample Interactive Reporting document, `row_level_security.bqy`, cannot be used to maintain user and group information.

A special group, PUBLIC, exists. It does not need to be explicitly defined in BRIOSECG. All users are members of the PUBLIC group. Any data access restriction defined against the PUBLIC group applies to every user unless explicitly overridden, as described later.

All users can be made part of a group at once by inserting a row where BUSER is 'PUBLIC' and BGROUP is that group name. While this may seem redundant, given the existence of the PUBLIC group, it offers some benefits:

- It allows the database catalog technique described above to work. For example, in Oracle, a role can be granted to PUBLIC.

- It allows restrictions for a group other than PUBLIC to quickly be applied to or removed from everyone in an instant.

- It provides more flexibility when using override specifications as described later.

> **Note:** Restrictions are never applied against a user named PUBLIC, but only the group PUBLIC. For this reason, do not use PUBLIC as a user name. Similarly, to avoid problems, do not name a group the same as a user name.

The BRIOSECP Table

The BRIOSECP ('P' for parameter) has one column, named BCONJUNC with a data type of CHAR(3). Its value is either the word AND or the word OR, and the Interactive Reporting document file that *administers* the RLS creates and populates the table.

The BRIOSECR Table

The BRIOSECR table is the heart of the row-level security feature. It defines the specific restrictions to be applied to users and the groups (including PUBLIC) to which they belong. These restrictions take the form of join operations (a user cannot access a column in the employee salary table unless it is joined to the employee table), and limits (WHERE clause expressions) to be applied to either the source table (SALARY) or table(s) (EMPLOYEE) to which it is joined. Table 191 lists the columns in the BRIOSECR table.

As suggested earlier, existing security definitions can sometimes be translated into the format described here. A view, stored procedure, or programmatic mechanism can be used to translate and/or populate the information needed in BRIOSECR by the Reporting and Analysis servers. When these methods are used, the servers require the column names and data types defined above. And again, do not attempt to use the sample Interactive Reporting document, `row_level_security.bqy`, to manage this information.

If a join table is specified and it does not already exist in the data model the user accesses, it will still be added to the final SQL generated to ensure the security restrictions are enforced. This process is iterative. When a table is added and the present user, either directly or by group membership, has restricted access to that added table, those restrictions will also be applied,

which may mean additional tables will be added, and those restrictions will also be checked, and so on. Circular references will result in an error if they are defined.

Table 191   Columns in the BRIOSECR Table

Column Name	Column Type	Functional Use
UNIQUE_ID	INT	This column contains an arbitrary numeric value. It should be unique, and it is useful for maintaining the table by whatever means the customer chooses. The servers do not rely upon this column, and the servers never access this column. To that extent, it is an optional column but recommended. (It is required when using the sample Interactive Reporting document, `row_level_security.bqy`.) When the RDBMS supports it, a unique constraint or unique index should be applied to the table on this column.
USER_GRP	VARCHAR	The name of the user or the name of a group to which a user belongs. If PUBLIC, the restrictions are applied to all users.
SRCDB	VARCHAR, can be null	Used to identify a topic in the Data Model. (In Interactive Reporting, a topic typically corresponds to a table in the database, but it could be a view in the database.) If the physical name property of the topic is of the form name1.name2.name3, this represents name1. Most often, this represents the database in which the topic exists. This field is optional unless required by the connection in use. The most likely circumstance in which to encounter this requirement will be with Sybase or Microsoft SQL Servers where theInteractive Reporting database connection file (the connection definition file) is set for access to multiple databases.
SRCOWNER	VARCHAR, can be null	Used to identify the owner/schema of the topic in the Data Model. This would be name2 in the three-part naming scheme shown above. If the topic property, physical name contains an owner, then it must be used here as well.
SRCTBL	VARCHAR	Used to identify the table/relation identified by the topic in the Data Model. This is name3 in the three-part naming scheme.
SRCCOL	VARCHAR	Used to identify a column in SRCTBL. This is a topic item in Data Model terminology, and is an item that might appear on the Request line in a query built from the Data Model. In the context of the security implementation, the item named here is the object of the restrictions being defined by this row of the security table BRIOSECR. If this column contains an *, all columns in SRCTBL are restricted.
JOINDB	VARCHAR, can be null	If present, defines the database name qualifier of a table/relation that must be joined to SRCTBL.
JOINOWNR	VARCHAR, can be null	If present, defines the schema/owner name qualifier of a table/relation that must be joined to SRCTBL.
JOINTBL	VARCHAR, can be null	If present, names the table/relation that must be joined to SRCTBL.
JOINCOLS	VARCHAR, can be null	If present, names the column name from SRCTBL to be joined to a column from JOINTBL.
JOINCOLJ	VARCHAR, can be null	If present, names the column name in JOINTBL that will be joined (always an equal join) to the column named in JOINCOLS.
CONSTRTT	CHAR(1), can be null	If present, identifies a table/relation to be used for applying a constraint (limit). This is a coded value. If the value in this column is S, the column to be limited is in SRCTBL. If J, a column in JOINTBL is to be limited. If the value in this column is O, it indicates that for the current user/group, the restriction on the source column for the group/user named in column OVRRIDEG is lifted, rendering it ineffective. If this value is NULL, then no additional restriction is defined. If the JOIN* columns are also all NULL, the column is not accessible at all to the user/group. This implements column level security. See the functional use description of CONSTRTV for more information on column level security.

Column Name	Column Type	Functional Use
CONSTRTC	VARCHAR, can be null	The column in the table/relation identified by CONSTRTT to which a limit is applied.
CONSTRTO	VARCHAR, can be null	The constraint operator, such as =, !=(not equal), etc. BETWEEN and IN are valid operators. Basically, any valid operator for the database can be supplied.
CONSTRTV	VARCHAR, can be null	The value(s) to be used as a limit. The value(s) properly form a condition that together with the content of CONSTRTC and CONSTRTO columns create valid SQL syntax for a "condition" in a WHERE clause. Subquery expressions, therefore, are allowed. Literal values should be enclosed in single quotes or whatever delimiter is needed by the database for the type of literal being defined. If the operator is BETWEEN, the AND keyword would separate values. If :USER is used in the value, then the user name is the limit value. If :GROUP is used, all groups of which the user is a member are used as the limiting values. Both :USER and :GROUP can be specified, separated by commas. The public group must be named explicitly; it is not supplied by reference to :GROUP.

When applying column level security, CONSTRTV provides the SQL expression that will effectively replace the column on the Request line. For example, the value zero (0) might appear to replace a numeric value that is used in theInteractive Reporting documents but should not be accessible by the specified user/group. While any valid SQL expression that can be used in a SELECT list is permitted, pick a value that is acceptable for the likely use. For example, the word NULL is permitted, but note that in some cases, it might not be the appropriate choice, as it could also end up in a GROUP BY clause. |
| OVRRIDEG | VARCHAR, can be null | The name of a group or user. Used when CONSTRTT is set to O. If the group named in OVRRIDEG has a restriction on the source element, then this restriction is effectively ignored for the user/group named in USER_GRP. SRCDB, SRCOWNER, SRCTBL, and SRCCOL as a collection must be equal between the row specifying the override and the row specifying the conditions to be overridden. (See examples.) |
| CONSTRNL | CHAR(1) | The value in this column determines whether null values are included in the restriction. When this column contains a 'Y,' SQL includes null values. If this column is blank, null values are excluded.

The following example shows the SQL generated when the CONSTRNL field is "Y":

The following syntax in SQL is shown as:

`OR TABLE.COLUMN IS NULL`

For example this sample shows how to include nulls on the "Quality Assurance" column:

`SELECT AL1.ROW_NUM, AL1.INT_COL, AL1.FLOAT_COL, AL1.NUMBER_COL, AL2.CHAR_COL, AL1.VARCHAR_COL, AL1.DATE_COL, AL1.TS_COL, AL2.TIME_COLFROM QS.QS_BRIO AL2 WHERE ((AL1.VARCHAR_COL = 'Quality Assurance' OR AL2.VARCHAR_COL IS NULL)`

Note in the above the instruction: AL2 WHERE ((AL1.VARCHAR_COL = 'Quality Assurance' OR AL2. VARCHAR_COL IS NULL). |

OR Logic Between Groups

Each permission/restriction applied to the same group or user is separated by AND logic, but for users who belong to multiple groups, these sets of permissions/restrictions can be separated optionally by OR conditions or by AND conditions. This option makes the logic disjunctive or conjunctive respectively. For example, a user who is in group 'A' and is allowed to see sales data in the "truck" category for the Eastern Region, and who also belongs in group 'B' and is allowed to see sales data in the 'minivan' category for the Eastern Region, can see both 'truck' and

'minivan' data from one query using OR Logic. If conjunctive logic were used, that user would see no data, since the category could not be 'truck' and 'minivan' simultaneously.

Row-Level Security Examples

The examples are based on the sample Access database provided by option when installing Interactive Reporting on a Windows platform, using the Interactive Reporting database connection file `Sample.oce`.

For these examples, the users BRIO and VIEW&PROCESS require access to data that is denied to the rest of the users. These two users both belong to the group AMERICAS, which corresponds to a region of the same name. However, the user BRIO is a corporate officer who should be able to see all data.

Only one piece of data will be accessed in the course of these examples: the amount sales column from the sales fact table. The examples are more far-reaching than this might seem.

Screenshots for these examples come from theInteractive Reporting documents to which processing restrictions are applied, and from the sample Interactive Reporting document, `row_level_security.bqy`, mentioned earlier. For the screen shots from the sample Interactive Reporting document, the columns in the BRIOSECR table follow in a top down, left to right manner for the most part with the fields on the screen. Deviations from this will be noted where possible. In particular, though, note that the UNIQUE_ID column is not shown in this sequence of fields, appropriate to its optional role to the functionality of the software, although it is used behind the scenes by the sample Interactive Reporting document.

Figure 6 shows the layout of the data in the database, to illustrate the possible joins as intended by the database designer.

Figure 6   Database Layout Showing Possible Joins

Figure 7 shows the data model in the published Interactive Reporting documents.

Row-Level Security Examples   585

Figure 7   Data Model in the Published Interactive Reporting Document

Defining Users and Groups

Based on the above description of the users and groups involved in this example, insert a minimum of two rows into the BRIOSECG table, as shown in Table 192.

Table 192   Rows to Insert in the BRIOSECG Table

BGROUP	BUSER
AMERICAS	BRIO
AMERICAS	VIEW&PROCESS

In the sample Interactive Reporting document for maintaining row-level security information, once the information has been added, it would look something like Figure 8 when the AMERICAS group is selected.

Figure 8   Sample Interactive Reporting Document with AMERICAS Group Selected

Dealing with "The Rest of the Users"

The requirement here was that users who were not part of the AMERICAS group should have no access to this information. There are several ways to do this, and in part, what the best way is depends on who the "rest of the users" are.

If they are extranet users, this probably means no access; users outside of the corporate network should not get sales data, even summary data, as this might be considered proprietary and certainly not for any potential competitors. Using the PUBLIC group, restrict the entire SALES_FACT table from accessing this information by using the asterisk to reflect all columns (see Figure 9).

Figure 9   Restriction on SALES_FACT Table

(This is an example of column level security. All values from this table, if they appear on the Request line, will be substituted with NULL.)

Where there are no extranet concerns, it might be appropriate for all the employees to know how their company is doing overall, such a blanket restriction is not recommended. Instead, restrict the use of the STORE_ID column, the only means by which the sales information can be tied back to any particular store, country, region, etc. This will look identical to the case above except that STORE_ID is specified instead of an asterisk for the Source Column Name.

Overriding Constraints

Obviously, members of the AMERICAS group are also members of PUBLIC. So, regardless of the way the PUBLIC group was restricted, those restrictions are not to be applied to the AMERICAS group for the sales information. That group might be restricted in different ways, or not at all, and the same mechanism ensures that happens while PUBLIC restrictions are in place. Figure 10 shows this when using the sample Interactive Reporting document, `row _level_security.bqy`.

Figure 10   Overriding Constraint on User/Group

This only overrides PUBLIC constraints for this particular column. Restrictions on PUBLIC against other columns are still enforced against members of the AMERICAS group as well. If the restriction is on all columns of a table, designated by an asterisk, the override must also be specified with an asterisk and then specific column constraints reapplied to groups as needed.

Cascading Restrictions

In order to give the members of the AMERICAS group access to sales information only for the appropriate region, the query includes references to columns in other tables which are not necessarily part of the existing data model. The row-level security will function the same whether or not the tables already existed in the data model.

As seen in the table relationships pictured above, the region information is bridged to the sales information by the store table. To implement a constraint that makes only sales information available for a particular region requires two entries in the BRIOSECR table, one to join sales to

stores, and one to join stores to regions. This latter case also requires a limit value for the region name. (A limit on REGION_ID could also accomplish the same goal, but is not as readable, especially in an example. See the discussion to follow about subqueries for another perspective on limits on ID type columns.)

The first restriction required for this example is on the STORE_ID column. In order to use that column, a join must be made back to the STORES table. Figure 11 shows how this join would be specified.

Figure 11   Join to STORES Table

Now, the join to the Regions table is added, with the appropriate constraining value, as shown in Figure 12.

Figure 12   Join to Regions Table

The only remaining part of the example is letting user BRIO, also a member of the AMERICAS group, see the data in an unrestricted way. Handling this case is left as an exercise for the reader.

Other Important Facts

This section contains miscellaneous information about the implementation of row-level security.

Custom SQL

Custom SQL is used to provide special SQL syntax that the software does not generate. In the absence of row-level security, users with proper permissions for theInteractive Reporting document can modify custom SQL to produce ad hoc results.

When row-level security is in place, Custom SQL is affected in two ways:

- If the published Interactive Reporting document contains an open Custom SQL window, it is used as is when the user processes a query. No restrictions are applied to the SQL. However, the user cannot modify the SQL. While this can be a handy feature, care should be taken

when publishing Interactive Reporting documents that require custom SQL that they don't compromise the security requirements.

● If the user chooses the Reset button on the Custom SQL window, the SQL shown includes the data restrictions, and the original intent of the Custom SQL is lost and the user will not be able to get it back except by requesting the Interactive Reporting document from the server again.

Similar issues apply to the use of imported SQL.

Limits

The row-level security feature affects limits three ways:

First, if a user is restricted from accessing the content of certain columns, and the user attempts to show values when setting a limit on the restricted column, the restrictions will be applied to the SQL used to get the show values list. That way, the user cannot see and specify a value they would not otherwise be permitted to access.

Second, setting limits can result in some perhaps unexpected behavior when coupled with row-level security restrictions. This is best explained by example. In order to read the amount of sales, the user is restricted to a join on the STORE_ID column back to the stores table and in addition, the user can only see information for the STORE_ID when the state is Ohio. This user tries to set a limit on the unrestricted column STATE, and chooses something other than Ohio, thinking this a way to subvert the data restrictions. Unfortunately for that user, no sales amount information will be returned at all in this case. The SQL will specify "where state = 'user selected value' AND state = 'OH'". Obviously, the state cannot be two different values at the same time, so no data will be returned.

Of course, a user may try to set a limit on the CITY column instead of the STATE column, thinking the city name might exist in multiple states. As long as the need exists to access the amount of SALES column in the SALES table with identifying store information, though, the state limit will still be applied, and no data the user should not be able to see will be accessible to that user. It just will not prevent a user from getting a list of stores when sales data is not part of that list. Generally speaking, restricting access to facts based on the foreign key in the fact table(s) works best. If it is necessary to restrict the user's access to a list of stores, these dimension restrictions work best when applied to all columns in the dimension table with a limit on the source table.

For example, using the requirements described above to restrict the amount of sales information in Ohio only, with the same restriction on the dimension-only queries, do not apply any limit on access of the amount sales information except that it must be joined back to the STORES table on STORE_ID. Then, add a restriction for all columns in the STORES table, limiting it to only stores in Ohio. This limits access to both fact and dimension data.

Third, when setting a limit using Show Values, it has already been noted that any restrictions on the column to be limited are applied to the SQL that generates the show values list. For example, using the restrictions described in the previous paragraph, attempting to show the values list for the CITY column would be constrained to those cities in Ohio. Now, consider the following scenario.

The SALES FACT table also has a TRANSACTION DATE and PRODUCT_ID column. The transaction date column is tied back to a PERIODS table, where dates are broken down into quarters, fiscal years, months, and so on. In this somewhat contrived example, a restriction is placed on the PERIODS table, where values there are joined back to the SALES TRANSACTION table and restricted by PRODUCT_ID values in a certain range. The user sets a limit on fiscal year in the PERIODS table and invokes show values in the Limit dialog box to pick the range. Because of the restrictions in place, only one fiscal year is available, and the user picks it. Now, the user builds a query that does not request the FISCAL YEAR column itself but does reference the PRODUCT_ID field and processes it. This query returns, for the sake of argument, 100 rows. Now the user decides there is a need to see the fiscal year value and adds it to the Request line. Reprocessing the query only returns 50 rows.

Why? In the first case, PRODUCT_ID values outside of the range allowed when querying the FISCAL YEAR column will appear in the results. In the second case, the query will cause the restriction on PRODUCT_ID range to be included. Restrictions are only applied when a user requests to see data. There was no request to see the FISCAL YEAR column in the first case, except while setting the limit. There is no restriction on seeing PRODUCT_ID values. This example is contrived because restricting access to a dimension based on data in a fact table would be extremely unusual. Nevertheless, it illustrates a behavior that should be kept in mind when implementing restrictions.

Naming

Another way to set the restrictions described above is by a subquery. Instead of directly setting the limit on the STATE column, limit the values in the STORE_ID column in the STORES table. The constraint operator would be IN, and the constraint values field might look something like this:

```
(SELECT S.STORE_ID FROM STORES S WHERE S.STATE = 'OH')
```

Now, no matter what limit the user sets in the STORES table, they will always be constrained to the set of store IDs that are allowed based on their group memberships and their own user name. Even if a city outside of the allowed state is chosen, such as a city that exists in more than one state, any stores that other city has will not show up in the results.

Using a subquery can be useful when incorporating existing security systems into the row-level security feature of Interactive Reporting. When constructing constraints of this type, it is especially important to know SQL. For example, to specify a subquery, it helps to know that a subquery is always enclosed in parentheses. It is also important to know how EPM Workspace generates SQL and to follow its naming conventions to make sure the syntax generated is appropriate.

Table and Column Names

For the most part, simple security constraints reference directly the actual object names in the database. Case sensitivity in names should be observed when and where required. For subqueries and other SQL constructs used to specify the constraint values, it is sometimes useful to refer to objects already used by the software's SQL generation process. To do this:

- For table references in the FROM clause, use `From.tablename`, where *tablename* is the display name seen in the Interactive Reporting document's data model as the display name. If the display name contains a space, use the underscore to represent the space.

- For column names, use `tablename.columnname`, following the same rule as above, except that the `From.` should *not* be used.

Alias Names

By default, when processing user queries, table references in the SQL are always given alias names. Alias names are convenient shorthand for long table references, and they are required when trying to build correlated subqueries. These alias names take the form "ALn", when the 'n' is replaced by an arbitrary number. These numbers are usually based on the topic priority properties of the data model and can easily change based on several factors. For example, a user with the proper permissions can rearrange the topics, thus giving them different priorities. Because these numbers are dynamic, constraint specifications should never rely on them. Instead, by using the naming scheme above, the appropriate alias will be added to the constraints. So, if the requirement is a correlated subquery, the appropriate name will be given to the column in the outer query when referenced by the correlated subquery.

In the example above, using a subquery to restrict STORE_ID values to those in a specific state, it was neither necessary nor desirable to use the Hyperion Solutions naming conventions. There, the set of values was to be derived in a subquery that operated independently of the main query. Consequently, the 'From.' was not used in the FROM clause of the subquery, and the alias names were given in a way to not conflict with the alias names generated automatically by the software.

To use a correlated subquery, then, consider syntax like the following:

```
…FROM STORES S WHERE S.STORE_ID = Stores.Store_Id
```

The reference to the right of the equal sign will pick up the alias name from the outer query and thus provide the correct correlation requirements.

Troubleshooting Interactive Reporting

24

Connectivity Troubleshooting with dbgprint

If you experience difficulties logging on to or querying a database, you may be able to solve the problem with the help of a `dbgprint` (debug print) file. The `dbgprint` file automatically logs detailed status information which can assist you when troubleshooting platform configuration and connectivity problems. A `dbgprint` file will usually be requested by the Hyperion Solutions Customer Support personnel if they help you to solve a connectivity-related problem.

Although this topic is written with reference to Interactive Reporting, the `dbgprint` instructions apply to other Hyperion tools as well. If you experience continued connectivity problems with any of these tools, or have difficulty understanding the contents of a `dbgprint` file, you can forward the contents of the `dbgprint` file to Hyperion Solutions Customer Support for assistance.

Note: `dbgprint` is strictly a diagnostic tool, and the information contained is useful only for troubleshooting. Because Hyperion tools repeatedly log information to the file, `dbgprint` considerably slows application performance and should only be used if you encounter connectivity problems.

dbgprint and Interactive Reporting Studio

`dbgprint` is a text file. When placed in a directory containing the Interactive Reporting Studio executable (`brioqry.exe`), Interactive Reporting Studio automatically writes internal status information to the `dbgprint` file.

➤ To use `dbgprint` with Interactive Reporting Studio:

1 Exit Interactive Reporting Studio if it is still running.

2 Start a text editor (that is, Notepad, Simple Text, WordPad, and so on).

3   Save an empty file as **dbgprint** (with no file extension) to the directory which contains the Interactive Reporting Studio executable.

Typically the `brioqry.exe` is saved to `HYPERION_HOME\BIPlus\Client\bin\brioqry.exe`.

If you are using Notepad, you first have to type a space or character before you can save the file. Do not save the file with a file extension.

In the UNIX environment you need to create a file named DbgPrint. Please note the capitalization. This file will be placed in the bin directory for Interactive Reporting Studio.

If you are operating in a Windows environment, make sure that no extensions are appended to the end of the file name. If you are using Notepad as the text editor, the `.txt` extension is automatically appended to the saved file. **Make sure you remove any extension before you proceed to the next step.**

4   Close the text editor and start Interactive Reporting Studio by opening the actual application file.

In some instances `dbgprint` does not log information if Interactive Reporting Studio was started through an alias or shortcut. Instead, start Interactive Reporting Studio using the Finder (Macintosh), or Windows Explorer. Clicking a shortcut only works if the Start In field in the Properties dialog box for the shortcut shows the path to the `brioqry.exe` file.

5   Once Interactive Reporting Studio is running, recreate the steps which resulted in the previous error problem, or follow any instructions given to you by a Hyperion Solution customer support representative.

Typical things you may be asked to do are:

- Connect to the database
- Retrieve a list of tables
- Add tables to the work space
- Create and process a query
- Set a limit

6   Once you have completed the above tasks, quit Interactive Reporting Studio and open the `dbgprint` **file.**

7   View the contents of the `dbgprint` **file.**

The file should contain status information detailing the Interactive Reporting Studio logon session. You will probably be asked to either fax or email the contents of the `dbgprint` file to Hyperion Solutions.

If the file is blank, review the previous steps and repeat the process.

Note: If you need to run another `dbgprint` file, save the contents of the file with a unique name. Each time you run the `brioqry.exe` file, the existing `dbgprint` file is overwritten.

dbgprint and Interactive Reporting Web Client

`DbgPrint` files can also be used with Interactive Reporting Web Client.

➤ To use `dbgprint` with Interactive Reporting Web Client:

1 Shut down the Web browser.

2 Start a text editor (for example, Notepad, Simple Text, MS-WordPad, and so on).

3 Save an empty file as **dbgprint** to the folder where the browser executable (`.exe`) resides, for example, C:\Program Files\Internet Explorer directory).

   If you are using Notepad, you first have to type a space or character before you can save the file.

   If you are operating in a Windows environment, make sure that no extensions are appended to the end of the file name. If you are using Notepad as the text editor, the `.txt` extension is automatically appended to the saved file. **Make sure you remove any extension before you proceed to the next step.**

4 Start the Web browser.

   The `DbgPrint` file starts collecting debug information about the processing of the queries.

25

Interactive Reporting Studio INI Files

About Interactive Reporting Studio INI Files

The Interactive Reporting Studio INI files are simple text files that are used to store system and application settings. Table 193 shows each INI used by application and the type of information it contains.

Table 193   INI Files used in Interactive Reporting

Application	INI File	Content
Interactive Reporting Studio	BQFORMAT.INI	Stores locale and custom numeric formats.
	BQMETA0.INI	Stores OMI metadata settings for supported Metadata sources.
	BQTOOLS.IN	Stores Custom Menu definitions (only present if custom menus are defined)
Interactive Reporting Web Client	BRIOQPLG.INI	Stores regional settings for the plug-in
	BQFORMAT.INI	Stores locale and custom numeric formats
	BQTOOLS.INI	Stores Custom Menu definitions (only present if custom menus are defined
EPM Workspace		ALL server INI files are stored in sub-directories off of the HYPERION_HOME directory, and NOT in the Windows OS directory. Internationalized versions of the BQFORMAT.INI and BQMETA0.INI files are also present.
	BQFORMAT.INI	Stores locale and custom numeric formats for use with Oracle Enterprise Performance Management Workspace, Fusion Edition.
	INTELLIGENCE.INI	Stores default configuration settings for the user interface.
	SQR.INI	Stores SQR configuration info for the Job Factory

26

Customizing Interactive Reporting Web Client

Interactive Reporting Web Client Co-existence

Interactive Reporting Web Client users can work with different versions of Reporting and Analysis on the same client machine. To enable this feature, new Interactive Reporting Web Client installations include multiple versions of the application. Users can work with all servers older than Release 8.3 + server through Release 8.5. The feature was added because the Reporting and Analysis installation does not have a backward compatibility mechanism and only supports the 8.5 version of the Interactive Reporting Web Client. The correct version that gets used is determined at run-time when theInteractive Reporting document (BQY) is opened. For example, if a user opens n Interactive Reportingdocument (BQY) in a Release 8.3 server, the 8.3 version of the Interactive Reporting Web Client is launched to serve the request.

Two Interactive Reporting Web Client versions are installed in this release: an 8.3 version and the current 8.5 release. Each Interactive Reporting Web Client plug-in entity is separated on the disk and in the registry. Each version is installed in a separate folder under <IE Install Directory> \PLUGINS\Hyperion\<Insight Series Version> or under <Firefox Install Directory>\Plugins \Hyperion\<Insight Series Version>. A mechanism has been implemented to find and start a particular Interactive Reporting Web Client executable based on its version. The Interactive Reporting Web Client Launcher that provides 6x/8x co-existence has been enhanced for 8.3/8.5 and future co-existence. There is a single launcher in the system associated with the .bqy mime-type and file

If an user installs the 8.x version over theReporting and Analysis installation, then the use needs to installed the Oracle's Hyperion Reporting and Analysis version again in order to have both work correctly.

Locally Saved Interactive Reporting Documents

If a user is working with a locally saved Interactive Reporting document and saves it to disk from the Interactive Reporting Web Client, the file is appended with version information. If the user

opens the file again, the appropriate version of the Interactive Reporting Web Client is launched. After this, the Interactive Reporting Web Client checks the server version to connect to the server. If the server is newer than the current Interactive Reporting Web Client, the user is prompted to upgrade to the current version. In this case, the older version accesses the URL of the server, launches the zero administration page, and performs the upgrade. Interactive Reporting Web Client is opened from the disk using file name in the upgraded version.

If the Interactive Reporting document was created in previous versions of the Interactive Reporting Web Client or in the Oracle's Hyperion® Interactive Reporting Studio, the user is prompted for the version to use. The user can select from a list of versions of Interactive Reporting Web Client, including the 6.x version. If the user selects a 6.x version, no version record is attached to the document because there is no command on the Tools menu to open the document in another version of Interactive Reporting Web Client.

After Interactive Reporting Web Client is launched, the user can change the Interactive Reporting Web Client designated to open the document at any time by selecting the following options from the Tools menu:

- Change the Web Client version to open this file

- Make this Web Client default for locally savedInteractive Reporting document.

Interactive Reporting Web Client Mismatches

If there is a mismatch between the Interactive Reporting Web Client series required by the server and the user selects a compatible version, a file version record is saved and the document is reloaded.

Required Interactive Reporting Web Client series information is only provided by the 8.5 version servers. If this information is unavailable, it is assumed that the user is connected to either an 8.3 or 6.x server (which is detected by specific cookies and the absence of later version server cookies). Otherwise, it is assumed the user is connected to an 8.3 server. If the user opens a document in Interactive Reporting Web Client with a greater version number than is required by the server (for example, a document from a 6.x server is opened in an 8.3 server), the Save on Exit prompt is suppressed when the document is reloaded to prevent the user from upgrading accidentally to the newer series.

➤ To change the Web Client Version:

1   Select **Tools** , then **Change Web Client Version**.

    A message dialog is displayed stating: Several versions of Interactive Reporting Web Client are detected on your system. Choose the version of Interactive Reporting Web Client to use to open the document.

2   Select the version to use to open the document from the version drop-down.

3   To store the version selection for future documents, select **Remember the choice and don't ask again**.

4   Click **OK**.

    If Interactive Reporting Web Client expects a version other than the version selected, a message dialog is displayed stating: "You are trying to connect to server of different version

then this Interactive Reporting Web Client. Server version x.x is expected. Please try to use the server of this version or choose another Interactive Reporting Web Client from the following list." At this point, you can either select the expected version or another version." If you select a version other than expected, some functionality and features in the original document may not be available.

5 Click **OK**.

Increasing the Idle Connection Time-outs

Using a configurable polling time, administrators can control the frequency of messages that keep a connection to Interactive Reporting Web Client active. This feature allows queries of varied lengths to process from when running behind a proxy or firewall with finite idle-connection time-outs. By default there is no polling and this feature needs to be explicitly enabled.

➤ To increase the idle connection time-outs:

1 Open the ws.conf file from the <SERVLETS DEPLOYMENT>/config folder in your favorite editor.

2 Add the line:

```
WebClient.Applications.DAServlet.PollingTimeSec=0
```

3 Change the PollingTimeSec parameter value from 0 to the desired value.

- The Default value is 0 for the PollingTimeSec parameter value indicating there is no polling mechanism in place to keep the connection active.

- The PollingTimeSec parameter value is in seconds.

- The PollingTimeSec parameter value is less than the proxy and firewall timeout value (otherwise this configuration change would have no effect).

- Setting the PollingTimeSec parameter value too low increases network traffic. For example: if a firewall is set for a 15 min idle timeout, set the polling time to be 720 sec (== 12 min). There is no need to set the value to 10 seconds in a case like this, as it creates unnecessary network overhead.

  Once this entry is added to the ws.conf file, the Servlet Configurator (<INSTALL_HOME/servlets/bin, servletconfig.bat for Windows and servletconfig.sh for UNIX can be used to manipulate the value.

4 Save the file.

Opening Interactive Reporting Web Client in Multiple Browser Windows

Interactive Reporting Web Client can be launched in a single browser window or multiple browser windows. Use this feature to display either one or several Interactive Reporting documents (BQY) at the same time. When multiple browser windows are opened for the same

Interactive Reporting documents (BQY), the last saved document is the one saved to the repository.

➤ To open Interactive Reporting Web Client in multiple browser windows:

1 Open the ws.conf file from the <SERVLETS DEPLOYMENT>/config folder in your favorite editor.

2 Add the line:

```
WebClient.Applications.DAServlet.OpenNewWindow=true
```

3 Save the file.

➤ To open Interactive Reporting Web Client in a single browser window:

1 Open the ws.conf file from the <SERVLETS DEPLOYMENT>/config folder in your favorite editor.

2 Add the line:

```
WebClient.Applications.DAServlet.OpenNewWindow=false
```

3 Save the file.

Interactive Reporting Web Client Installer Customization

When installing Interactive Reporting Web Client, customization options are available which enable you to optionally install:

- Help files
- Font files
- Translation files

This section also describes how to modify the Configuration file to customize the installation of Interactive Reporting Web Client.

Installing Help and Font Files

➤ To install help and font files:

1 In Explore mode, select an Oracle's Hyperion® Interactive Reporting document.

2 Select **Open As** on the shortcut menu.

A submenu is displayed.

3 Select **Interactive Reporting Web Client** from the submenu.

The Interactive Reporting Web Client Install wizard is displayed.

4 Select to install the help files and font files.

5 Click **Next**.

Complete the installation as prompted.

Selecting the Number of Installed Languages

Users do not always need full internationalization support and ability to switch the user interface to different languages. Your administration determines if only one language (which matches the user's login language) is implemented, or a subset of installed languages is installed.

To allow the user to install additional translations files and custom versions of ini files, the files can be placed in an ad hoc folder ("CustomFiles") along with the installer package (for example workspace\zeroadmin\component\Insight\CustomFiles), and relative path to which it must be transferred to in the installer. Three methods are available for transferring files to the user machine.

First, the administrator can transfer files into the Interactive Reporting Web Client home directory (for example, c:\program files\internet explorer\plugins\hyperion). In this case, files are placed corresponding to the relative path in which they stored in the "CustomFiles" folder. For example if briottbl.txt file is placed in the :CustomFiles\9.3\Translations\Latin folder on the server, it is transferred to c:\program files\internet explorer\plugins\hyperion\9.0.2\Translations\Latin\briottbl.txt. This reduces the changes to the subfolder name corresponding to Interactive Reporting Web Client version for each new version of the installer (for example, changing the \9.0.2\Translations to \9.1\Translations). To simplify the maintenance of custom installations, a special folder name "PluginFolder" is supported. If briottbl.txt is placed in the CustomFiles\PluginFolder\Translations\Latin folder on the server, it is transferred to the folder corresponding to the version of the installed Interactive Reporting Web Client (for example. \9.0.2\Translations\Latin\briottbl.txt, \9.1\Translations\Latin\briottbl.txt etc.).

In the second case, the files can be transferred into the Windows system directory. This option is supported only for updating existing Interactive Reporting Web Client, .ini files (brioqplg_ev.ini etc.) and it is not possible to transfer any other file to the Windows system directory.

The third option enables you to transfer *.ttf font files. These files can be transferred to the Windows Fonts folder and properly registered. The registration requires that font name be made available to the installer (the administrator might write a script to retrieve the font names).

Editing the Configuration File

The easiest way to customize theInteractive Reporting Web Client is by editing the settings.ini file manually. The structure of this file should be as follows.

```
OneTranslationLanguage=<visible|invisible>,<enabled|disabled>
SilentInstall=<true|false>
<Component>=<visible|invisible>,<installed|skipped>
InstallFile=<path to file>
InstalledTranslationLanguages=<language>,<language>,
```

<Component>—is any of supported for customized installation components (Insight8.3, Font etc.).

<visible|invisible>—defines whether a given option is available to user to change or not.

OneTranslationLanguage—corresponds to "install only one language matching user's login" option.

SilentInstall—sets the silent install mode. If the option is set to true, neither the component selection dialog or the installation complete notification dialog are shown, and no user interaction is required.

InstallFile and <path to file>—provides information to the installer about additional files that should be installed. There can be an arbitrary number of InstallFiles entries each defining file to install. The path should be relative to the CustomFiles folder.

InstalledTranslationLanguages—defines the subset of installed translations. The <language> parameter is any of the supported translation languages (French,Italian,Japanese etc.). If an option is present in the settings.ini, only translation files for selected languages are installed.

Configuration File Examples

Examples of how you might configure the settings.ini are shown below:

In the default configuration, Insight 8.3 is installed as well as all translation languages:

```
OneTranslationLanguage=invisible,disabled
SilentInstall=false
Insight8.3=invisible,installed
Help=visible,installed
Fonts=visible,installed
Translation=invisible,installed
```

Instructions are given to install only one translation file matching the language of the installation:

```
OneTranslationLanguage=invisible,enabled
SilentInstall=false
Insight8.3=invisible,installed
Help=visible,installed
Fonts=visible,installed
Translation=invisible,installed
```

Instructions are given to install only a selected subset of translation languages:

```
OneTranslationLanguage=invisible,enabled
SilentInstall=false
Insight8.3=invisible,installed
Help=visible,installed
Fonts=visible,installed
Translation=invisible,installed
InstalledTranslationLanguages=French,Italian,Japanese
```

Instructions are given not to install Insight 8.3:

```
OneTranslationLanguage=invisible,disabled
SilentInstall=false
Insight8.3=invisible,skipped
Help=visible,installed
Fonts=visible,installed
Translation=invisible,installed
```

Instructions are given to allow the user the choice of installing Insight 8.3, install a selected subset of translation languages, but not to install Fonts or Help files:

```
OneTranslationLanguage=invisible,disabled
SilentInstall=false
Insight8.3=visible,skipped
Help=invisible,skipped
Fonts=invisible,skipped
Translation=invisible,installed
InstalledTranslationLanguages=French,Italian,Japanese
```

Instructions are given not to install anything except the main Oracle's Hyperion® Interactive Reporting Web Client plugin and translations, and to perform silent install (no user interaction). Note that it doesn't matter that the individual component's state is set to visible. The SilentInstall switch overrides it:

```
OneTranslationLanguage=invisible,disabled
SilentInstall=true
Insight8.3=visible,skipped
Help=visible,skipped
Fonts=visible,skipped
Translation=invisible,installed
```

Instructions are given to install additional custom translation files and the non-default brioqplg_ev.ini:

```
OneTranslationLanguage=invisible,disabled
SilentInstall=false
Insight8.3=invisible,installed
Help=visible,installed
Fonts=visible,installed
Translation=invisible,installed
InstallFile=PluginFolder\Translations\Latin\briottbl.txt
InstallFile=Translations\Latin\briottbl.txt ; the launcher's translation
InstallFile=brioqplg_ev.ini
```

Glossary

! See bang character.

#MISSING See missing data.

access permissions A set of operations that a user can perform on a resource.

accessor Input and output data specifications for data-mining algorithms.

account blocking The process by which accounts accept input data in the consolidated file. Blocked accounts do not receive their value through the additive consolidation process.

account eliminations Accounts which have their values set to zero in the consolidated file during consolidation.

account type A property that determines how an account's value flows over time and its sign behavior. Account type options can include expense, income, asset, liability, and equity.

accountability map A visual, hierarchical representation of the responsibility, reporting, and dependency structure of the accountability teams (also known as critical business areas) in an organization.

active service A service whose Run Type is set to Start rather than to Hold.

active-active high availability system A system in which all the available members can service requests, and no member is idle. An active-active system generally provides more scalability options than an active-passive system. Contrast with active-passive high availability system.

active-passive high availability system A system with active members, which are always servicing requests, and passive members that are activated only when an active member fails. Contrast with active-active high availability system.

activity-level authorization Defines user access to applications and the types of activities they can perform on applications, independent of the data that will be operated on.

ad hoc report An online analytical query that an end user creates dynamically.

adapter Software that enables a program to integrate with data and metadata from target and source systems.

adaptive states Interactive Reporting Web Client level of permission.

adjustment See journal entry.

Advanced Relational Access The integration of a relational database with an Essbase multidimensional database so that all data remains in the relational database and is mapped to summary-level data in the Essbase database.

agent An Essbase server process that starts and stops applications and databases, manages connections from users, and handles user-access security. The agent is referred to as ESSBASE.EXE.

aggregate cell A cell comprising several cells. For example, a data cell that uses Children(Year) expands to four cells containing Quarter 1, Quarter 2, Quarter 3, and Quarter 4 data.

aggregate function A type of function, such as sum or calculation of an average, that summarizes or performs analysis on data.

aggregate limit A limit placed on an aggregated request line item or aggregated metatopic item.

aggregate storage database The database storage model designed to support large-scale, sparsely distributed data which is categorized into many, potentially large dimensions. Upper level members and formulas are dynamically calculated, and selected data values are aggregated and stored, typically with improvements in overall aggregation time.

aggregate view A collection of aggregate cells based on the levels of the members within each dimension. To reduce calculation time, values are pre-aggregated and stored as aggregate views. Retrievals start from aggregate view totals and add up from there.

aggregation The process of rolling up and storing values in an aggregate storage database; the stored result of the aggregation process.

aggregation script In aggregate storage databases only, a file that defines a selection of aggregate views to be built into an aggregation.

alias table A table that contains alternate names for members.

alternate hierarchy A hierarchy of shared members. An alternate hierarchy is based upon an existing hierarchy in a database outline, but has alternate levels in the dimension. An alternate hierarchy allows the same data to be seen from different points of view.

ancestor A branch member that has members below it. For example, the members Qtr2 and 2006 are ancestors of the member April.

appender A Log4j term for destination.

application 1) A software program designed to run a specific task or group of tasks such as a spreadsheet program or database management system. 2) A related set of dimensions and dimension members that are used to meet a specific set of analytical requirements, reporting requirements, or both.

application administrator A person responsible for setting up, configuring, maintaining, and controlling an application. Has all application privileges and data access permissions.

application currency The default reporting currency for the application.

Application Migration Utility A command-line utility for migrating applications and artifacts.

application server cluster A loosely joined group of application servers running simultaneously, working together for reliability and scalability, and appearing to users as one application server instance. See also vertical application cluster and horizontal application cluster.

area A predefined set of members and values that makes up a partition.

arithmetic data load A data load that performs operations on values in the database, such as adding 10 to each value.

artifact An individual application or repository item; for example, scripts, forms, rules files, Interactive Reporting documents, and financial reports. Also known as an object.

assemblies Installation files for EPM System products or components.

asset account An account type that stores values that represent a company's assets.

assignment The association of a source and destination in the allocation model that controls the direction of allocated costs or revenue flow.

attribute A characteristic of a dimension member. For example, Employee dimension members may have attributes of Name, Age, or Address. Product dimension members can have several attributes, such as a size and flavor.

attribute association A relationship in a database outline whereby a member in an attribute dimension describes a characteristic of a member of its base dimension. For example, if product 100-10 has a grape flavor, the product 100-10 has the Flavor attribute association of grape. Thus, the 100-10 member of the Product dimension is associated with the Grape member of the Flavor attribute dimension.

Attribute Calculations dimension A system-defined dimension that performs these calculation operations on groups of members: Sum, Count, Avg, Min, and Max. This dimension is calculated dynamically and is not visible in the database outline. For example, using the Avg member, you can calculate the average sales value for Red products in New York in January.

attribute dimension A type of dimension that enables analysis based on the attributes or qualities of dimension members.

attribute reporting A reporting process based on the attributes of the base dimension members. See also base dimension.

attribute type A text, numeric, Boolean, date, or linked-attribute type that enables different functions for grouping, selecting, or calculating data. For example, because the Ounces attribute dimension has the type numeric, the number of ounces specified as the attribute of each product can be used to calculate the profit per ounce for that product.

authentication Verification of identity as a security measure. Authentication is typically based on a user name and password. Passwords and digital signatures are forms of authentication.

authentication service A core service that manages one authentication system.

auto-reversing journal A journal for entering adjustments that you want to reverse in the next period.

automated stage A stage that does not require human intervention; for example, a data load.

axis (1) A straight line that passes through a graphic used for measurement and categorization. (2) A report aspect used to arrange and relate multidimensional data, such as filters, pages, rows, and columns. For example, for a data query in Simple Basic, an axis can define columns for values for Qtr1, Qtr2, Qtr3, and Qtr4. Row data would be retrieved with totals in the following hierarchy: Market, Product.

backup A duplicate copy of an application instance.

balance account An account type that stores unsigned values that relate to a particular time.

balanced journal A journal in which the total debits equal the total credits.

bang character (!) A character that terminates a series of report commands and requests information from the database. A report script must be terminated with a bang character; several bang characters can be used within a report script.

base currency The currency in which daily business transactions are performed.

base dimension A standard dimension that is associated with one or more attribute dimensions. For example, assuming products have flavors, the Product dimension is the base dimension for the Flavors attribute dimension.

base entity An entity at the bottom of the organization structure that does not own other entities.

batch calculation Any calculation on a database that is done in batch; for example, a calculation script or a full database calculation. Dynamic calculations are not considered to be batch calculations.

batch file An operating system file that can call multiple ESSCMD scripts and run multiple sessions of ESSCMD. On Windows-based systems, batch files have BAT file extensions. On UNIX, batch files are written as a shell script.

Batch Loader An FDM component that enables the processing of multiple files.

batch POV A collection of all dimensions on the user POV of every report and book in the batch. While scheduling the batch, you can set the members selected on the batch POV.

batch processing mode A method of using ESSCMD to write a batch or script file that can be used to automate routine server maintenance and diagnostic tasks. ESSCMD script files can execute multiple commands and can be run from the operating system command line or from within operating system batch files. Batch files can be used to call multiple ESSCMD scripts or run multiple instances of ESSCMD.

block The primary storage unit which is a multidimensional array representing the cells of all dense dimensions.

block storage database The Essbase database storage model categorizing and storing data based on the sparsity of data values defined in sparse dimensions. Data values are stored in blocks, which exist only for sparse dimension members for which there are values.

Blocked Account An account that you do not want calculated in the consolidated file because you want to enter it manually.

book 1) In Financial Reporting, a container that holds a group of similar documents. Books may specify dimension sections or dimension changes. 2) In Data Relationship Management, a collection of exports that can be run together as a group. Export results can be combined together or output separately.

book POV The dimension members for which a book is run.

bookmark A link to a reporting document or a Web site, displayed on a personal page of a user. The types of bookmarks are My Bookmarks and image bookmarks.

bounding rectangle The required perimeter that encapsulates the Interactive Reporting document content when embedding Interactive Reporting document sections in a personal page, specified in pixels for height and width or row per page.

broadcast message A simple text message sent by an administrator to a user who is logged on to a Planning application. The message details information such as system availability, notification of application refresh, or application backups.

build method A method used to modify database outlines. Choice of a build method is based on the format of data in data source files.

business process A set of activities that collectively accomplish a business objective.

business rules Logical expressions or formulas that are created within an application to produce a desired set of resulting values.

cache A buffer in memory that holds data temporarily.

calc script A set of commands that define how a database is consolidated or aggregated. A calculation script may also contain commands that specify allocation and other calculation rules separate from the consolidation process.

Calculated Accounts Accounts with formulas that you cannot alter. These formulas are fixed to maintain the accounting integrity of the model that you are building. For example, the formula for Net Income, a Calculated Account, is modeled into Strategic Finance and cannot be changed in historical or forecast periods.

calculated member in MaxL DML A member designed for analytical purposes and defined in the optional WITH section of a MaxL DML query.

Calculation Manager A module of Enterprise Performance Management Architecture (EPMA) that Planning and Financial Management users can use to design, validate, and administrate business rules in a graphical environment. c

calculation status A consolidation status that indicates that some values or formula calculations have changed. You must reconsolidate to get the correct values for the affected entity.

calendar User-defined time periods and their relationship to each other. Q1, Q2, Q3, and Q4 comprise a calendar or fiscal year.

cascade The process of creating multiple reports for a subset of member values.

Catalog pane An area that displays a list of elements available to the active section. If Query is the active section, a list of database tables is displayed. If Pivot is the active section, a list of results columns is displayed. If Dashboard is the active section, a list of embeddable sections, graphic tools, and control tools are displayed.

categories Groupings by which data is organized. For example, Month.

cause and effect map A map that depicts how the elements that form your corporate strategy relate and how they work together to meet your organization's strategic goals. A Cause and Effect map tab is automatically created for each Strategy map.

CDF See custom-defined function.

CDM See custom-defined macro.

cell (1) The data value at the intersection of dimensions in a multidimensional database; the intersection of a row and a column in a worksheet. (2) A logical group of nodes belonging to one administrative domain.

cell note A text annotation for a cell in an Essbase database. Cell notes are a type of LRO.

CHANGED status Consolidation status that indicates data for an entity has changed.

chart template A template that defines the metrics to display in Workspace charts.

child A member with a parent above it in the database outline.

choice list A list of members that a report designer can specify for each dimension when defining the report's point of view. A user who wants to change the point of view for a dimension that uses a choice list can select only the members specified in that defined member list or those members that meet the criteria defined in the function for the dynamic list.

clean block A data block in which the database is fully calculated, if a calculation script calculates all dimensions at once, or if the SET CLEARUPDATESTATUS command is used in a calculation script.

cluster An array of servers or databases that behave as a single resource which share task loads and provide failover support; eliminates one server or database as a single point of failure in a system.

cluster interconnect A private link used by a hardware cluster for heartbeat information, to detect node failure.

cluster services Software that manages cluster member operations as a system. With cluster services, you can define a set of resources and services to monitor through a heartbeat mechanism between cluster members and to move these resources and services to a different cluster member as efficiently and transparently as possible.

clustered bar charts Charts in which categories are viewed side-by-side; used only with vertical bar charts.

code page A mapping of bit combinations to a set of text characters. Different code pages support different sets of characters. Each computer contains a code page setting for the character set requirements of the language of the computer user. In the context of this document, code pages map characters to bit combinations for non-Unicode encodings. See also encoding.

column In Data Relationship Management, a field of data associated with an import source or the results of a query, compare, validation, or export.

committed access An Essbase Kernel Isolation Level setting that affects how Essbase handles transactions. Under committed access, concurrent transactions hold long-term write locks and yield predictable results.

computed item A virtual column (as opposed to a column that is physically stored in the database or cube) that can be calculated by the database during a query, or by Interactive Reporting Studio in the Results section. Computed items are calculations of data based on functions, data items, and operators provided in the dialog box and can be included in reports or reused to calculate other data.

connection file See Interactive Reporting connection file (.oce)

consolidated file (Parent) A file into which all of the business unit files are consolidated; contains the definition of the consolidation.

consolidation The process of aggregating data from dependent entities to parent entities. For example, if the dimension Year consists of the members Qtr1, Qtr2, Qtr3, and Qtr4, its consolidation is Year.

consolidation file (*.cns) A graphical interface that enables you to add, delete, or move Strategic Finance files in the consolidation process using either a Chart or Tree view. It also enables you to define and modify the consolidation.

consolidation rule The rule that is executed during the consolidation of the node of the hierarchy. This rule can contain customer-specific formulas appropriate for the correct consolidation of parent balances. Elimination processing can be controlled within these rules.

content Information stored in the repository for any type of file.

content browser A component that eanbles users to browse and select content to be placed on a Workspace Page.

context variable A variable that is defined for a particular task flow to identify the context of the taskflow instance.

contribution The value added to a parent from a child entity. Each child has a contribution to its parent.

controls groups Groupings used in FDM to maintain and organize certification and assessment information, especially helpful for meeting Sarbanes-Oxley requirements.

conversion rate See exchange rate.

cookie A segment of data placed on your computer by a Web site.

correlated subqueries Subqueries that are evaluated once for every row in the parent query; created by joining a topic item in the subquery with a topic in the parent query.

critical business area (CBA) An individual or a group organized into a division, region, plant, cost center, profit center, project team, or process; also called accountability team or business area.

critical success factor (CSF) A capability that must be established and sustained to achieve a strategic objective; owned by a strategic objective or a critical process and is a parent to one or more actions.

crosstab reporting Reporting that categorizes and summarizes data in table format. The table cells contain summaries of the data that fit within the intersecting categories. For example, a crosstab report of product sales information could show size attributes, such as Small and Large, as column headings and color attributes, such as Blue and Yellow, as row headings. The cell in the table where Large and Blue intersect could contain the total sales of all Blue products that are sized Large.

cube A block of data that contains three or more dimensions. An Essbase database is a cube.

cube deployment In Essbase Studio, the process of setting load options for a model to build an outline and load data into an Essbase application and database.

cube schema In Essbase Studio, the metadata elements, such as measures and hierarchies, representing the logical model of a cube.

currency conversion A process that converts currency values in a database from one currency into another. For example, to convert one U. S. dollar into the European euro, the exchange rate (for example, 0.923702) is multiplied by the dollar (1* 0.923702). After conversion, the European euro amount is .92.

Currency Overrides A feature allowing the selected input method for any input period to be overridden to enable input of that period's value as Default Currency/Items. To override the input method, enter a pound sign (#) before or after the number.

currency partition A dimension type that separates local currency members from a base currency, as defined in an application. Identifies currency types, such as Actual, Budget, and Forecast.

custom calendar Any calendar created by an administrator.

custom dimension A dimension created and defined by users. Channel, product, department, project, or region could be custom dimensions.

custom property A property of a dimension or dimension member that is created by a user.

custom report A complex report from the Design Report module, composed of any combination of components.

custom-defined function (CDF) Essbase calculation functions developed in Java and added to the standard Essbase calculation scripting language using MaxL. See also custom-defined macro.

custom-defined macro (CDM) Essbase macros written with Essbase calculator functions and special macro functions. Custom-defined macros use an internal Essbase macro language that enables the combination of calculation functions and they operate on multiple input parameters. See also custom-defined function.

cycle through Perform multiple passes through a database while calculating it.

dashboard A collection of metrics and indicators that provide an interactive summary of your business. Dashboards enable you to build and deploy analytic applications.

data cache A buffer in memory that holds uncompressed data blocks.

data cell See cell.

data file cache A buffer in memory that holds compressed data (PAG) files.

data form A grid display that enables users to enter data into the database from an interface such as a Web browser, and to view and analyze data or related text. Certain dimension member values are fixed, giving users a specific view into the data.

data function Function that computes aggregate values, including averages, maximums, counts, and other statistics that summarize groupings of data.

data load location In FDM, a reporting unit responsible for submitting source data into the target system. Typically, one FDM data load location exists for each source file loaded to the target system.

data load rules A set of criteria that determines how to load data from a text-based file, a spreadsheet, or a relational data set into a database.

data lock A feature that prevents changes to data according to specified criteria, such as a period or scenario.

data mining The process of searching through an Essbase database for hidden relationships and patterns in a large amount of data.

data model A representation of a subset of database tables.

data value See cell.

database connection A file that stores definitions and properties used to connect to data sources and enables database references to be portable and widely used.

date measure In Essbase, a member tagged as Date in the dimension where measures are represented. The cell values are displayed as formatted dates. Dates as measures can be useful for analysis types that are difficult to represent using the Time dimension. For example, an application may need to track acquisition dates for a series of capital assets, but the acquisition dates span too large a period to allow for feasible Time dimension modeling. See also typed measure.

Default Currency Units The unit scale of data. For example, If you select to define your analysis in thousands and enter 10, this unit is interpreted as 10,000.

dense dimension In block storage databases, a dimension likely to contain data for every combination of dimension members. For example, time dimensions are often dense because they can contain all combinations of all members. Contrast with sparse dimension.

dependent entity An entity that is owned by another entity in the organization.

derived text measure In Essbase Studio, a text measure whose values are governed by a predefined rule expressed as a range. For example, a derived text measure, called "Sales Performance Index," based on a measure Sales, could consist of the values "High," "Medium," and "Low." This derived text measure is defined to display "High," "Medium," and "Low" depending on the range in which the corresponding sales values fall. See also text measure.

descendant Any member below a parent in the database outline. In a dimension that includes years, quarters, and months, the members Qtr2 and April are descendants of the member Year.

Design Report An interface in Web Analysis Studio for designing custom reports, from a library of components.

destination 1) In Business Rules, a block of the database where calculated values are stored; 2) In Profitability and Cost Management, the association of a source and destination in the allocation model that controls the direction of allocated costs or revenue flow.

destination currency The currency to which balances are converted. You enter exchange rates and convert from the source currency to the destination currency. For example, when you convert from EUR to USD, the destination currency is USD.

detail chart A chart that provides the detailed information that you see in a Summary chart. Detail charts appear in the Investigate Section in columns below the Summary charts. If the Summary chart shows a Pie chart, then the Detail charts below represent each piece of the pie.

dimension A data category used to organize business data for the retrieval and preservation of values. Dimensions usually contain hierarchies of related members grouped within them. For example, a Year dimension often includes members for each time period, such as quarters and months.

dimension build The process of adding dimensions and members to an Essbase outline.

dimension build rules Specifications, similar to data load rules, that Essbase uses to modify an outline. The modification is based on data in an external data source file.

dimension tab In the Pivot section, the tab that enables you to pivot data between rows and columns.

dimension table (1) A table that includes numerous attributes about a specific business process. (2) In Essbase Integration Services, a container in the OLAP model for one or more relational tables that define a potential dimension in Essbase.

dimension type A dimension property that enables the use of predefined functionality. Dimensions tagged as time have a predefined calendar functionality.

dimensionality In MaxL DML, the represented dimensions (and the order in which they are represented) in a set. For example, the following set consists of two tuples of the same dimensionality, because they both reflect the dimensions (Region, Year): { (West, Feb), (East, Mar) }

direct rate A currency rate that you enter in the exchange-rate table. The direct rate is used for currency conversion. For example, to convert balances from JPY to USD, in the exchange-rate table, enter a rate for the period/scenario where the source currency is JPY and the destination currency is USD.

dirty block A data block containing cells that have been changed since the last calculation. Upper-level blocks are marked as dirty if their child blocks are dirty (that is, if they have been updated).

display type One of three Web Analysis formats saved to the repository: spreadsheet, chart, and pinboard.

dog-ear The flipped page corner in the upper-right corner of the chart header area.

domain In data mining, a variable representing a range of navigation within data.

drill-down Navigation through the query result set using the dimensional hierarchy. Drilling down moves the user perspective from aggregated data to detail. For example, drilling down can reveal hierarchical relationships between years and quarters or quarters and months.

drill-through The navigation from a value in one data source to corresponding data in another source.

driver In Profitability and Cost Management, an allocation method that describes the mathematical relationship between the sources that use the driver and the destinations to which those sources allocate cost or revenue. For Business Modeling, see also cost driver and activity driver.

duplicate alias name A name that occurs more than once in an alias table and can be associated with more than one member in a database outline. Duplicate alias names can be used with duplicate member outlines only.

duplicate member name Multiple occurrences of a member name in a database, with each occurrence representing a different member. For example, a database has two members named New York. One member represents New York state and the other member represents New York city.

duplicate member outline A database outline containing duplicate member names.

Dynamic Calc and Store members Members in a block storage outline that Essbase calculates only upon the first retrieval of the value. Essbase then stores the calculated value in the database. Subsequent retrievals do not require calculating.

Dynamic Calc members Members in a block storage outline that Essbase calculates only at retrieval time. Essbase discards calculated values after completing the retrieval request.

dynamic calculation In Essbase, a calculation that occurs only when you retrieve data on a member that is tagged as Dynamic Calc or Dynamic Calc and Store. The member's values are calculated at retrieval time instead of being precalculated during batch calculation.

dynamic hierarchy In aggregate storage database outlines only, a hierarchy in which members are calculated at retrieval time.

dynamic member list A system-created named member set that is based on user-defined criteria. The list is refreshed automatically whenever it is referenced in the application. As dimension members are added and deleted, the list automatically reapplies the criteria to reflect the changes.

dynamic reference A pointer in the rules file to header records in a data source.

dynamic report A report containing data that is updated when you run the report.

Dynamic Time Series A process that performs period-to-date reporting in block storage databases.

dynamic view account An account type indicating that account values are calculated dynamically from the data that is displayed.

Eliminated Account An account that does not appear in the consolidated file.

elimination The process of zeroing out (eliminating) transactions between entities within an organization.

employee A user responsible for, or associated with, specific business objects. Employees need not work for an organization; for example, they can be consultants. Employees must be associated with user accounts, for authorization purposes.

encoding A method for mapping bit combinations to characters for creating, storing, and displaying text. Each encoding has a name; for example, UTF-8. Within an encoding, each character maps to a specific bit combination; for example, in UTF-8, uppercase A maps to HEX41. See also code page, locale.

ending period A period enabling you to adjust the date range in a chart. For example, an ending period of "month" produces a chart showing information through the end of the current month.

Enterprise View An Administration Services feature that enables management of the Essbase environment from a graphical tree view. From Enterprise View, you can operate directly on Essbase artifacts.

entity A dimension representing organizational units. Examples: divisions, subsidiaries, plants, regions, products, or other financial reporting units.

EPM Oracle home A subdirectory of Middleware home containing the files required by EPM System products. The EPM Oracle home location is specified during installation with EPM System Installer.

Equity Beta The riskiness of a stock, measured by the variance between its return and the market return, indicated by an index called "beta." For example, if a stock's return normally moves up or down 1.2% when the market moves up or down 1%, the stock has a beta of 1.2.

essbase.cfg An optional configuration file for Essbase. Administrators may edit this file to customize Essbase Server functionality. Some configuration settings may also be used with Essbase clients to override Essbase Server settings.

EssCell A function entered into a cell in Essbase Spreadsheet Add-in to retrieve a value representing an intersection of specific Essbase database members.

ESSCMD A command-line interface for performing Essbase operations interactively or through batch script files.

ESSLANG The Essbase environment variable that defines the encoding used to interpret text characters. See also encoding.

ESSMSH See MaxL Shell.

exceptions Values that satisfy predefined conditions. You can define formatting indicators or notify subscribing users when exceptions are generated.

exchange rate type An identifier for an exchange rate. Different rate types are used because there may be multiple rates for a period and year. Users traditionally define rates at period end for the average rate of the period and for the end of the period. Additional rate types are historical rates, budget rates, forecast rates, and so on. A rate type applies to a specific time.

expense account An account that stores periodic and year-to-date values that decrease net worth if they are positive.

Extensible Markup Language (XML) A language comprising a set of tags used to assign attributes to data that can be interpreted between applications according to a schema.

external authentication Logging on to Oracle EPM System products with user information stored outside the application. The user account is maintained by the EPM System, but password administration and user authentication are performed by an external service, using a corporate directory such as Oracle Internet Directory (OID) or Microsoft Active Directory (MSAD).

externally triggered events Non-time-based events for scheduling job runs.

Extract, Transform, and Load (ETL) Data-source-specific programs for extracting data and migrating it to applications.

extraction command An Essbase reporting command that handles the selection, orientation, grouping, and ordering of raw data extracted from a database; begins with the less-than (<) character.

fact table The central table in a star join schema, characterized by a foreign key and elements drawn from a dimension table. This table typically contains numeric data that can be related to all other tables in the schema.

failover The ability to switch automatically to a redundant standby database, server, or network if the primary database, server, or network fails or is shut down. A system that is clustered for failover provides high availability and fault tolerance through server redundancy and fault-tolerant hardware, such as shared disks.

Favorites gadget A gadget that contains links to Reporting and Analysis documents and URLs. See also gadget.

file delimiter A character, such as a comma or tab, that separates fields in a data source.

filter A constraint on data sets that restricts values to specific criteria; for example, to exclude certain tables, metadata, or values, or to control access.

flow account An unsigned account that stores periodic and year-to-date values.

footer Text or images at the bottom of report pages, containing dynamic functions or static text such as page numbers, dates, logos, titles or file names, and author names.

format string 1) In Essbase, a method for transforming the way cell values are displayed. 2) In FDM, a parameter of a Format or Formatted Date derived property that indicates the format in which a property value should be returned.

formula In Data Relationship Management, business logic used by a derived property to dynamically calculate a property value.

frame An area on the desktop. Two main areas: the navigation and workspace frames.

free-form grid An object for presenting, entering, and integrating data from different sources for dynamic calculations.

free-form reporting Creating reports by entering dimension members or report script commands in worksheets.

function In Data Relationship Management, a syntactic element of a derived property formula that accepts parameters and returns dynamic values.

gadget A simple, specialized, lightweight application that provides easy viewing of EPM content and enables access to core Reporting and Analysis functionality.

geneology data Additional data that is optionally generated after allocation calculations. This data enables reporting on all cost or revenue flows from start to finish through all allocation steps.

generation A layer in a hierarchical tree structure that defines member relationships in a database. Generations are ordered incrementally from the top member of the dimension (generation 1) down to the child members. Use the unique generation name to identify a layer in the hierarchical tree structure.

generic jobs Non-SQR Production Reporting or non-Interactive Reporting jobs.

global report command A command in a running report script that is effective until it is replaced by another global command or the file ends.

grid POV A means for specifying dimension members on a grid without placing dimensions in rows, columns, or page intersections. A report designer can set POV values at the grid level, preventing user POVs from affecting the grid. If a dimension has one grid value, you put the dimension into the grid POV instead of the row, column, or page.

group A container for assigning similar access permissions to multiple users.

GUI Graphical user interface

hardware cluster a collection of computers that provides a single view of network services (for example, an IP address) or application services (such as databases and Web servers) to clients of these services. Each node in a hardware cluster is a standalone server that runs its own processes. These processes can communicate with one another to form what looks like a single system that cooperatively provides applications, system resources, and data to users.

high availability A system attribute that enables an application to continue to provide services in the presence of failures. This is achieved through removal of single points of failure, with fault-tolerant hardware, as well as server clusters; if one server fails, processing requests are routed to another server.

Historical Average An average for an account over a number of historical periods.

holding company An entity that is part of a legal entity group, with direct or indirect investments in all entities in the group.

horizontal application server cluster A cluster with application server instances on different machines.

host A server on which applications and services are installed.

host properties Properties pertaining to a host, or if the host has multiple Oracle EPM homes, to an Oracle EPM home.

Hybrid Analysis An analysis mapping low-level data stored in a relational database to summary-level data stored in Essbase, combining the mass scalability of relational systems with multidimensional data.

hyperlink A link to a file, a Web page, or an intranet HTML page.

Hypertext Markup Language (HTML) A programming language specifying how Web browsers display data.

identity A unique identification for a user or group in external authentication.

image bookmarks Graphic links to Web pages or repository items.

IMPACTED status A status that indicates changes in child entities consolidating into parent entities.

implied share A member with one or more children but only one that is consolidated, so the parent and child share a value.

import format In FDM, the definition of the structure of the source file that enables the loading of a source data file to an FDM data-load location.

inactive group A group for which an administrator has deactivated system access.

INACTIVE status A status that indicates entities deactivated from consolidation for the current period.

inactive user A user whose account was deactivated by an administrator.

income account An account storing periodic and year-to-date values that, if positive, increase net worth.

index (1) A method where Essbase uses sparse-data combinations to retrieve data in block storage databases. (2) The index file.

index cache A buffer containing index pages.

index entry A pointer to an intersection of sparse dimensions. Index entries point to data blocks on disk and use offsets to locate cells.

index file An Essbase file storing block storage data retrieval information, residing on disk, and containing index pages.

index page A subdivision in an index file. An index page contains pointers to data blocks.

input data Data loaded from a source rather than calculated.

installation assemblies Product installation files that plug in to EPM System Installer.

integration A process that is run to move data between Oracle's Hyperion applications using Shared Services. Data integration definitions specify the data moving between a source application and a destination application, and they enable the data movements to be grouped, ordered, and scheduled.

intelligent calculation A calculation method tracking updated data blocks since the last calculation.

Interactive Reporting connection file (.oce) Files encapsulating database connection information, including the database API (ODBC, SQL*Net, and so on), database software, the database server network address, and database user name. Administrators create and publish Interactive Reporting connection (.oce) files.

intercompany elimination See elimination.

intercompany matching The process of comparing balances for pairs of intercompany accounts within an application. Intercompany receivables are compared to intercompany payables for matches. Matching accounts are used to eliminate intercompany transactions from an organization's consolidated totals.

intercompany matching report A report that compares intercompany account balances and indicates whether the accounts are in balance.

interdimensional irrelevance A situation in which a dimension does not intersect with other dimensions. Because the data in the dimension cannot be accessed from the nonintersecting dimensions, the nonintersecting dimensions are not relevant to that dimension.

intersection A unit of data representing the intersection of dimensions in a multidimensional database; also, a worksheet cell.

intrastage assignment An assignment in the financial flow to an object within the same stage.

introspection A deep inspection of a data source to discover hierarchies based on the inherent relationships in the database. Contrast with scraping.

Investigation See drill-through.

isolation level An Essbase Kernel setting that determines the lock and commit behavior of database operations. Choices are: committed access and uncommitted access.

iteration A pass of the budget or planning cycle in which the same version of data is revised and promoted.

Java application server cluster An active-active application server cluster of Java Virtual Machines (JVMs).

Java Database Connectivity (JDBC) A client-server communication protocol used by Java-based clients and relational databases. The JDBC interface provides a call-level API for SQL-based database access.

job output Files or reports produced from running a job.

jobs Documents with special properties that can be launched to generate output. A job can contain Interactive Reporting, SQR Production Reporting, or generic documents.

join A link between two relational database tables or topics based on common content in a column or row. A join typically occurs between identical or similar items within different tables or topics. For example, a record in the Customer table is joined to a record in the Orders table because the Customer ID value is the same in each table.

journal entry (JE) A set of debit-credit adjustments to account balances for a scenario and period.

JSP Java Server Page.

KeyContacts gadget A gadget that contains a group of Smart Space users and provides access to Smart Space Collaborator. For example, you can have a KeyContacts gadget for your marketing team and another for your development team. See also gadget.

latest A spreadsheet keyword used to extract data values from the member defined as the latest time period.

layer (1) The horizontal location of members in a hierarchical structure, specified by generation (top down) or level (bottom up). (2) Position of objects relative to other objects. For example, in the Sample Basic database, Qtr1 and Qtr4 are in the same layer, so they are also in the same generation, but in a database with a ragged hierarchy, Qtr1 and Qtr4 might not be in same layer, though they are in the same generation.

layout area An area on a Workspace Page where content can be placed.

legend box A box containing labels that identify the data categories of a dimension.

level A layer in a hierarchical tree structure that defines database member relationships. Levels are ordered from the bottom dimension member (level 0) up to the parent members.

level 0 block A data block for combinations of sparse, level 0 members.

level 0 member A member that has no children.

liability account An account type that stores "point in time" balances of a company's liabilities. Examples: accrued expenses, accounts payable, and long-term debt.

lifecycle management The process of migrating an application, a repository, or individual artifacts across product environments.

line item detail The lowest level of detail in an account.

lineage The relationship between different metadata elements showing how one metadata element is derived from one or more other metadata elements, ultimately tracing the metadata element to its physical source. In Essbase Studio, a lineage viewer displays the relationships graphically. See also traceability.

link (1) A reference to a repository object. Links can reference folders, files, shortcuts, and other links. (2) In a taskflow, the point where the activity in one stage ends and another begins.

link condition A logical expression evaluated by the taskflow engine to determine the sequence of launching taskflow stages.

linked data model Documents that are linked to a master copy in a repository

linked partition A shared partition that enables you to use a data cell to link two databases. When a user clicks a linked cell in a worksheet, Essbase opens a new sheet displaying the dimensions in the linked database. The user can then drill down those dimensions.

linked reporting object (LRO) A cell-based link to an external file such as cell notes, URLs, or files with text, audio, video, or pictures. (Only cell notes are supported for Essbase LROs in Financial Reporting.) Contrast with local report object.

load balancer Hardware or software that directs the requests to individual application servers in a cluster and is the only point of entry into the system.

load balancing Distribution of requests across a group of servers, which helps to ensure optimal end user performance.

local currency An input currency type. When an input currency type is not specified, the local currency matches the entity's base currency.

local report object A report object that is not linked to a Financial Reporting report object in Explorer. Contrast with linked reporting object.

local results A data model's query results. Results can be used in local joins by dragging them into the data model. Local results are displayed in the catalog when requested.

locale A computer setting that specifies a location's language, currency and date formatting, data sort order, and the character set encoding used on the computer. Essbase uses only the encoding portion. See also encoding, ESSLANG.

locale header record A text record at the beginning of some non-Unicode-encoded text files, such as scripts, that identifies the encoding locale.

location alias A descriptor that identifies a data source. The location alias specifies a server, application, database, user name, and password. Location aliases are set by DBAs at the database level using Administration Services Console, ESSCMD, or the API.

locked A user-invoked process that prevents users and processes from modifying data.

locked data model A data model that cannot be modified by a user.

LOCKED status A consolidation status indicating that an entity contains data that cannot be modified.

Log Analyzer An Administration Services feature that enables filtering, searching, and analysis of Essbase logs.

logic group In FDM, one or more logic accounts generated after a source file is loaded into FDM. Logic accounts are calculated accounts derived from the source data.

logical Web application An aliased reference used to identify the internal host name, port, and context of a Web application. In a clustered or high-availability environment, this is the alias name that establishes a single internal reference for the distributed components. In EPM System, a nonclustered logical Web application defaults to the physical host running the Web application.

LRO See linked reporting object.

managed server An application server process running in its own Java Virtual Machine (JVM).

manual stage A stage that requires human intervention.

Map File A file that stores the definition for sending data to or retrieving data from an external database. Map files have different extensions (.mps to send data; .mpr to retrieve data).

Map Navigator A feature that displays your current position on a Strategy, Accountability, or Cause and Effect map, indicated by a red outline.

Marginal Tax Rate The rate used to calculate the after-tax cost of debt; represents the tax rate applied to the last earned income dollar (the rate from the highest tax bracket into which income falls) and includes federal, state, and local taxes. Based on current level of taxable income and tax bracket, you can predict marginal tax rate.

Market Risk Premium The additional rate of return paid over the risk-free rate to persuade investors to hold "riskier" investments than government securities. Calculated by subtracting the risk-free rate from the expected market return. These figures should closely model future market conditions.

master data model An independent data model that is referenced as a source by multiple queries. When used, "Locked Data Model" is displayed in the Query section's Content pane; the data model is linked to the master data model displayed in the Data Model section, which an administrator may hide.

mathematical operator A symbol that defines how data is calculated in formulas and outlines. Can be any of the standard mathematical or Boolean operators; for example, +, -, *, /, and %.

MaxL The multidimensional database access language for Essbase, consisting of a data definition language (MaxL DDL) and a data manipulation language (MaxL DML). See also MaxL DDL, MaxL DML, and MaxL Shell

MaxL DDL The data definition language used by Essbase for batch or interactive system-administration tasks.

MaxL DML The data manipulation language used in Essbase for data query and extraction.

MaxL Perl Module A Perl module (essbase.pm) that is part of Essbase MaxL DDL. This module can be added to the Perl package to provide access to Essbase databases from Perl programs.

MaxL Script Editor A script-development environment in Administration Services Console. MaxL Script Editor is an alternative to using a text editor and the MaxL Shell for administering Essbase with MaxL scripts.

MaxL Shell An interface for passing MaxL statements to Essbase Server. The MaxL Shell executable file is located in the Essbase bin directory (UNIX: essmsh; Windows: essmsh.exe).

MDX (multidimensional expression) A language used for querying and calculation in multidimensional-compliant databases.

measures Numeric values in an OLAP database cube that are available for analysis. Measures are margin, cost of goods sold, unit sales, budget amount, and so on. See also fact table.

member A discrete component within a dimension. A member identifies and differentiates the organization of similar units. For example, a time dimension might include members Jan, Feb, and Qtr1.

member list A named system- or user-defined group that references members, functions, or member lists within a dimension.

member load In Essbase Integration Services, the process of adding dimensions and members (without data) to Essbase outlines.

member selection report command A type of Report Writer command that selects member ranges based on outline relationships, such as sibling, generation, and level.

member-specific report command A type of Report Writer formatting command that is executed as it is encountered in a report script. The command affects only its associated member and executes the format command before processing the member.

merge A data load option that clears values only from the accounts specified in the data load file and replaces them with values in the data load file.

metadata A set of data that defines and describes the properties and attributes of the data stored in a database or used by an application. Examples of metadata are dimension names, member names, properties, time periods, and security.

metadata elements Metadata derived from data sources and other metadata that is stored and cataloged for Essbase Studio use.

metadata sampling The process of retrieving a sample of members in a dimension in a drill-down operation.

metadata security Security set at the member level to restrict users from accessing certain outline members.

metaoutline In Essbase Integration Services, a template containing the structure and rules for creating an Essbase outline from an OLAP model.

Middleware home A directory that includes the Oracle WebLogic Server home and can also include the EPM Oracle home and other Oracle homes. A Middleware home can reside on a local file system or on a remote shared disk that is accessible through NFS.

migration audit report A report generated from the migration log that provides tracking information for an application migration.

migration definition file (.mdf) A file that contains migration parameters for an application migration, enabling batch script processing.

migration log A log file that captures all application migration actions and messages.

migration snapshot A snapshot of an application migration that is captured in the migration log.

MIME Type An attribute that describes the data format of an item, so that the system knows which application should open the object. A file's MIME (Multipurpose Internet Mail Extension) type is determined by the file extension or HTTP header. Plug-ins tell browsers which MIME types they support and which file extensions correspond to each MIME type.

mining attribute In data mining, a class of values used as a factor in analysis of a set of data.

minireport A report component that includes layout, content, hyperlinks, and the query or queries to load the report. Each report can include one or more minireports.

minischema A graphical representation of a subset of tables from a data source that represents a data modeling context.

missing data (#MISSING) A marker indicating that data in the labeled location does not exist, contains no value, or was never entered or loaded. For example, missing data exists when an account contains data for a previous or future period but not for the current period.

model 1) In data mining, a collection of an algorithm's findings about examined data. A model can be applied against a wider data set to generate useful information about that data. 2) A file or content string containing an application-specific representation of data. Models are the basic data managed by Shared Services, of two major types: dimensional and nondimensional application objects. 3) In Business Modeling, a network of boxes connected to represent and calculate the operational and financial flow through the area being examined.

multidimensional database A method of organizing, storing, and referencing data through three or more dimensions. An individual value is the intersection point for a set of dimensions. Contrast with relational database.

Multiload An FDM feature that allows the simultaneous loading of multiple periods, categories, and locations.

My Workspace Page Customizable Workspace Pages created by users. They are marked specially so that they can be easily accessed from one single place without having to navigate the repository.

named set In MaxL DML, a set with its logic defined in the optional WITH section of a MaxL DML query. The named set can be referenced multiple times in the query.

native authentication The process of authenticating a user name and password from within the server or application.

nested column headings A report column heading format that displays data from multiple dimensions. For example, a column heading that contains Year and Scenario members is a nested column. The nested column heading shows Q1 (from the Year dimension) in the top line of the heading, qualified by Actual and Budget (from the Scenario dimension) in the bottom line of the heading.

NO DATA status A consolidation status indicating that this entity contains no data for the specified period and account.

non-dimensional model A Shared Services model type that includes application objects such as security files, member lists, calculation scripts, and Web forms.

non-unique member name See duplicate member name.

null value A value that is absent of data. Null values are not equal to zero.

numeric attribute range A feature used to associate a base dimension member that has a discrete numeric value with an attribute that represents a value range. For example, to classify customers by age, an Age Group attribute dimension can contain members for the following age ranges: 0-20, 21-40, 41-60, and 61-80. Each Customer dimension member can be associated with an Age Group range. Data can be retrieved based on the age ranges rather than on individual age values.

ODBC Open Database Connectivity. A database access method used from any application regardless of how the database management system (DBMS) processes the information.

OK status A consolidation status indicating that an entity has already been consolidated, and that data has not changed below it in the organization structure.

OLAP Metadata Catalog In Essbase Integration Services, a relational database containing metadata describing the nature, source, location, and type of data that is pulled from the relational data source.

OLAP model In Essbase Integration Services, a logical model (star schema) that is created from tables and columns in a relational database. The OLAP model is then used to generate the structure of a multidimensional database. See also online analytical processing (OLAP).

online analytical processing (OLAP) A multidimensional, multiuser, client-server computing environment for users who analyze consolidated enterprise data in real time. OLAP systems feature drill-down, data pivoting, complex calculations, trend analysis, and modeling.

Open Database Connectivity (ODBC) Standardized application programming interface (API) technology that allows applications to access multiple third-party databases.

Oracle home A directory containing the installed files required by a specific product, and residing within the directory structure of Middleware home. See also Middleware home.

organization An entity hierarchy that defines each entity and their relationship to others in the hierarchy.

origin The intersection of two axes.

outline The database structure of a multidimensional database, including all dimensions, members, tags, types, consolidations, and mathematical relationships. Data is stored in the database according to the structure defined in the outline.

outline synchronization For partitioned databases, the process of propagating outline changes from one database to another database.

P&L accounts (P&L) Profit and loss accounts. P&L refers to a typical grouping of expense and income accounts that comprise a company's income statement.

page A display of information in a grid or table often represented by the Z-axis. A page can contain data from one field, derived data from a calculation, or text.

page file An Essbase data file.

page heading A report heading type that lists members represented on the current page of the report. All data values on the page have the members in the page heading as a common attribute.

page member A member that determines the page axis.

palette A JASC-compliant file with a .PAL extension. Each palette contains 16 colors that complement each other and can be used to set the dashboard color elements.

parallel calculation A calculation option. Essbase divides a calculation into tasks and calculates some tasks simultaneously.

parallel data load In Essbase, the concurrent execution of data load stages by multiple process threads.

parallel export The ability to export Essbase data to multiple files. This may be faster than exporting to a single file, and it may resolve problems caused by a single data file becoming too large for the operating system to handle.

parent adjustments The journal entries that are posted to a child in relation to its parent.

parents The entities that contain one or more dependent entities that report directly to them. Because parents are entities associated with at least one node, they have entity, node, and parent information associated with them.

partition area A subcube within a database. A partition is composed of one or more areas of cells from a portion of the database. For replicated and transparent partitions, the number of cells within an area must be the same for the data source and target to ensure that the two partitions have the same shape. If the data source area contains 18 cells, the data target area must also contain 18 cells to accommodate the number of values.

partitioning The process of defining areas of data that are shared or linked between data models. Partitioning can affect the performance and scalability of Essbase applications.

pattern matching The ability to match a value with any or all characters of an item entered as a criterion. Missing characters may be represented by wild-card values such as a question mark (?) or an asterisk (*). For example, "Find all instances of apple" returns apple, but "Find all instances of apple*" returns apple, applesauce, applecranberry, and so on.

percent consolidation The portion of a child's values that is consolidated to its parent.

percent control The extent to which an entity is controlled within the context of its group.

percent ownership The extent to which an entity is owned by its parent.

performance indicator An image file used to represent measure and scorecard performance based on a range you specify; also called a status symbol. You can use the default performance indicators or create an unlimited number of your own.

periodic value method (PVA) A process of currency conversion that applies the periodic exchange rate values over time to derive converted results.

permission A level of access granted to users and groups for managing data or other users and groups.

persistence The continuance or longevity of effect for any Essbase operation or setting. For example, an Essbase administrator may limit the persistence of user name and password validity.

personal pages A personal window to repository information. You select what information to display and its layout and colors.

personal recurring time events Reusable time events that are accessible only to the user who created them.

personal variable A named selection statement of complex member selections.

perspective A category used to group measures on a scorecard or strategic objectives within an application. A perspective can represent a key stakeholder (such as a customer, employee, or shareholder/financial) or a key competency area (such as time, cost, or quality).

pinboard One of the three data object display types. Pinboards are graphics composed of backgrounds and interactive icons called pins. Pinboards require traffic lighting definitions.

pins Interactive icons placed on graphic reports called pinboards. Pins are dynamic. They can change images and traffic lighting color based on the underlying data values and analysis tools criteria.

pivot Alter the perspective of retrieved data. When Essbase first retrieves a dimension, it expands data into rows. You can then pivot or rearrange the data to obtain a different viewpoint.

planner A user who can input and submit data, use reports that others create, execute business rules, use task lists, enable e-mail notification for themselves, and use Smart View. Planners comprise the majority of users.

planning unit A data slice at the intersection of a scenario, version, and entity; the basic unit for preparing, reviewing, annotating, and approving plan data.

plot area The area bounded by X, Y, and Z axes; for pie charts, the rectangular area surrounding the pie.

plug account An account in which the system stores any out-of-balance differences between intercompany account pairs during the elimination process.

post stage assignment Assignments in the allocation model that are assigned to locations in a subsequent model stage.

POV (point of view) A feature for setting data focus by selecting members that are not already assigned to row, column, or page axes. For example, selectable POVs in FDM could include location, period, category, and target category. In another example, using POV as a filter in Smart View, you could assign the Currency dimension to the POV and select the Euro member. Selecting this POV in data forms displays data in Euro values.

precalculation Calculating the database before user retrieval.

precision Number of decimal places displayed in numbers.

predefined drill paths Paths used to drill to the next level of detail, as defined in the data model.

presentation A playlist of Web Analysis documents, enabling reports to be grouped, organized, ordered, distributed, and reviewed. Includes pointers referencing reports in the repository.

preserve formulas User-created formulas kept within a worksheet while retrieving data.

primary measure A high-priority measure important to your company and business needs. Displayed in the Contents frame.

Process Monitor Report A list of locations and their positions within the FDM data conversion process. You can use the process monitor report to monitor the status of the closing process. The report is time-stamped. Therefore, it can be used to determine to which locations at which time data was loaded.

product In Shared Services, an application type, such as Planning or Performance Scorecard.

Production Reporting See SQR Production Reporting.

project An instance of Oracle's Hyperion products grouped together in an implementation. For example, a Planning project may consist of a Planning application, an Essbase cube, and a Financial Reporting Server instance.

provisioning The process of granting users and groups specific access permissions to resources.

proxy server A server acting as an intermediary between workstation users and the Internet to ensure security.

public job parameters Reusable named job parameters created by administrators and accessible to users with requisite access privileges.

public recurring time events Reusable time events created by administrators and accessible through the access control system.

PVA See periodic value method.

qualified name A member name in a qualified format that differentiates duplicate member names in a duplicate member outline. For example, [Market].[East].[State]. [New York] or [Market].[East].[City].[New York].

query governor An Essbase Integration Server parameter or Essbase Server configuration setting that controls the duration and size of queries made to data sources.

reciprocal assignment An assignment in the financial flow that also has the source as one of its destinations.

reconfigure URL A URL that is used to reload servlet configuration settings dynamically when users are already logged on to the Workspace.

record In a database, a group of fields making up one complete entry. For example, a customer record may contain fields for name, address, telephone number, and sales data.

recurring template A journal template for making identical adjustments in every period.

recurring time event An event specifying a starting point and the frequency for running a job.

redundant data Duplicate data blocks that Essbase retains during transactions until Essbase commits updated blocks.

regular journal A feature for entering one-time adjustments for a period. A regular journal can be balanced, balanced by entity, or unbalanced.

Related Accounts Accounts related to the main account and grouped under the same main account number. The account structure groups all main and related accounts under the same main account number. The main account is distinguished from related accounts by the first suffix of the account number.

relational database A type of database that stores data in related two-dimensional tables. Contrast with multidimensional database.

replace A data load option that clears existing values from all accounts for periods specified in the data load file and loads values from the data load file. If an account is not specified in the load file, its values for the specified periods are cleared.

replicated partition A portion of a database, defined through Partition Manager, used to propagate an update to data mastered at one site to a copy of data stored at another site. Users can access the data as though it were part of their local database.

Report Extractor An Essbase component that retrieves report data from the Essbase database when report scripts are run.

report object In report designs, a basic element with properties defining behavior or appearance, such as text boxes, grids, images, and charts.

report script A text file containing Essbase Report Writer commands that generate one or more production reports.

Report Viewer An Essbase component that displays complete reports after report scripts are run.

reporting currency The currency used to prepare financial statements, and converted from local currencies to reporting currencies.

repository Storage location for metadata, formatting, and annotation information for views and queries.

resources Objects or services managed by the system, such as roles, users, groups, files, and jobs.

restore An operation to reload data and structural information after a database has been damaged or destroyed, typically performed after shutting down and restarting the database.

restructure An operation to regenerate or rebuild the database index and, in some cases, data files.

result frequency The algorithm used to createa set of dates to collect and display results.

review level A Process Management review status indicator representing the process unit level, such as Not Started, First Pass, Submitted, Approved, and Published.

Risk Free Rate The rate of return expected from "safer" investments such as long-term U.S. government securities.

role The means by which access permissions are granted to users and groups for resources.

roll-up See consolidation.

root member The highest member in a dimension branch.

runtime prompt A variable that users enter or select before a business rule is run.

sampling The process of selecting a representative portion of an entity to determine the entity's characteristics. See also metadata sampling.

saved assumptions User-defined Planning assumptions that drive key business calculations (for example, the cost per square foot of office floor space).

scaling Scaling determines the display of values in whole numbers, tens, hundreds, thousands, millions, and so on.

scenario A dimension for classifying data; for example, Actuals, Budget, Forecast1, or Forecast2.

schema In relational databases, a logical model that represents the data and the relationships between the data.

scope The area of data encompassed by any Essbase operation or setting; for example, the area of data affected by a security setting. Most commonly, scope refers to three levels of granularity, where higher levels encompass lower levels. The levels, from highest to lowest: the entire system (Essbase Server), applications on Essbase Server, or databases within Essbase Server applications. See also persistence.

score The level at which targets are achieved, usually expressed as a percentage of the target.

scorecard A business object that represents the progress of an employee, strategy element, or accountability element toward goals. Scorecards ascertain this progress based on data collected for each measure and child scorecard added to the scorecard.

scraping An inspection of a data source to derive the most basic metadata elements from it. Contrast with introspection.

secondary measure A low-priority measure, less important than primary measures. Secondary measures do not have Performance reports but can be used on scorecards and to create dimension measure templates.

security agent A Web access management provider (for example, Oracle Access Manager, Oracle Single Sign-On, or CA SiteMinder) that protects corporate Web resources.

security platform A framework enabling Oracle EPM System products to use external authentication and single sign-on.

serial calculation The default calculation setting. Divides a calculation pass into tasks and calculates one task at a time.

services Resources that enable business items to be retrieved, changed, added, or deleted. Examples: Authorization and Authentication.

servlet A piece of compiled code executable by a Web server.

shared disks See shared storage.

shared member A member that shares storage space with another member of the same name, preventing duplicate calculation of members that occur multiple times in an Essbase outline.

Shared Services Registry The part of the Shared Services repository that manages EPM System deployment information for most EPM System products, including installation directories, database settings, computer names, ports, servers, URLs, and dependent service data.

shared storage A set of disks containing data that must be available to all nodes of a failover cluster; also called shared disks.

Shared Workspace Pages Workspace Pages shared across an organization that are stored in a special System folder and can be accessed by authorized users from the Shared Workspace Pages Navigate menu.

sibling A child member at the same generation as another child member and having the same immediate parent. For example, the members Florida and New York are children of East and each other's siblings.

silent response files Files providing data that an installation administrator would otherwise be required to provide. Response files enable EPM System Installer or EPM System Configurator to run without user intervention or input.

single point of failure Any component in a system that, if it fails, prevents users from accessing the normal functionality.

single sign-on (SSO) The ability to log on once and then access multiple applications without being prompted again for authentication.

smart tags Keywords in Microsoft Office applications that are associated with predefined actions available from the Smart Tag menu. In Oracle EPM System products, smart tags can also be used to import Reporting and Analysis content and to access Financial Management and Essbase functions.

SmartCut A link to a repository item, in URL form.

snapshot Read-only data from a specific time.

source currency The currency from which values originate and are converted through exchange rates to the destination currency.

sparse dimension In block storage databases, a dimension unlikely to contain data for all member combinations when compared to other dimensions. Contrast with dense dimension. For example, not all customers have data for all products.

SPF files Printer-independent files created by an SQR Production Reporting server, containing a representation of the actual formatted report output, including fonts, spacing, headers, footers, and so on.

Spotlighter A tool that enables color coding based on selected conditions.

SQL spreadsheet A data object that displays the result set of a SQL query.

SQR Production Reporting A specialized programming language for data access, data manipulation, and creating SQR Production Reporting documents.

stage 1) A task description that forms one logical step within a taskflow, usually performed by an individual. A stage can be manual or automated. 2) For Profitability, logical divisions within the model that represent the steps in the allocation process within your organization.

stage action For automated stages, the invoked action that executes the stage.

staging area A database that you create to meet the needs of a specific application. A staging area is a snapshot or restructured version of one or more RDBMS.

staging table A database that you create to meet the needs of a specific application. A staging area is a snapshot or restructured version of one or more RDBMSs.

standard dimension A dimension that is not an attribute dimension.

standard journal template A journal function used to post adjustments that have common adjustment information for each period. For example, you can create a standard template that contains the common account IDs, entity IDs, or amounts, and then use the template as the basis for many regular journals.

Status bar The bar at the bottom of the screen that displays helpful information about commands, accounts, and the current status of your data file.

stored hierarchy In aggregate storage databases outlines only, a hierarchy in which the members are aggregated according to the outline structure. Stored hierarchy members have certain restrictions; for example, they cannot contain formulas.

strategic objective (SO) A long-term goal defined by measurable results. Each strategic objective is associated with one perspective in the application, has one parent, the entity, and is a parent to critical success factors or other strategic objectives.

Strategy map Represents how the organization implements high-level mission and vision statements into lower-level, constituent strategic goals and objectives.

structure view Displays a topic as a simple list of component data items.

Structured Query Language A language used to process instructions to relational databases.

Subaccount Numbering A system for numbering subaccounts using nonsequential whole numbers.

subscribe Flags an item or folder to receive automatic notification whenever the item or folder is updated.

Summary chart In the Investigates Section, a chart that rolls up detail charts shown below in the same column, plotting metrics at the summary level at the top of each chart column.

supervisor A user with full access to all applications, databases, related files, and security mechanisms for a server.

supporting detail Calculations and assumptions from which the values of cells are derived.

suppress rows A setting that excludes rows containing missing values and underscores characters from spreadsheet reports.

symmetric multiprocessing (SMP) A server architecture that enables multiprocessing and multithreading. Performance is not significantly degraded when a large number of users simultaneously connect to an single instance.

sync Synchronization of Shared Services and application models.

synchronized The condition that exists when the latest version of a model resides in both the application and in Shared Services. See also model.

system extract A feature that transfers data from application metadata into an ASCII file.

tabs Navigable views of accounts and reports in Strategic Finance.

target Expected results of a measure for a specified period of time (day, quarter, and so on).

task list A detailed status list of tasks for a particular user.

taskflow The automation of a business process in which tasks are passed from one taskflow participant to another according to procedural rules.

taskflow definition Business processes in the taskflow management system that consist of a network of stages and their relationships; criteria indicating the start and end of the taskflow; and information about individual stages, such as participants, associated applications, associated activities, and so on.

taskflow instance A single instance of a taskflow including its state and associated data.

taskflow management system A system that defines, creates, and manages the execution of a taskflow, including definitions, user or application interactions, and application executables.

taskflow participant The resource that performs the task associated with the taskflow stage instance for both manual and automated stages.

Taxes - Initial Balances Strategic Finance assumes that the Initial Loss Balance, Initial Gain Balance, and Initial Balance of Taxes Paid entries have taken place in the period before the first Strategic Finance time period.

TCP/IP See Transmission Control Protocol/Internet Protocol.

text measure In Essbase, a member tagged as Text in the dimension where measures are represented. The cell values are displayed as predefined text. For example, the text measure Satisfaction Index may have the values Low, Medium, and High. See also typed measure, text list, derived text measure.

time dimension The time period that the data represents, such as fiscal or calendar periods.

time events Triggers for job execution.

time scale A scale that displays metrics by a specific time span, such as monthly or quarterly.

time series reporting A process for reporting data based on a calendar date (for example, year, quarter, month, or week).

Timeline Viewer An FDM feature that enables users to view dates and times of completed process flow steps for specific locations.

Title bar A bar that displays the Strategic Finance name, the file name, and the scenario name Version box.

toast message A message that fades in the lower-right corner of the screen.

token An encrypted identification of one valid user or group on an external authentication system.

top and side labels Column and row headings on the top and sides of a Pivot report.

top-level member A dimension member at the top of the tree in a dimension outline hierarchy, or the first member of the dimension in sort order if there is no hierarchical relationship among dimension members. If a hierarchical relationship exists, the top-level member name is generally the same as the dimension name.

trace allocations A Profitability feature that enables you to visually follow the flow of financial data, either forwards or backwards, from a single intersection throughout the model.

trace level The level of detail captured in a log file.

traceability The ability to track a metadata element to its physical source. For example, in Essbase Studio, a cube schema can be traced from its hierarchies and measure hierarchies to its dimension elements, date/time elements, measures, and, ultimately, to its physical source elements. See also lineage.

traffic lighting Color-coding of report cells, or pins based on a comparison of two dimension members, or on fixed limits.

transformation 1) A process that transforms artifacts so that they function properly in the destination environment after application migration. 2) In data mining, the modification of data (bidirectionally) flowing between the cells in the cube and the algorithm.

translation See currency conversion.

Transmission Control Protocol/Internet Protocol (TCP/IP) A standard set of communication protocols linking computers with different operating systems and internal architectures. TCP/IP utilities are used to exchange files, send mail, and store data to various computers that are connected to local and wide area networks.

transparent login A process that logs in authenticated users without launching the login screen.

transparent partition A shared partition that enables users to access and change data in a remote database as though it is part of a local database.

triangulation A means of converting balances from one currency to another through a third common currency. In Europe, this currency is the euro for member countries. For example, to convert from the French franc to the Italian lira, the common currency is defined as the European euro. Therefore, to convert balances from the French franc to the Italian lira, balances are converted from the French franc to the European euro and from the European euro to Italian lira.

triggers An Essbase feature whereby data is monitored according to user-specified criteria that, when met, cause Essbase to alert the user or system administrator.

trusted user Authenticated user.

tuple MDX syntax element that references a cell as an intersection of a member from each dimension. If a dimension is omitted, its top member is implied. Examples: (Jan); (Jan, Sales); ([Jan], [Sales], [Cola], [Texas], [Actual]).

two-pass An Essbase property that is used to recalculate members that are dependent on the calculated values of other members. Two-pass members are calculated during a second pass through the outline.

unary operator A mathematical indicator (+, -, *, /, %) associated with an outline member. The unary operator defines how the member is calculated during a database roll-up.

Unicode-mode application An Essbase application wherein character text is encoded in UTF-8, enabling users with computers set up for different languages to share application data.

unique member name A nonshared member name that exists only once in a database outline.

unique member outline A database outline that is not enabled for duplicate member names.

upgrade The process of replacing a software release with a newer release. The term upgrade does not apply to installing a maintenance release. See also maintenance release, migration.

upper-level block A type of data block wherein at least one of the sparse members is a parent-level member.

user directory A centralized location for user and group information, also known as a repository or provider. Popular user directories include Oracle Internet Directory (OID), Microsoft Active Directory (MSAD), and Sun Java System Directory Server.

user variable A variable that ynamically renders data forms based on a user's member selection, displaying only the specified entity. For example, a user variable named Department displays specific departments and employees.

user-defined attribute (UDA) An attribute, associated with members of an outline to describe a characteristic of the members, that can be used to return lists of members that have the specified associated UDA.

user-defined member list A named, static set of members within a dimension defined by the user.

validation The process of checking a business rule, report script, or partition definition against the outline to ensure that the object being checked is valid.

validation rules Rules used in FDM to enforce data integrity. For example, in FDM, validation rules ensure that certain conditions are met after data is loaded from FDM to the target application.

value dimension A dimension that is used to define input value, translated value, and consolidation detail.

variance The difference between two values (for example, between planned and actual values).

version A possible outcome used within the context of a scenario of data. For example, Budget - Best Case and Budget - Worst Case where Budget is scenario and Best Case and Worst Case are versions.

vertical application server cluster A cluster with multiple application server instances on the same machine.

view A year-to-date or periodic display of data.

visual cue A formatted style, such as a font or a color, that highlights specific data value types. Data values may be dimension members; parent, child, or shared members; dynamic calculations; members containing a formula; read-only data cells; read-and-write data cells; or linked objects.

WebLogic Server home A subdirectory of Middleware home containing installed files required by a WebLogic Server instance. WebLogic Server home is a peer of Oracle homes.

weight A value assigned to an item on a scorecard that indicates the relative importance of that item in the calculation of the overall scorecard score. The weighting of all items on a scorecard accumulates to 100%. For example, to recognize the importance of developing new features for a product, the measure for New Features Coded on a developer's scorecard would be assigned a higher weighting than a measure for Number of Minor Defect Fixes.

wild card Character that represents any single character (?) or group of characters (*) in a search string.

WITH section In MaxL DML, an optional section of the query used for creating reusable logic to define sets or members. Sets or custom members can be defined once in the WITH section and then referenced multiple times during a query.

workbook An entire spreadsheet file with many worksheets.

workflow The steps required to process data from start to finish in FDM. The workflow consists of Import (loading data from the GL file), Validate (ensures that all members are mapped to a valid account), Export (loads the mapped members to the target application), and Check (verifies accuracy of data by processing data with user-defined validation rules).

Workspace Page A page created with content from multiple sources including documents, URL, and other content types. Enables a user to aggregate content from Oracle and non-Oracle sources.

write-back The ability for a retrieval client, such as a spreadsheet, to update a database value.

ws.conf A configuration file for Windows platforms.

wsconf_platform A configuration file for UNIX platforms.

XML   See Extensible Markup Language.

XOLAP   An Essbase multidimensional database that stores only the outline metadata and retrieves all data from a relational database at query time. XOLAP supports aggregate storage databases and applications that contain duplicate member names.

Y axis scale   A range of values on Y axis of charts displayed in Investigate Section. For example, use a unique Y axis scale for each chart, the same Y axis scale for all Detail charts, or the same Y axis scale for all charts in the column. Often, using a common Y axis improves your ability to compare charts at a glance.

Zero Administration   A software tool that identifies version number of the most up-to-date plug-in on the server.

ZoomChart   A tool for viewing detailed information by enlarging a chart. A ZoomChart enables you to see detailed numeric information on the metric that is displayed in the chart.

Index

S