Getting Started with Brio Intelligence 6.6

Query and Results

BRIO SOFTWARE™

4980 Great America Parkway
Santa Clara, CA 95054 USA
+1(408)496-7400

Getting Started with Brio Intelligence 6.6 — *Query and Results*

Part Number 1209863

Brio Software
4980 Great America Parkway
Santa Clara, CA 95054
+1(408)496-7400

support@Brio.com
sales@Brio.com
www.Brio.com

Refer to the Brio Software License Agreement in this package before installing or using the product.

If you find any errors or problems with this documentation, please notify Brio Software. Brio Software does not guarantee that this document is without error. The information in this document is subject to change without notice.

Trademarks

Contents in Brief

Contents

About This Book

CHAPTER 1 Introducing Brio Intelligence

CHAPTER 2 Getting Started with Brio Intelligence: A Tutorial

CHAPTER 5 Working with Query Results

CHAPTER 6 Querying Multidimensional Databases

CHAPTER 7 Using Limits

CHAPTER 8 Working with Computed Items

CHAPTER 9 Applying Sorts

Glossary

Index

About This Book

Welcome to *Getting Started with Brio Intelligence 6.6*. This book is designed to help you learn the Brio Intelligence application, part of an integrated suite of powerful and easy-to-use business intelligence tools for query, OLAP analysis, and analytical reporting across the extended enterprise.

Audience

Getting Started with Brio Intelligence 6.6 is written for all levels of Brio Intelligence users, from those who need to simply retrieve and view data in a report format, to those who need to build queries and reports as well as analyze data.

In This Book

Getting Started with Brio Intelligence 6.6 – Query and Results, one of four books that explain how to use Brio Intelligence (see “Related Documents” on page -xiv), provides an overview of Brio Intelligence and explains the Brio Intelligence user interface and basic commands. It includes how to retrieve data using Brio Intelligence, how to query new data and change existing queries, and how to query a single database as well as multiple databases. It also covers how to work with query results.

- Chapter 1, “Introducing Brio Intelligence,” introduces business intelligence software tools and provides an overview of how anyone can use Brio Intelligence to access and analyze database information.
- Chapter 2, “Getting Started with Brio Intelligence: A Tutorial,” offers a tutorial on basic data analysis techniques and familiarizes you with the powerful capabilities and features of the Brio Intelligence application.

- Chapter 3, “Brio Intelligence Basics,” provides an overview of the Brio Intelligence workspace and describes fundamental Brio Intelligence features and functions.
- Chapter 4, “Querying Relational Databases,” explains how to use Brio Intelligence to connect to and query a relational database. It provides basic information about data models and the Brio Intelligence Repository, as well as how to build and process queries and subqueries.
- Chapter 5, “Working with Query Results,” explains how to work with the results sets obtained from your relational database query or data import. It includes how to enhance your results set, as well as how to export your results to different file formats.
- Chapter 6, “Querying Multidimensional Databases,” explains how to use Brio Intelligence to connect to and query a multidimensional database. It details how to build, refine, and process OLAP queries and how to apply filters and limits.
- Chapter 7, “Using Limits,” explains how to use limits to refine your data and filter away the data you do not need for more intelligent analysis.
- Chapter 8, “Working with Computed Items,” describes how to use calculations to compute new data items. Such calculations are important for supplementing the information already stored in the database
- Chapter 9, “Applying Sorts,” discusses features that enable you to sort data in various Brio Intelligence sections, including sort lines, single sorts, and nested sorts.

In addition, a glossary and index provide definitions and easy access to information contained in the book.

Typographic Conventions

This book uses the following type conventions:

- Options, buttons, or tabs that you need to choose and text that you need to type are indicated in bold.

  Select **Typical Install**. Type 1234.

- Key names are shown in square brackets.

  Press [Down Arrow]

- Two key names joined with a plus sign (+) are consecutive keystrokes. Press and hold down the first key while pressing the second key.

  Press **[Ctrl+Z]**.

- Options in a menu command path are separated with an arrow. The example indicates that you are to open the File menu and choose the Open menu item.

  Choose **File**→**Open**. **[Ctrl+O]**

Note When an instruction includes a menu command, the toolbar icon (if one exists) for the command appears in the left margin. The keyboard shortcut (if one exists) for the command is listed in brackets at the end of the line.

- Variables you replace with specific information are shown in *italics*.

  sp_adduser *login_id*

- Files, directories, and paths are shown in a `monospace` font.

  `Sample1.bqy` is located in the `BrioQuery/Samples directory.`

- A Note, Tip, or Caution is a brief side-note that deserves special attention or does not fit within the normal flow of text. These types of information are set off in the text by an icon in the margin.

☆ **Tip** This is an example tip.

Caution! This is an example caution.

Related Documents

Along with the *Getting Started with Brio Intelligence 6.6* book, there are three additional Brio Intelligence books:

- *Data Analysis and Reporting with Brio Intelligence 6.6* describes how to use the Brio Intelligence application's powerful reporting features—pivots, charts, and tables—and the Report Designer to create spectacular reports. It also explains common report features such as printing, drill-down, and spotlighter.
- *Brio Intelligence 6.6 Administrator's Guide* explains data modeling, including how to modify existing data models, and create new data models. It also discusses metadata definitions, database connectivity, and document scheduling.
- *Brio Intelligence Object Model and Executive Information Systems* explains the Brio Intelligence Object Model and how to create custom EIS applications using JavaScript.

Help

Brio Intelligence comes with a number of user manuals as well as an extensive online help system. If you need help with Brio Intelligence and cannot find the answers you need in the documentation, and you have a current Brio Technical Support agreement, call Brio Technical Support at +1(800)337-6324 (within North America) or +1(619)610-5769. You may also send an email message to *support@brio.com*.

Please be prepared to provide your valid customer number and compan name. You also need to know the version of Brio Intelligence you are using.

1 Introducing Brio Intelligence

Welcome to *Getting Started with Brio Intelligence 6.6*. This book shows how anyone —both technical and nontechnical people—can use Brio Intelligence to access and analyze database information.

This chapter introduces the conceptual background of business intelligence software tools and provides an overview of database concepts. It also explains the Brio Intelligence approach to querying databases. This chapter contains:

- Brio Intelligence Overview
- Brio Intelligence Approach to Business Intelligence
- How Brio Intelligence Works
- About Brio Intelligence Documents
- Turning Data into Information

Brio Intelligence Overview

Brio Intelligence 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. Brio Intelligence provides these capabilities:

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

The Brio Intelligence client 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.

Brio Intelligence Features

Brio Intelligence features include:

- Support for all industry-standard databases, including AS/400, Informix, Oracle, Sybase, Microsoft SQL Server, SAP, Red Brick Warehouse, Broadbase, and Essbase.
- A point-and-click interface for intuitive custom query and report building.
- Support for Microsoft Windows, the MacOS, and Motif (UNIX), with complete file compatibility between the platforms.
- Easy, nonprocedural navigation between query and reporting sections.
- Client editions that maintain a consistent interface for the client and administrative editions of Brio Intelligence, but that provide different levels of access to Brio Intelligence features.
- An extensive online help system which provides assistance for features and document construction.
- A drag-and-drop Outliner 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.

The Brio Intelligence User Experience

Although there are many types of Brio Intelligence users, the integrated interface works the same way whether using a Brio Intelligence client/server-based tool or a Web-based tool. This means that users of any level have only one tool to learn and use for all of their business intelligence needs.

Brio Intelligence comprises three client/server-based tools and three Web-based tools. Table 1-1 lists these tools along with their core query, analysis, and reporting features and shows the increasing levels of data modeling functionality.

Table 1-1 Features of Brio Intelligence Web-based and Client/Server-based Tools

Tool Type	Product Name	View	Process	Analyze	Create Queries	Create Reports	Create Data Models	Access Repository	Administer Repository	Access Database Tables	Register to ODS	EIS Scripting
Web	Freeview	✓										
	Quickview	✓	✓					✓				
	Insight	✓	✓	✓	✓	✓	✓*	✓		✓*		
Client/Server	Navigator	✓	✓	✓	✓	✓		✓				
	Explorer	✓	✓	✓	✓	✓	✓	✓		✓		
	Designer	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

* Create data models based on prebuilt document or data model.

The Brio Intelligence tools are:

- Brio Intelligence Navigator – A client/server-based tool that provides simplified query, analysis, and reporting for analysts who need more structured access to the data stored in the repository, but need full analysis and reporting functionality. Navigator users can use data models to create their own queries and use all the Brio Intelligence reporting and analysis features to work with information.
- Brio Intelligence Explorer – A client/server-based tool that delivers query, analysis, and reporting capabilities for power users who need to directly access data sources—or to explore the information organized in pre-built data models stored in the repository. Explorer users can use distributed predefined data models or create new data models from database tables for their own or distributed use.
- Brio Intelligence Designer – A client/server-based tool that delivers query, analysis, and reporting capabilities and centralized solution administration for developers, database administrators, and system administrators. Designer users manage the Brio Intelligence environment by building data, which they distribute to other Brio Intelligence users. They also create and administer job repositories, and build custom, Web-enabled dashboards (also called EIS or Executive Information System) using the Brio Intelligence open application development environment.
- Freeview – A read-only Web plug-in that allows users the ability to view Brio Intelligence reports.
- Quickview – A Web plug-in that offers simplified report viewing and data refresh for users who need to view published, formatted reports within their browser—a perfect way to publish analysis results to the masses for up-to-the-minute communication.
- Insight – A Web-based tool that delivers query, analysis, and reporting functionality for intranet, Internet, or extranet access to information. Based on user profiles or report-level security, the client environment adapts in six stages from full query, analysis, and reporting with data refresh to static report viewing. Insight users can use distributed predefined data models to create their own queries or to create new data models. They can use all the Brio Intelligence reporting and analysis features to analyze the data from their own queries and work with resulting datasets.

Brio Intelligence Approach to Business Intelligence

Brio Intelligence is easy to administer across the enterprise because its structure takes into account the different ways that people work with databases and their information needs, the user's technical experience with client/server database technology, and the way Brio Intelligence is deployed in the computing environment.

Figure 1-1 shows a Brio Intelligence deployment that uses both the client/server-based tools and the Web-based tools.

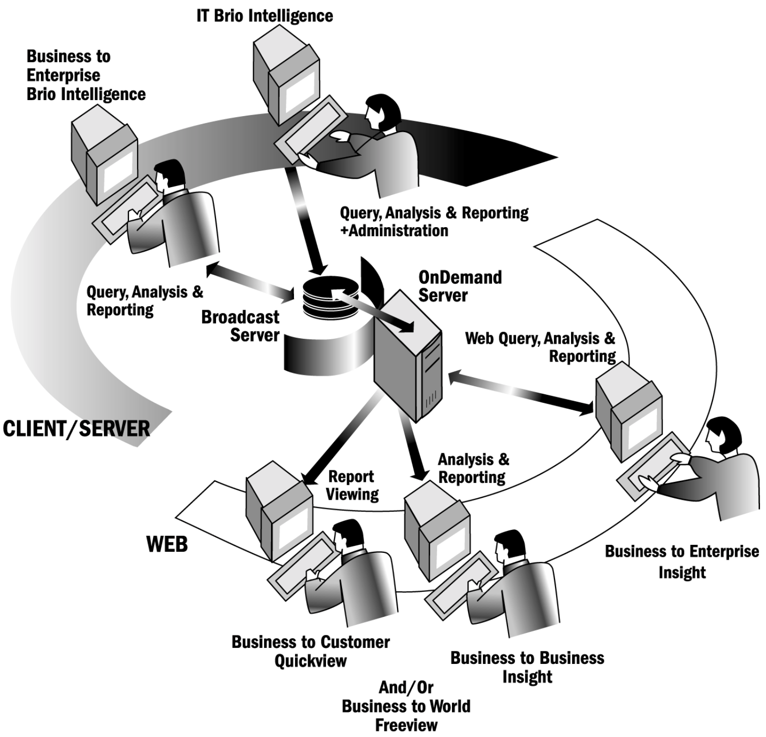
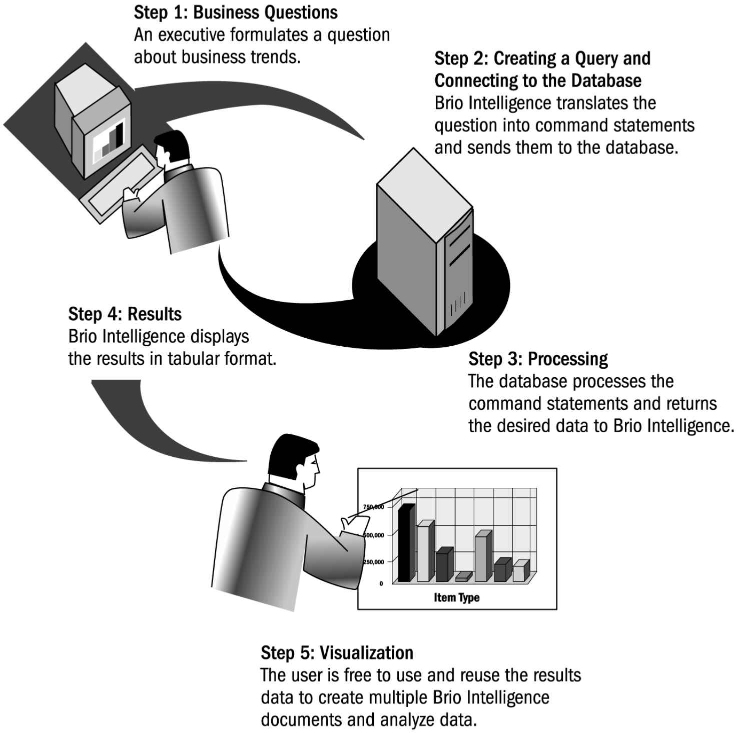

IT Brio Intelligence
Business to Enterprise Brio Intelligence
Query, Analysis & Reporting +Administration
Query, Analysis & Reporting
Broadcast Server
OnDemand Server
Web Query, Analysis & Reporting
CLIENT/SERVER
Report Viewing
Analysis & Reporting
WEB
Business to Enterprise Insight
Business to Customer Quickview
Business to Business Insight
And/Or Business to World Freeview

Figure 1-1 Brio Intelligence Deployment with Both Client/Server-based and Web-based Tools

Advanced data modeling, document control, and JavaScripting are included in the same application. A centralized repository stores and automatically updates data models to reflect changes in the database. Brio Intelligence serves as the source of a complete business intelligence delivery system. Working in tandem with Brio Broadcast Server, Brio Intelligence can schedule queries and deliver them to users for viewing across the intranet through Web browsers.

How Brio Intelligence Works

Brio Intelligence 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. Brio Intelligence automatically builds a query to your specifications and sends it to the database, and then displays the retrieved data as a table of results. Figure 1-2 illustrates this process.



Step 1: Business Questions
An executive formulates a question about business trends.

Step 2: Creating a Query and Connecting to the Database
Brio Intelligence translates the question into command statements and sends them to the database.

Step 3: Processing
The database processes the command statements and returns the desired data to Brio Intelligence.

Step 4: Results
Brio Intelligence displays the results in tabular format.

Step 5: Visualization
The user is free to use and reuse the results data to create multiple Brio Intelligence documents and analyze data.

Figure 1-2 Brio Intelligence Process

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."

Brio Intelligence supports the following relational database systems:

- AS/400
- Broadbase
- DB2
- Informix
- Redbrick
- SQL Server
- Sybase and Sybase IQ
- Teradata

Multidimensional Databases

A *multidimensional database* is a data *cube* that provides multidimensional views of business data. Multidimensional databases are OLAP servers that enable a user 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.

Brio Intelligence supports the following multidimensional database systems:

- DB2 OLAP
- Essbase
- MetaCube
- SAP BW
- SQL Server 7.0 for OLAP

About Brio Intelligence Documents

Brio Intelligence documents are files you create and use to retrieve information from a database, analyze the information, and build reports. Since Brio Intelligence is an integrated query, analysis, and reporting tool, Brio Intelligence documents have multiple sections, each of which governs one part of the query and reporting process. You create sections progressively as you query a database, retrieve results, and then generate reports.

Brio Intelligence documents may 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 limits, 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 Brio Intelligence 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 EIS sections, which provide an automated push-button interface to a document for use by other users across the enterprise.

Data Source Connections

For Brio Intelligence users, the process of creating a new document and connecting to a database is simple. You select a connection file, or Open Catalog Extension (OCE), for the database server you plan to use and enter your database password. You can select either a new or an existing OCE.

The way you choose an OCE depends on the data model or document with which you are working, and also on which edition of Brio Intelligence you are using:

- When a data model is present in the Query workspace, Brio Intelligence automatically prompts you with the correct connection when your actions require a database connection. You need to be connected to a database when you download a data model and when you process your query to retrieve a data set. In addition, you must be connected to show values for a server limit, to use server functions to create a computed item, or to schedule a document.
- When you open Brio Intelligence to begin a work session (for example, by downloading a data model from a Brio Intelligence Repository, or creating a new data model), you must select the correct connection for the targeted database.

OCEs retain all the information necessary to log on to a specific configuration of database and connection API software. In addition, an OCE retains DBMS-specific connection preferences as well as 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 his or her own database user name when using the OCE, which enforces security measures and privileges that are centralized at the database server.

OCEs have significant advantages in network environments with many database users. One connection can be created for each database connection in the environment and shared with each end-user. Because passwords are not saved with the OCE, there is no danger that distribution will provide unauthorized access to any user who receives the wrong OCE file or acquires it from other sources.

Data Models

After connecting to a database, Brio Intelligence 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 that specify which data to fetch from the database and retrieve for analysis.

Data models make the database more accessible because you can:

- Substitute descriptive names for arcane database table and column names.
- Create custom views of the data.
- Add computed fields for performing calculations on the retrieved data.

In addition to standard data models derived from database tables, Brio Intelligence lets you create metatopics—virtual views independent of the actual database. You can use metatopics to standardize complex calculations and simplify views of the underlying data with intuitive topics customized for business needs.

Depending on their Brio Intelligence tool, users can create their own data models or use data models provided by other users or through the centralized Brio Intelligence Repository, a catalogued storehouse of database views for querying.

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

Thus, you can offer users a raw look at the table schema, or you can hide the complexity by first creating one or more metatopics and then promoting the data model. Each time the user adds a new query, Brio 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 query’s 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. Hyperion's Essbase uses MaxL (Multidimensional Access Language). MDSQL (Multidimensional Query Language) is yet another query language.

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

Brio Intelligence offers two query methods, each of which appears as a separate section within a Brio Intelligence document:

- 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. A Brio Intelligence 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.

A Brio Intelligence 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 Brio Intelligence application’s powerful reporting and analysis tools to create custom views, cross-sections, and drill-downs to slice and dice data and view the multidimensional relationships it contains.

You may create as many different views of the data as you want, and you can 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 Brio Intelligence 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 Brio Intelligence, or you can export the data to other applications for further analysis.

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 you to have some easy method of aggregating data and representing it for easy interpretation.

The Brio Intelligence application's 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.

Brio Intelligence 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 that since the database server returns less data, preaggregation reduces network traffic and takes less time. However, if you preaggregate data at the server, you might find that later when you want to drill down into the data that you cannot reach the data depths that you need. This is because you excluded the more detailed 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. Brio Intelligence automatically aggregates your data for you in report sections.

What Next

You now understand some basic database and data interpretation concepts. This framework will help you make the most of Brio Intelligence. Proceed to Chapter 2, "Getting Started with Brio Intelligence: A Tutorial," and get ready to explore all that Brio Intelligence has to offer.

2

Getting Started with Brio Intelligence: A Tutorial

This chapter offers a tutorial on basic data analysis techniques and familiarizes you with the Brio Intelligence application's powerful capabilities and features. It contains:

- Starting Brio Intelligence
- Opening and Saving a Sample File
- Looking at a Simple Query
- Viewing Results
- Pivoting Data
- Charting Data
- Designing Reports

To use this tutorial, you must have access to Brio Intelligence and the Brio Intelligence sample database and documents. A connection to your company database is *not* necessary.

Starting Brio Intelligence

To start Brio Intelligence on Windows:

- Choose **Start**→**Programs**→**Brio Intelligence Designer** (or **Brio Intelligence Navigator** or **Brio Intelligence Explorer**).

To start Brio Intelligence on the Macintosh:

- Double-click the **Brio Intelligence** icon in the Brio Intelligence folder.

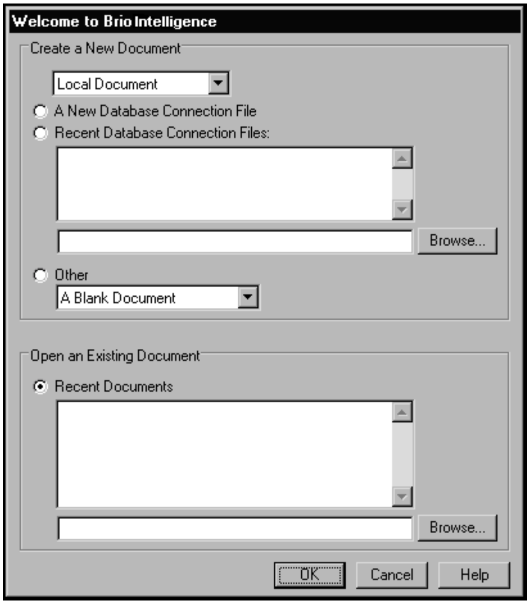
Whenever you start Brio Intelligence, you must first choose whether to create a new document or open an existing document.

Opening and Saving a Sample File

During the installation process, Brio Intelligence installs several sample files. This tutorial uses the document named `Sample1.bqy` to familiarize you with the many Brio Intelligence features.

To open `Sample1.bqy`:

1 In the Welcome to Brio Intelligence dialog box, select **Recent Documents** from the Open Existing Documents area.



Welcome to Brio Intelligence

Create a New Document

Local Document ▾

○ A New Database Connection File

○ Recent Database Connection Files:

Browse...

○ Other

A Blank Document ▾

Open an Existing Document

● Recent Documents

Browse...

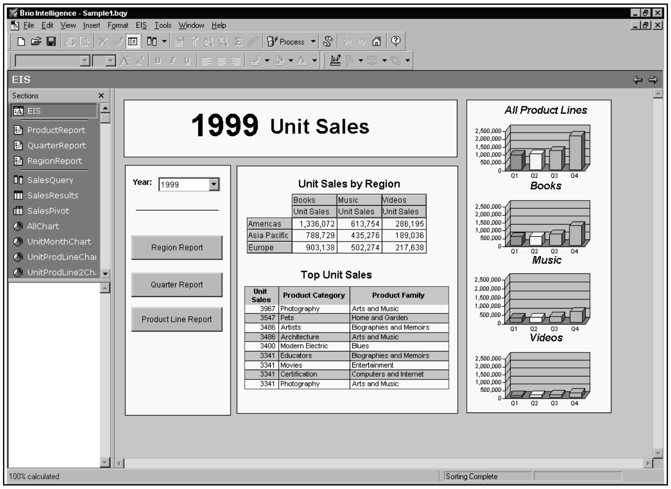
OK Cancel Help

2 Click **Browse** and navigate to the Samples folder in the Brio Intelligence folder.

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

3 Select **`Sample1.bqy`** and click **Open**.

The Brio Intelligence workspace appears and displays the sample document with the EIS section active.



1999 Unit Sales

Year: 1999

Unit Sales by Region

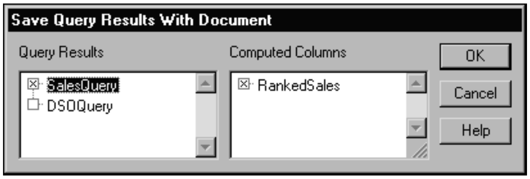
	Books	Music	Videos
	Unit Sales	Unit Sales	Unit Sales
Americas	1,336,072	613,754	286,195
Asia Pacific	788,729	435,276	189,036
Europe	903,138	502,274	217,638

Top Unit Sales

Unit Sales	Product Category	Product Family
3967	Photography	Arts and Music
3547	Pets	Home and Garden
3486	Artists	Biographies and Memoirs
3486	Architecture	Arts and Music
3400	Modern Electric	Blues
3341	Educators	Biographies and Memoirs
3341	Movies	Entertainment
3341	Certification	Computers and Internet
3341	Photography	Arts and Music

4 Choose **File→Save Options→Save Query Results with Document**.

The Save Query Results With Document dialog box opens.



Query Results

☒ SalesQuery
☐ DSOQuery

Computed Columns

☒ RankedSales

5 Make sure that the all of the items in the Query Results and the Computed Columns lists are selected, and then click **OK**.

6 Choose **File→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

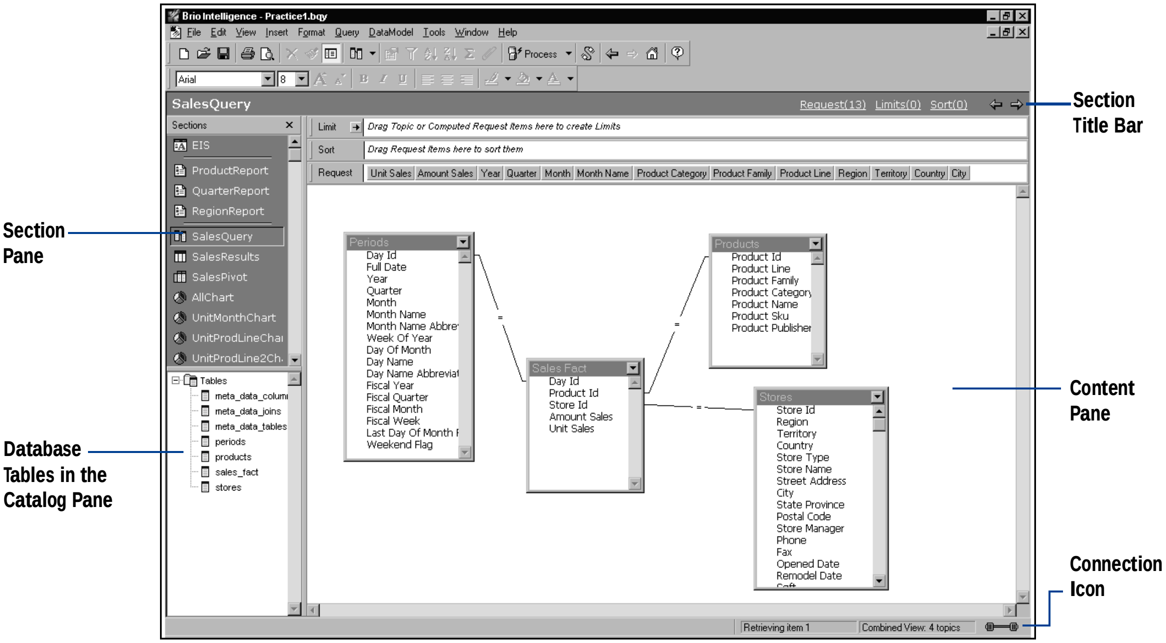
The Query section is the foundation of a Brio Intelligence document. It is the space where you build questions for the database.

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:

➤ In the Sections pane, select **SalesQuery**.



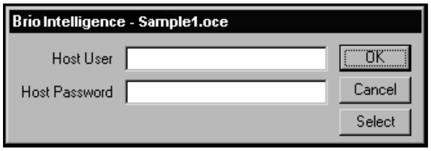
Viewing Database Tables

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 Brio 6.0 Sample1.oce dialog box appears and prompts you for a user name and password.



An OCE is an Open Catalog Extension file that allows you to connect to a database. The following icons on the Status bar indicate your connection status:

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 Brio Intelligence 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.

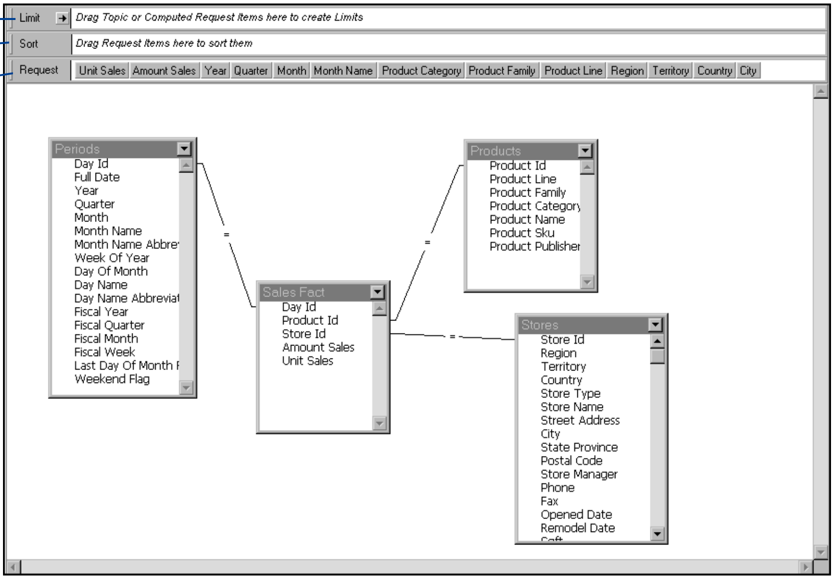
The Tables tree in the Catalog pane expands to show all of the tables in the database.

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*.


Limit Line
Sort Line
Request Line

You build queries by adding topics from the Content pane to the Request line. You can drag and ddrop any topic item to the Request line. When you process a query, Brio Intelligence 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 *limits* to the data, or specify columns by which to *sort* the data. You can apply limits and sorts in either the Query section or the Results section.

In the Query section, limits 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.

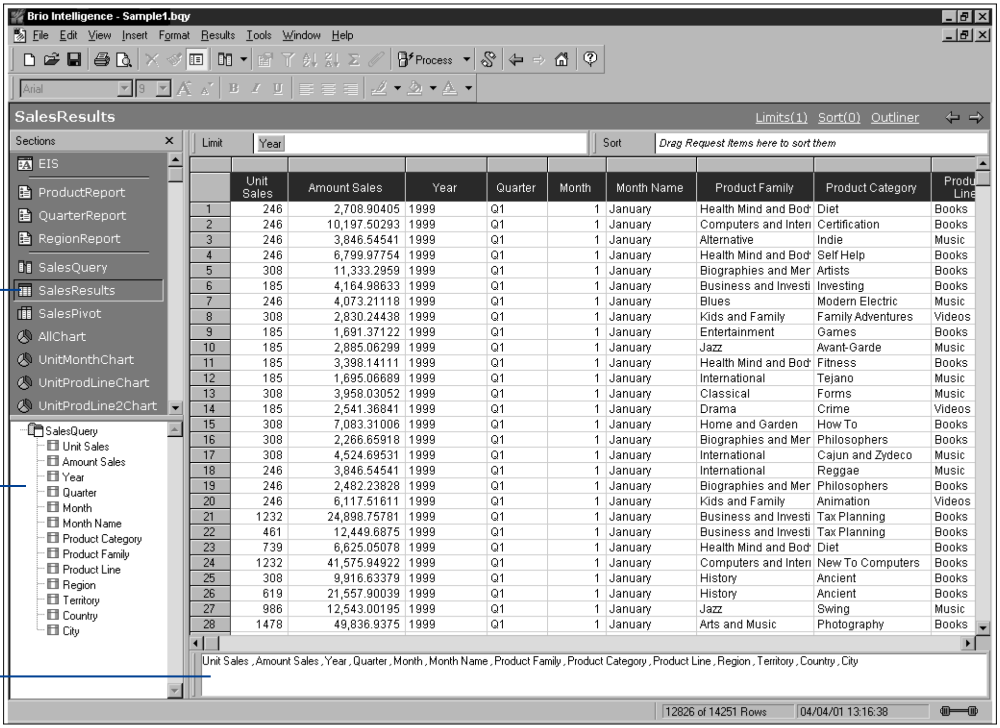
Viewing Results

Data returned from a query appears 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:

➤ In the Sections pane, select **SalesResults**.

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



Limit: Year

Sort: *Drag Request Items here to sort them*

	Unit Sales	Amount Sales	Year	Quarter	Month	Month Name	Product Family	Product Category	Produ Line
1	246	2,708.90405	1999	Q1	1	January	Health Mind and Bod	Diet	Books
2	246	10,197.50293	1999	Q1	1	January	Computers and Interi	Certification	Books
3	246	3,846.54541	1999	Q1	1	January	Alternative	Indie	Music
4	246	6,799.97754	1999	Q1	1	January	Health Mind and Bod	Self Help	Books
5	308	11,333.2959	1999	Q1	1	January	Biographies and Mer	Artists	Books
6	185	4,164.98633	1999	Q1	1	January	Business and Investi	Investing	Books
7	246	4,073.21118	1999	Q1	1	January	Blues	Modern Electric	Music
8	308	2,830.24438	1999	Q1	1	January	Kids and Family	Family Adventures	Videos
9	185	1,691.37122	1999	Q1	1	January	Entertainment	Games	Books
10	185	2,885.06299	1999	Q1	1	January	Jazz	Avant-Garde	Music
11	185	3,398.14111	1999	Q1	1	January	Health Mind and Bod	Fitness	Books
12	185	1,695.06689	1999	Q1	1	January	International	Tejano	Music
13	308	3,958.03052	1999	Q1	1	January	Classical	Forms	Music
14	185	2,541.36841	1999	Q1	1	January	Drama	Crime	Videos
15	308	7,083.31006	1999	Q1	1	January	Home and Garden	How To	Books
16	308	2,266.65918	1999	Q1	1	January	Biographies and Mer	Philosophers	Books
17	308	4,524.69531	1999	Q1	1	January	International	Cajun and Zydeco	Music
18	246	3,846.54541	1999	Q1	1	January	International	Reggae	Music
19	246	2,482.23828	1999	Q1	1	January	Biographies and Mer	Philosophers	Books
20	246	6,117.51611	1999	Q1	1	January	Kids and Family	Animation	Videos
21	1232	24,898.75781	1999	Q1	1	January	Business and Investi	Tax Planning	Books
22	461	12,449.6875	1999	Q1	1	January	Business and Investi	Tax Planning	Books
23	739	6,625.05078	1999	Q1	1	January	Health Mind and Bod	Diet	Books
24	1232	41,575.94922	1999	Q1	1	January	Computers and Interi	New To Computers	Books
25	308	9,916.63379	1999	Q1	1	January	History	Ancient	Books
26	619	21,557.90039	1999	Q1	1	January	History	Ancient	Books
27	986	12,543.00195	1999	Q1	1	January	Jazz	Swing	Music
28	1478	49,836.9375	1999	Q1	1	January	Arts and Music	Photography	Books

Unit Sales , Amount Sales , Year , Quarter , Month , Month Name , Product Family , Product Category , Product Line , Region , Territory , Country , City

12826 of 14251 Rows 04/04/01 13:16:38

SalesResults

Request Items in the Catalog Pane

Columns in the Outliner

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 Outliner.

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.

Limiting Data

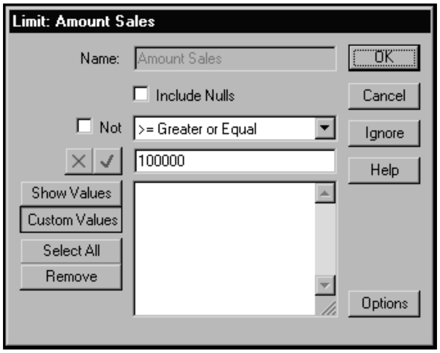
At times you may have more data in a column than you want. Use the Limit line to limit the data displayed in a column.

To limit data:

1 If the Limit line does not show, click **Limits(0)** on the Section title bar.

2 Drag **Amount Sales** from the Catalog pane to the Limit line. **[Ctrl+L]**

The Limit dialog box appears.



Limit: Amount Sales

Name: Amount Sales

□ Include Nulls

□ Not >= Greater or Equal

100000

3. Select **>=Greater or Equal** from the Limit 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 limit:

➤ Delete **Amount Sales** from the Limit line. **[Del]**

The data reappears when you remove the limit.

Calculating Data

Brio Intelligence can perform calculations on columns of numeric data.

To sum up numeric data:

1. Select the **Amount Sales** column.
2. Σ Choose **Results**→**Grand Total** and click **OK** in the dialog box that appears.
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 you see does not match $132,881, you need to remove any limits imposed on the results. Delete any limits from the Limit 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:

➤ Choose **Format**→**Column**→**Auto-Size Width**. **[Ctrl+A][Ctrl+E]**

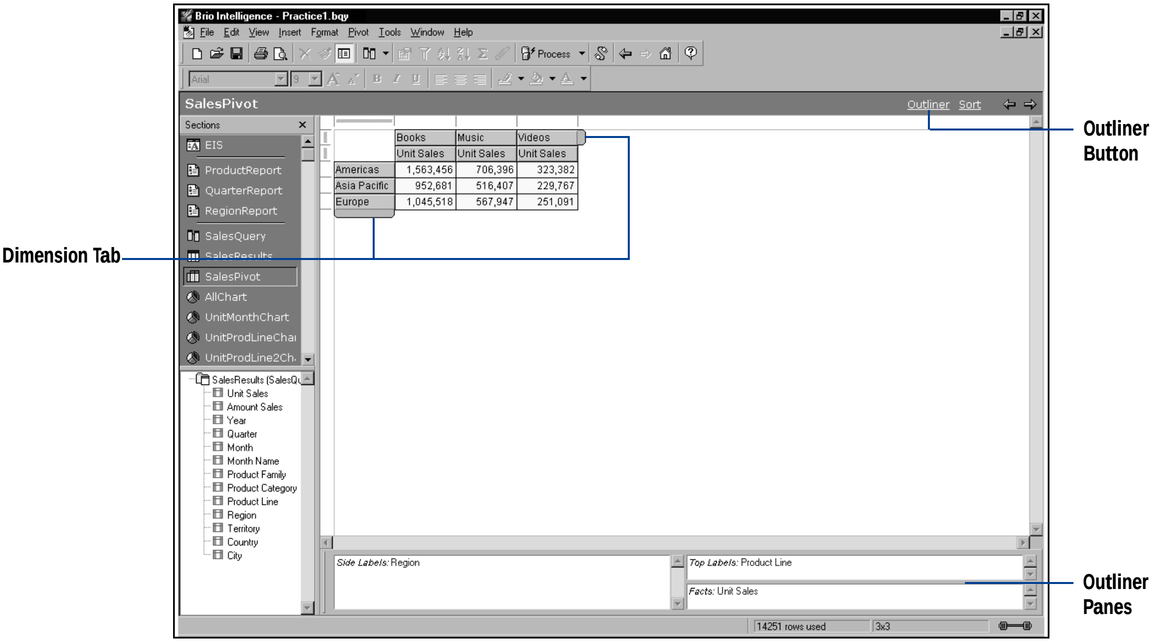
Pivoting Data

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:

➤ In the Section pane, select **SalesPivot**.

The SalesPivot section appears.



	Books	Music	Videos
	Unit Sales	Unit Sales	Unit Sales
Americas	1,563,456	706,396	323,382
Asia Pacific	952,681	516,407	229,767
Europe	1,045,518	567,947	251,091

Side Labels: Region

Top Labels: Product Line

Facts: Unit Sales

14251 rows used 3x3

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. Choose **Insert→New Pivot**.
3. If the Outliner is not already visible, click **Outliner** on the Section title bar.

Drag one or more items from the Catalog pane into each of the Outliner panes. Remove items from the Outliner panes and add new ones. Use the Top Labels pane and Side Labels pane for text. Use the Facts pane for numeric values.

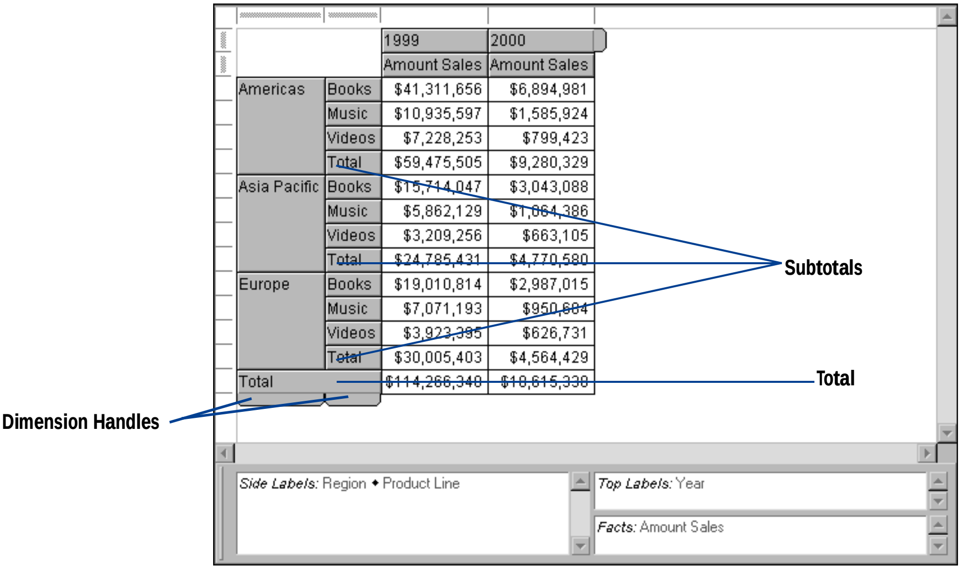
To pivot views:

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.



		1999	2000
		Amount Sales	Amount Sales
Americas	Books	$41,311,656	$6,894,981
	Music	$10,935,597	$1,585,924
	Videos	$7,228,253	$799,423
	Total	$59,475,505	$9,280,329
Asia Pacific	Books	$15,714,047	$3,043,088
	Music	$5,862,129	$1,064,386
	Videos	$3,209,256	$663,105
	Total	$24,785,431	$4,770,580
Europe	Books	$19,010,814	$2,987,015
	Music	$7,071,193	$950,684
	Videos	$3,923,395	$626,731
	Total	$30,005,403	$4,564,429
Total		$114,266,340	$18,615,338

Side Labels: Region ✦ Product Line

Top Labels: Year

Facts: Amount Sales

To calculate totals and subtotals (as displayed above):

1. Drag **Product Line** and **Region** to the Side Labels pane in Outliner.
2. Drag **Year** to the Top Labels pane in Outliner.
3. Drag **AmountSales** to the Facts pane in Outliner.

4 Select the dimension handle for **Region** (click at the bottom of the Region column).

5 Choose **Pivot→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 choose **Pivot→Add Totals**.

A row is added that shows the subtotals (also know as break totals) for each product line by region.

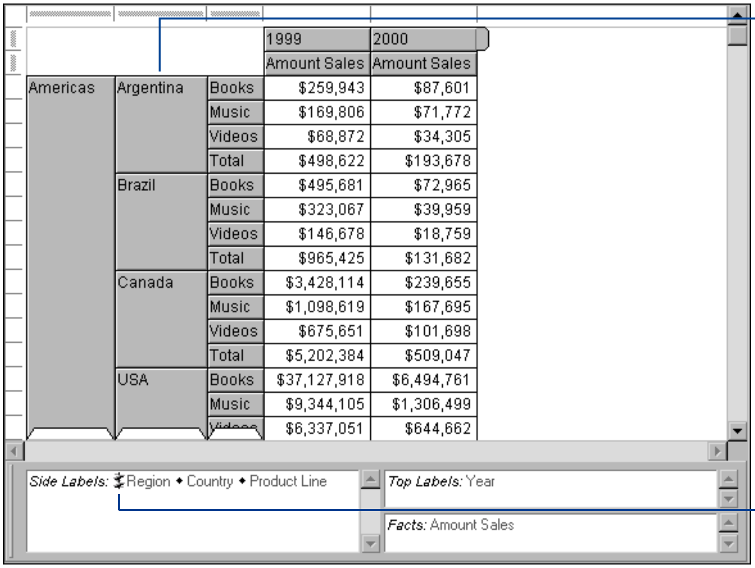
Drilling Down

More data is available for analysis than is currently visible in your pivot table.

To drill anywhere:

➤ Select the Region column and choose **Pivot→Drill Anywhere→*Country***.

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



			1999	2000
			Amount Sales	Amount Sales
Americas	Argentina	Books	$259,943	$87,601
		Music	$169,806	$71,772
		Videos	$68,872	$34,305
		Total	$498,622	$193,678
	Brazil	Books	$495,681	$72,965
		Music	$323,067	$39,959
		Videos	$146,678	$18,759
		Total	$965,425	$131,682
	Canada	Books	$3,428,114	$239,655
		Music	$1,098,619	$167,695
		Videos	$675,651	$101,698
		Total	$5,202,384	$509,047
	USA	Books	$37,127,918	$6,494,761
		Music	$9,344,105	$1,306,499
		Videos	$6,337,051	$644,662

Side Labels: Region • Country • Product Line

Top Labels: Year

Facts: Amount Sales

Drill to Country

Drill Indicator

To restore the original pivot table without the Unit Sales column:

➤ Select the Country column and choose **Pivot→Drillup**.

Hiding Data

You can temporarily hide data.

To hide and focus on items:

➤ Select a label such as the Americas label and choose **Pivot→Hide Items**.

To restore your excluded items:

➤ Choose **Pivot→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, choose **View→Toolbars→Formatting**.

To add a line color:

1 Click the **label**, **dimension handle**, or **column** whose line color you want to change.

2 On the Format toolbar, open the **Line Color** list and select a color from the palette.

To add a fill color:

1 Click the **label**, **dimension handle**, or **column** whose fill color you want to change.

2 On the Format toolbar, open the **Fill Color** list and select a color from the palette.

To add a text color:

1 Click the **label**, **dimension handle**, or **column** whose text color you want to change.

2 On the Format toolbar, open the **Text Color** list and select a color from the palette.

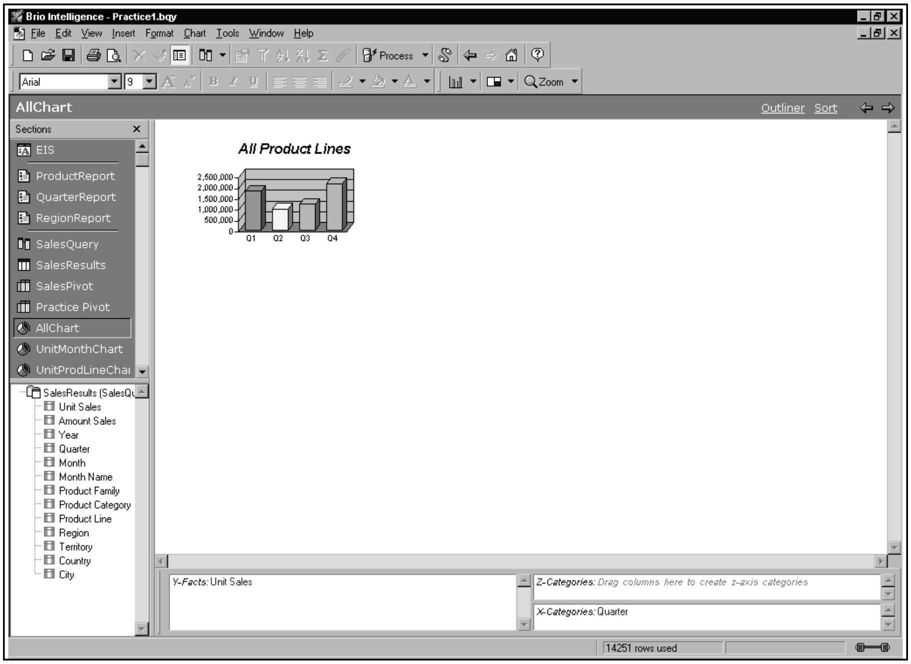
Charting Data

Brio Intelligence 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:

➤ In the Section pane, select **AllChart**.

The AllChart section appears.



Y-Facts: Unit Sales

Z-Categories: *Drag columns here to create z-axis categories*

X-Categories: Quarter

14251 rows used

Brio Intelligence rapidly converts data from one chart type to another.

To change the chart type and format:

➤ From the Chart list, select a chart format.

As you shift from one chart type to another, data may be shifted to different axes.

In a pie chart, you can draw attention to individual slices by pulling them out of the pie. To do so, select a slice choose Pull Out Slice on the shortcut menu.

To create a new chart:

1 In the Section pane, select **SalesQuery**.

2 Choose **Insert→New Chart**.

Chart appears in the Section pane.

3 In the Section pane, double-click **Chart**.

The Section Label dialog box appears.

4 Delete Chart and type **Unit Sales Region**, then click **OK**.

5 If the Outliner is not already visible, click **Outliner** on the Section title bar.

Numeric values (facts) are placed in the Y-Facts pane in Outliner. Non-numeric data (dimensions) are placed in the X-Categories and Z-Categories panes in Outliner.

6 Drag **Unit Sales** to the X-Facts pane in Outliner.

7 Drag **Region** to the Z-Categories pane in Outliner.

8 Drag **Product Line** to the Y-Categories pane in Outliner.

9 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.

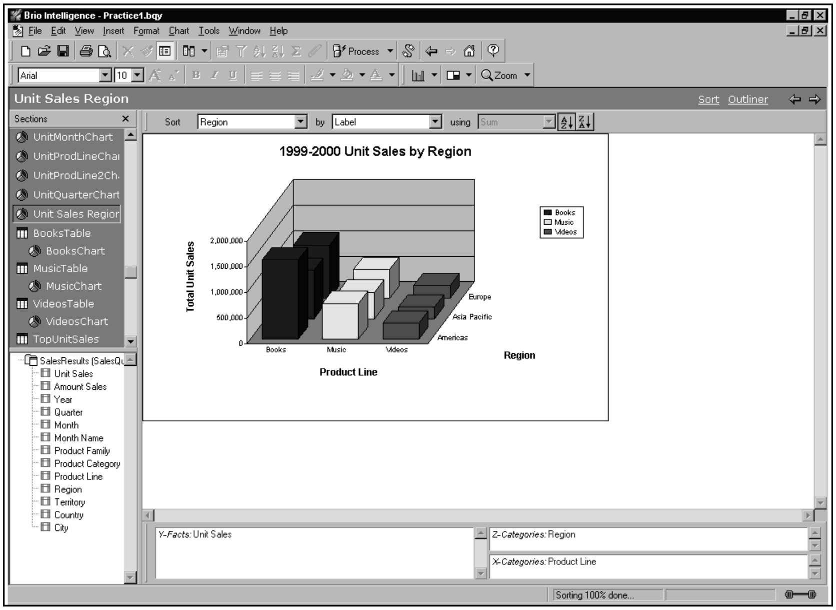
To sort your chart:

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 appears.

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

Drag the text box to any position on the chart.

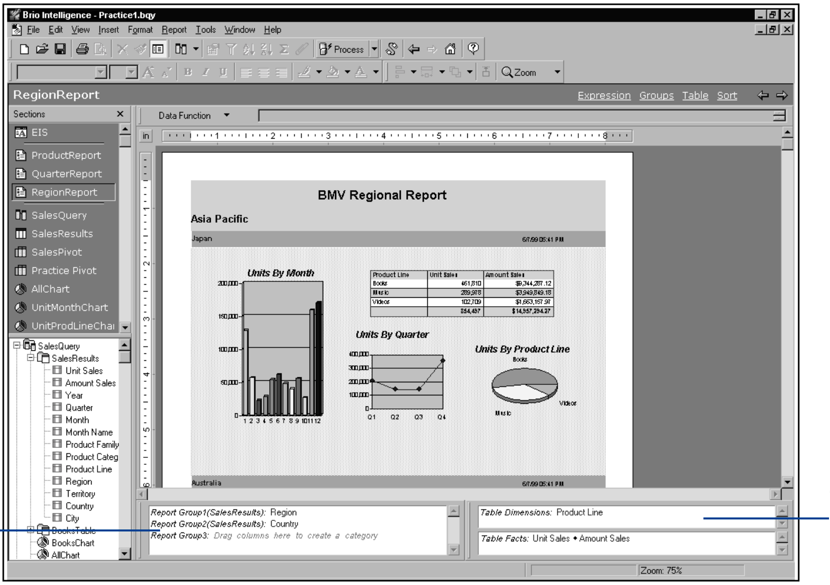
Designing Reports

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 appears.



BMV Regional Report

Asia Pacific

Japan 6/7/99 05:41 PM

Product Line	Unit Sales	Amount Sales
Books	461,810	$9,344,287.12
Music	289,978	$3,949,849.18
Videos	102,709	$1,663,157.97
	854,497	$14,957,294.27

Australia 6/7/99 05:41 PM

Report Group1(SalesResults): Region
Report Group2(SalesResults): Country
Report Group3: Drag columns here to create a category

Table Dimensions: Product Line

Table Facts: Unit Sales ◆ Amount Sales

Zoom: 75%

Groups Outliner

Table Outliner

To create a new tabular report:

1 Choose **Insert→New Report**.

Blank columns appear in the Content pane.

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

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

The Groups Outliner allows you to drag nonquantifiable 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, you instruct Brio Intelligence to organize data in repeating collections of records according to the Report group.

The Table Outliner 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 nonquantifiable items to itemize the facts.

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

4 Drag **Year** and **Quarter** into the Dimensions pane in Table Outliner.

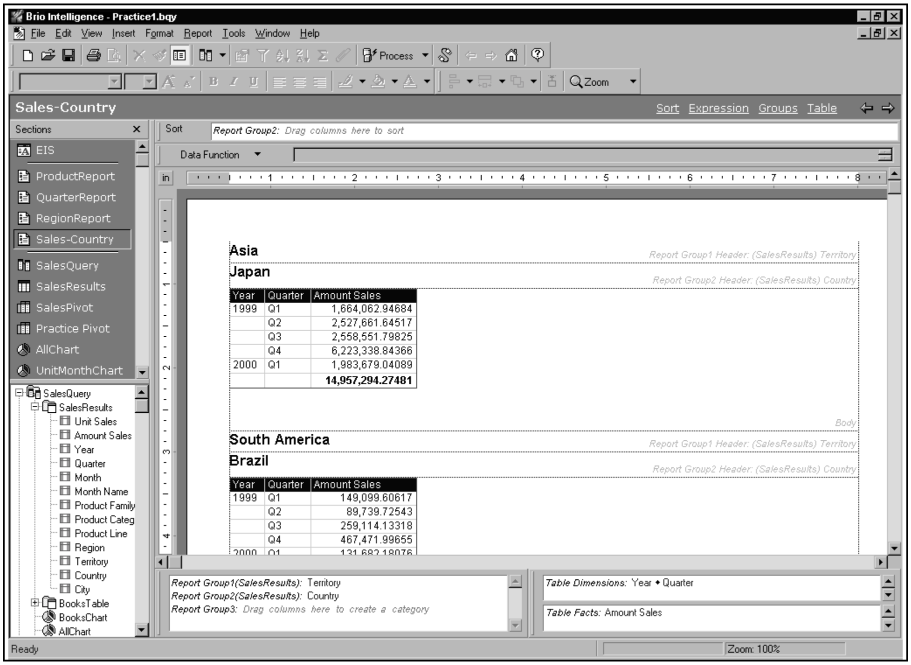
5 Drag **Amount Sales** into the Facts pane in Table Outliner.

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

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

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.



Report Group2: Drag columns here to sort

Asia

Report Group1 Header: (SalesResults) Territory

Japan

Report Group2 Header: (SalesResults) Country

Year	Quarter	Amount Sales
1999	Q1	1,664,062.94684
	Q2	2,527,661.64517
	Q3	2,558,551.79825
	Q4	6,223,338.84366
2000	Q1	1,983,679.04089
		14,957,294.27481

Body

South America

Report Group1 Header: (SalesResults) Territory

Brazil

Report Group2 Header: (SalesResults) Country

Year	Quarter	Amount Sales
1999	Q1	149,099.60617
	Q2	89,739.72543
	Q3	259,114.13318
	Q4	467,471.99655
2000	Q1	131,682.18076

Report Group1(SalesResults): Territory
Report Group2(SalesResults): Country
Report Group3: Drag columns here to create a category

Table Dimensions: Year ♦ Quarter

Table Facts: Amount Sales

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's chart contains data specific to that year.

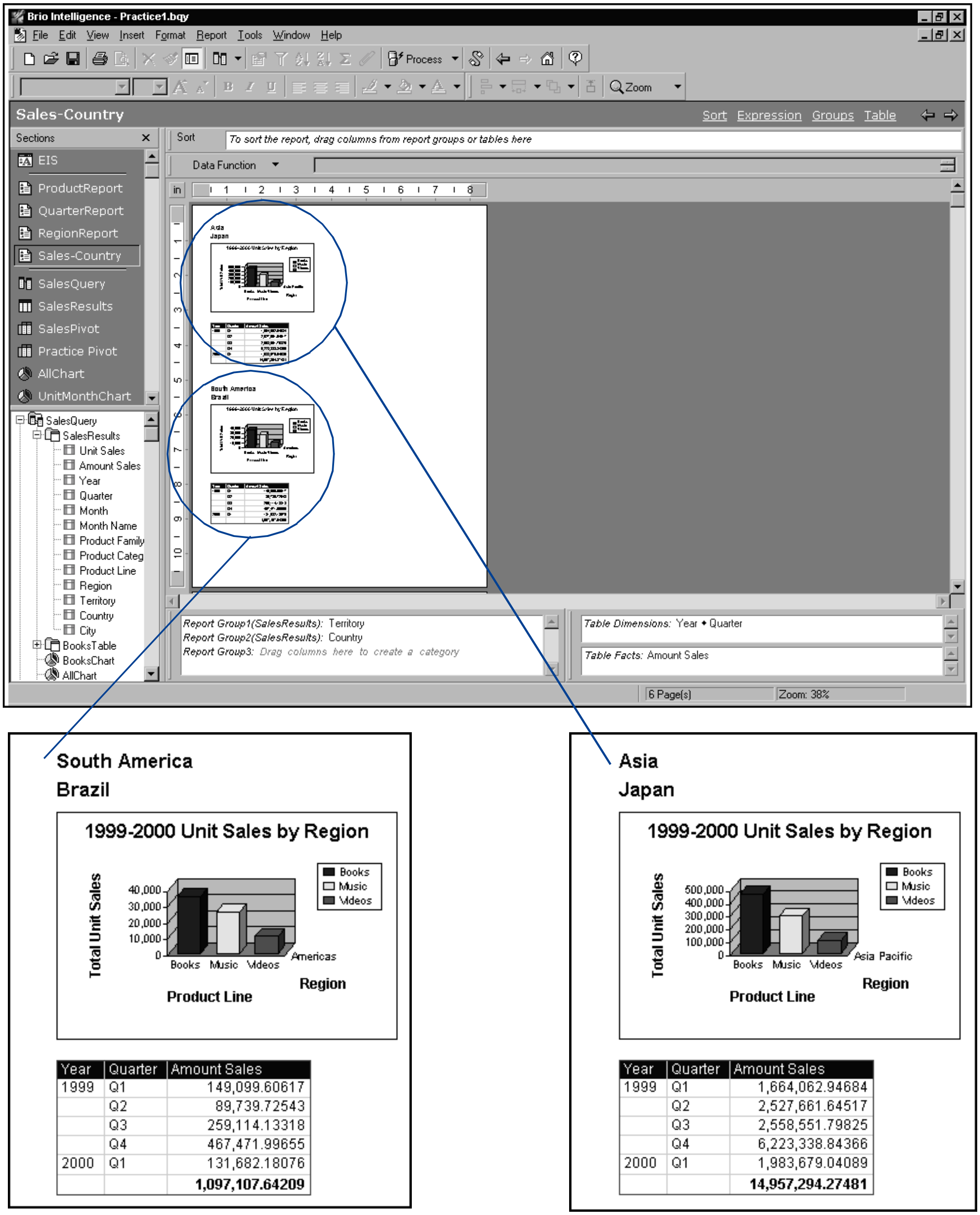
To create a Smart report:

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.



To sort the report, drag columns from report groups or tables here

Report Group1(SalesResults): Territory
Report Group2(SalesResults): Country
Report Group3: Drag columns here to create a category

Table Dimensions: Year ♦ Quarter
Table Facts: Amount Sales

6 Page(s)   Zoom: 38%

South America
Brazil

Year	Quarter	Amount Sales
1999	Q1	149,099.60617
	Q2	89,739.72543
	Q3	259,114.13318
	Q4	467,471.99655
2000	Q1	131,682.18076
		1,097,107.64209

Asia
Japan

Year	Quarter	Amount Sales
1999	Q1	1,664,062.94684
	Q2	2,527,661.64517
	Q3	2,558,551.79825
	Q4	6,223,338.84366
2000	Q1	1,983,679.04089
		14,957,294.27481

Sorting Columns

To sort table columns:

➤ Select a table column, and then 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 Choose **Report→Report Setup**.

The Report Page Setup dialog box appears.

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, then click **OK**.

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

To insert a page break:

➤ Select a Report Group header (be careful not to select the Report Group label) and on the shortcut menu, choose **Page Break After** or **Page Break Before**.

To remove a page break:

➤ Select a Report Group header with a page break applied, and on the shortcut menu, choose **Page Break After** or **Page Break Before** to remove the check mark.

What Next

You have seen some of the powerful functionality of Brio Intelligence and the ease with which you can use it. While there is still a great deal more to learn about Brio Intelligence, you now have a solid foundation for learning additional techniques. To find out more about the Brio Intelligence workspace and commands that are available to you, proceed to Chapter 3, “Brio Intelligence Basics.”

3

Brio Intelligence Basics

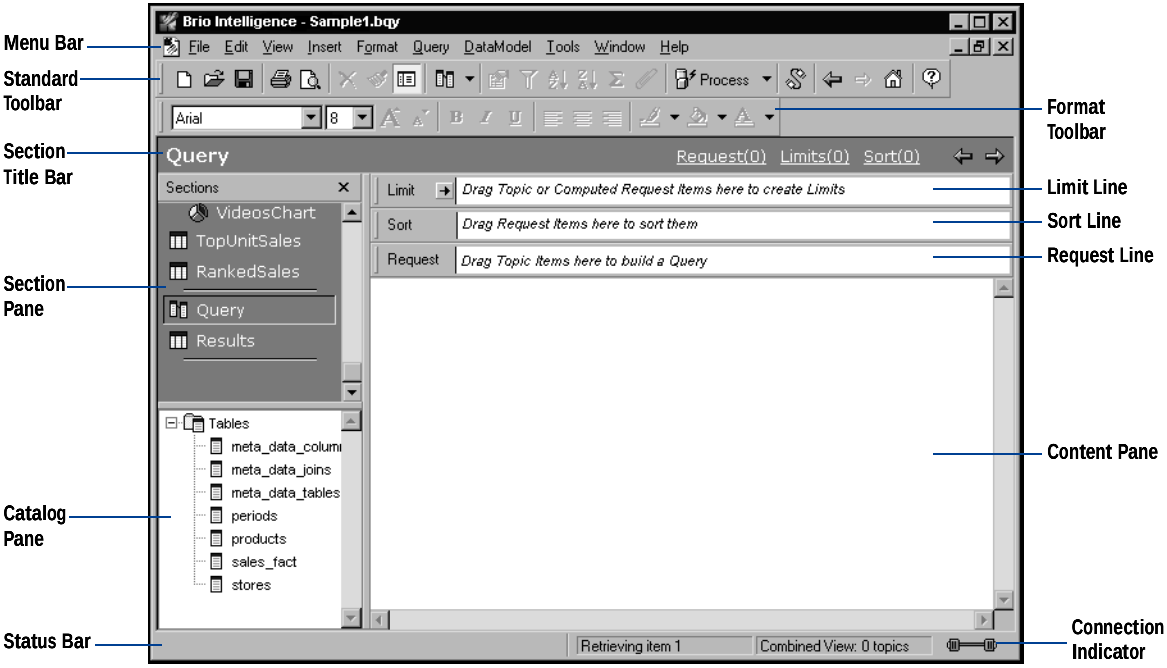
This chapter provides an overview of the Brio Intelligence workspace and describes fundamental Brio Intelligence features and functions. These features, available through the File, Edit, View, Insert, and Tools menus, help you manage Brio Intelligence documents and provide numerous ways to manipulate data in the query and results sections.

This chapter contains:

- Exploring the Brio Intelligence Workspace
- Managing Documents
- Using Edit Commands
- Changing Workspace Views
- Inserting Sections and Breaks
- Formatting Text and Other Elements
- Working with Document Sections
- Setting Brio Intelligence Options

Exploring the Brio Intelligence Workspace

When you start Brio Intelligence and create a new document or open an existing document, the Brio Intelligence workspace appears.



The Brio Intelligence workspace is a tri-pane window that provides the tools and the access you need to perform database operations. You can resize the window and customize the appearance of most of the elements in the workspace, and you can decide whether certain elements should be visible.

Toolbars, buttons, Outliners, and palettes provide quick access to most Brio Intelligence functions. All the Brio Intelligence toolbars, Outliners, and palettes are dockable; that is, they can be detached and dragged as floating windows to any location on the workspace. You can also hide these objects from view to reclaim screen space for report viewing and EIS applications.

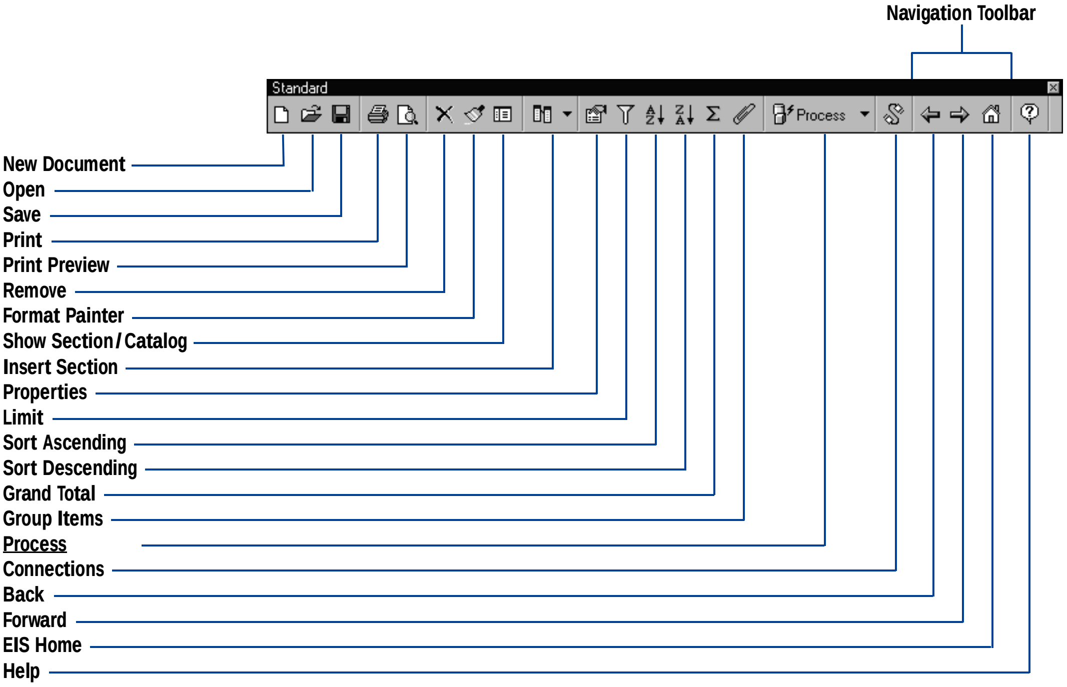
Main Menu

The Main menu provides access to all Brio Intelligence menus and functions. In addition, certain menus change based on the active Brio Intelligence section. For example, only the Query section has the Query and Data Model menus. Every section has a unique menu whose label matches the name of the section. Brio Intelligence also provides extensive shortcut menus.

Toolbars

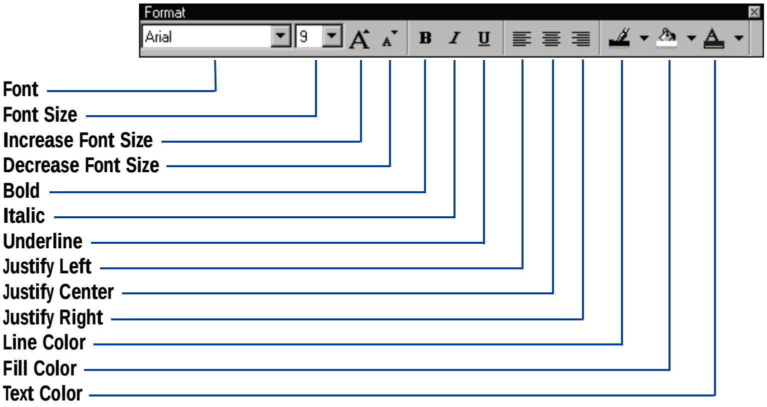
Toolbars provide quick access to frequently used functions and features. Available toolbars include:

- Standard toolbar – contains icons for commonly used operations. The availability of an icon depends on the active Brio Intelligence section.



- Navigation toolbar– Returns to an EIS section from another section when the Section catalog, Section title bar, toolbars, and menus have been turned off. This toolbar is hidden by default. When activated, it includes the Back and Forward buttons and an EIS Home button.

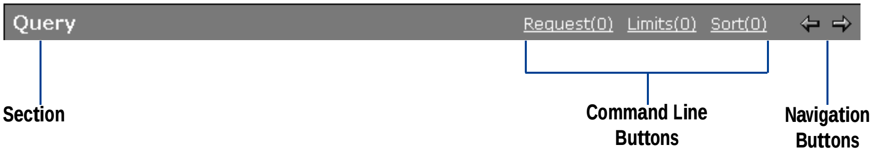
- Format toolbar – provides text formatting, styling, and editing commands.



- Section toolbar – Provides commands for a specific Brio Intelligence document section.

Section Title Bar

The Section title bar displays the name of the active section (that is, the section in which you are currently working). The Section title bar includes navigation buttons, as well as buttons that correspond to the command lines and Outliners available in a particular section.



Command Lines

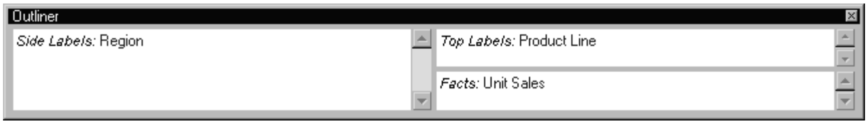
You use command lines to complete important operations during the query and reporting processes. The command lines that appear are determined by the active section. The command lines available in each section are:

- Query Section – Request, Limit, and Sort
- Results Section – Limit and Sort
- Report Section – Sort and Expression

Note All command lines can be docked, resized, or hidden.

Outliners

Outliners are drag-and-drop templates used in the Pivot, Chart, Results, OLAPQuery, and Report sections. Each Outliner pane corresponds to a specific layout element of the report. Outliners allow you to easily view, plot, and manipulate the data in the Content pane.

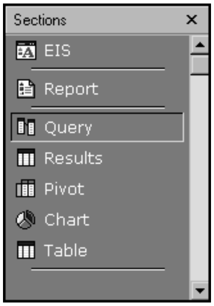


Side Labels: Region

Top Labels: Product Line

Facts: Unit Sales

Section Pane



The Section pane lists the sections available in the current document. Possible sections include:

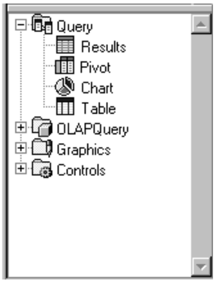
- EIS (Executive Information System) – A document front-end that can be customized, which makes it easy for developers to build and deploy analytic applications and for end users to access information.
- Report – A dynamic and analytical designer that provides the ability to easily develop a complete range of reports. Use the Report Designer's visual layout capabilities to drag and drop columns, expression, charts, logos, bitmaps, and other items to quickly design and customize your reports.
- Query – The foundation of the Brio Intelligence document, the Query section enables you to connect to a relational database and download a prebuilt query or data model, or build a new data model and create your own query.
- Results – Created when you process a query or import data. Brio Intelligence retrieves data to your desktop and displays it in the Results sections as columns in a table.
- Pivot – An interactive table that quickly summarizes, or cross-tabulates, large amounts of data. You can rotate its rows and columns to see different summaries of the source data, or display the details for areas of interest.
- Charts – A fully interactive, two- or three-dimensional view of your data that provides powerful ways to visually analyze your data.
- Table – A single-dimension report that displays your data in columns. Tables are often used as building blocks in other sections.

- OLAPQuery – A query section specifically designed for connecting to multidimensional databases.
- OLAPResults – Results section for an OLAPQuery.

By default, a Brio Intelligence document has at least one Query section and one Results section.

Catalog Pane

The Catalog pane contains the objects you use to build contents.

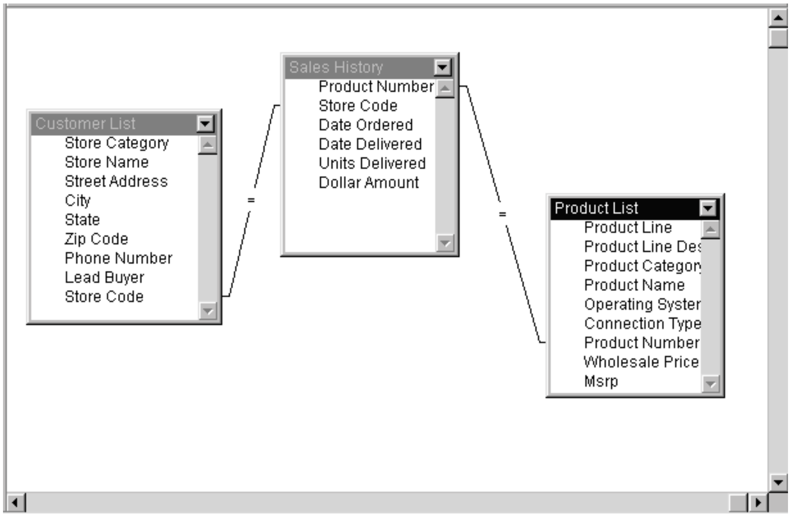


For example in the Pivot section, the Catalog pane contains Results columns. In the Query section, the Catalog pane contains the tables available in your database. In the Report Designer section, the catalog pane contains the Results, Pivot, and Chart sections in addition to available graphic and field elements

To use an object, drag it from the Catalog pane to the Content pane or to the Outliner.

Content Pane

The Content pane displays a data model or provides a view of the content from a Query, Pivot, Chart, or EIS section.



Status Bar

The Status bar, located below the Content pane, displays information about the current database connection, document, and document section:

- The area immediately to the left of the connection icon provides information about the current document section.
- The area to the far left displays information about data retrieval and the current database connection:

Connected

Disconnected

Managing Documents

Managing documents involves:

- Maintaining Documents
- Saving Documents
- Specifying Save Options
- Working with Data Sources
- Exporting Data
- Printing Documents
- Emailing Documents
- Quitting Brio Intelligence

You can accomplish all of these tasks by using the commands on the File menu.

Note See “File Menu Command Reference” on page 3-27 for an overview of all the File menu commands.

Maintaining Documents

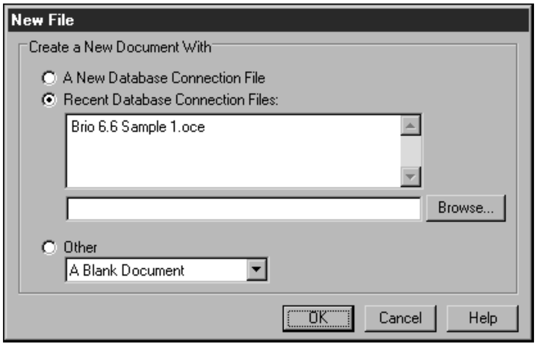
Use the following commands to create new documents, open existing documents, and close documents.

Note Users of the Web-based plug-ins (Freeview, Quickview, and Insight) access files through a Web browser.

To create a new Brio Intelligence document:

1 Choose **File**→**New**, or click the **New** icon on the Standard toolbar. **[Ctrl+N]**

The New File dialog box appears.



New File

Create a New Document With

○ A New Database Connection File

● Recent Database Connection Files:

Brio 6.6 Sample 1.oce

○ Other

A Blank Document

2 Select a database connection file.

For more information about connection files, see “Data Source Connections” on page 1-9. See also the *Brio Intelligence 6.6 Administrator’s Guide*.

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

To open an existing Brio Intelligence document:

1 Choose **File**→**Open**, or click the **Open Document** icon on the Standard Toolbar.**[Ctrl+O]**

The Open File dialog box appears.

2 Select the file you want to open and click **OK**.

To close a document:

➤ Choose **File**→**Close**. **[Ctrl+W]**

Brio Intelligence prompts you to save changes to your document.

Saving Documents

Saving a document saves the current formatting and layout of all Brio Intelligence sections in the document.

To save a document:

Choose **File**→**Save**. **[Ctrl+S]**

To save the document under a different name:

1 Choose **File**→**Save As**.

The Save File dialog box appears.

2 Type a name for the document in the **File Name** field and click **Save**.

Specifying Save Options

To select specific save options:

➤ Choose **File**→**Save Options** and select the desired option.

Save options include:

- Save Query Results With Document
- Compress Document
- Password Protect Document
- Password Protect Designer Mode

Save Query Results With Document

This option saves the results of a query and any computed columns (from the results) with a document and allows you to analyze and generate reports using the results set without being connected to the database.

For more information about saving results, see “Saving Results Sets” on page 5-12.

Compress Document

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

☆ **Tip** To configure Brio Intelligence to compress all documents when saving, use the Compress All Documents option. See “Working with Document Sections” on page 3-34 for more information.

Password Protect Document

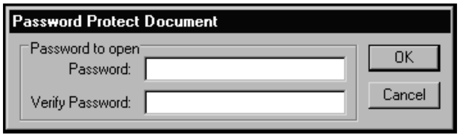
Use the Password Protect Document option to restrict access to a document and ensure the confidentiality of sensitive documents. This option requires users to input a correct password for document access.

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

To assign a password to a document:

1 Choose **File**→**Save Options**→**Password Protect Document**.

The Password Protect Document dialog box appears.



2 In the **Password** field, type the password you want to assign to this document.

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 appears.

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 an EIS section and when you are working with document scripts.

For more information on this command, see *BrioQuery Object Model and Executive Information Systems*.

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.

Open From Repository

Use the Open From Repository command to select a prebuilt data model as the basis for a new document.

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.

Note Your administrator grants you access to the Brio Intelligence Repository that contains the data models you need. Before selecting from a Repository catalog, make sure you have connections for the correct databases. If necessary, check with your administrator to see which database contains the Brio Intelligence Repository, or in which owner area of the Repository a particular object is located.

To open a document using a repository object:

1 Choose **File**→**Open From Repository**→**Select**.

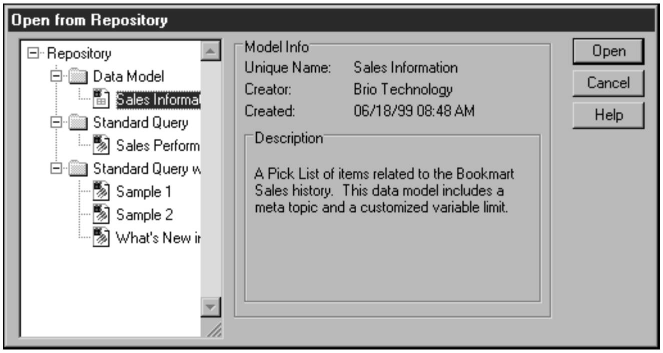
The Select Connection dialog box appears.

Note You can also choose to use the connection file currently in use, if there is one. Current OCEs are listed below the Select menu item.

2 Select the **connection file** you want 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 appears and displays information about the selected object.



Open from Repository

Model Info

Unique Name: Sales Information

Creator: Brio Technology

Created: 06/18/99 08:48 AM

Description

A Pick List of items related to the Bookmart Sales history. This data model includes a meta topic and a customized variable limit.

- 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 you want to use and click **Open**.

Brio Intelligence downloads the repository object to the appropriate section.

Save To Repository

Brio Intelligence 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 Brio Intelligence users.

Register For OnDemand Server

Use the Register For OnDemand Server command to register Brio Intelligence documents or repository data models to the Brio OnDemand Server for use on the Web.

Note You must be authorized by your Brio Intelligence administrator to register a document or repository data model for the OnDemand Server.

To register an object for use by the OnDemand Server:

1 Choose **File**→**Register For OnDemand Server**→**Select**.

The Select Connection dialog box appears.

Note You can also choose to use the connection file currently in use, if there is one. Current OCEs are listed below the Select menu item.

2 Select the connection file you want to use and click **OK**.

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

The Register For OnDemand Server dialog box appears.

4 Provide the following information:

- Database owner name of the OnDemand Server’s repository
- A unique, descriptive name for the document
- A description of the document contents

5 Select the database connection used by OnDemand Server to process the document query.

6 Use the arrows to move groups from the Available Groups to the Selected Groups list.

As each group is added, it appears in the list with an icon and check mark. The icon indicates the group’s privilege level; the check mark indicates whether or not the user members must provide a database user name and password when connecting to the OnDemand Server.

7 Visual Warehouse users only: To register the document to IBM's Information Catalog Repository, click **Register to IBM Information**.

After the document has been uploaded to the repository (after the next step), the Connect To IBM Information Catalog dialog appears.

8 Click **OK** to upload the data model to the repository.

Schedule

Use the Schedule command to schedule a document to the Broadcast Server for processing. For information on using the schedule command, see the *Brio Intelligence 6.6 Administrator's Guide*.

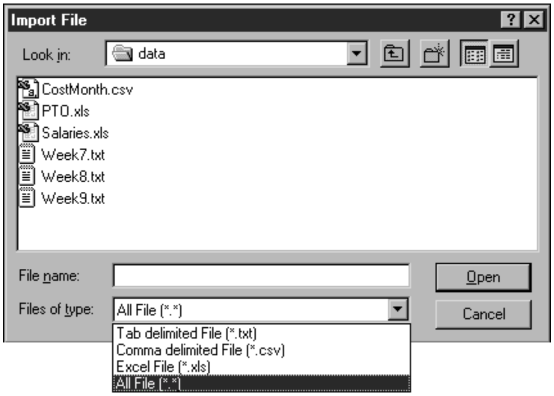
Import Data Files

Use the Import Data File command to use data stored in Microsoft Excel, tab-delimited, or comma-delimited file formats. The data appears in the Results section. You can then use the imported data to build reports and perform data analysis.

To import a data file:

1 Choose **File→Import Data File→Data File**.

The Import File dialog box appears.



2 Navigate to the **location** of the **file** you want to import.

Select a file type from the Files Of Type drop-down list to make the file easier to find. Choose from:

- Tab delimited (`.txt`)
- Comma delimited (`.csv`)
- Excel File (`.xls`)

3 Select the **file** you want to import and click **OK**.

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

Import SQL

The Import SQL command allows you to take a complete SQL statement from a text file, import 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 you want to import begins with a SELECT statement.
- The Query section of your Brio Intelligence document 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.
- You cannot drag items from the table to the Request line.
- You cannot use the custom SQL feature.
- You cannot 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 Choose **File**→**Import Data File**→**SQL**.

The Import File dialog box appears.

2 Navigate to the location of the file you want to import.

3 Select the file you want to import and click **OK**.

Brio Intelligence prompts you for the number of data columns. The number that appears by default in the dialog box is an estimate.

4 Type the number of columns and click **OK**.

Brio Intelligence 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

With Brio Intelligence, you can export data to other file formats for use with non-Brio Intelligence applications.

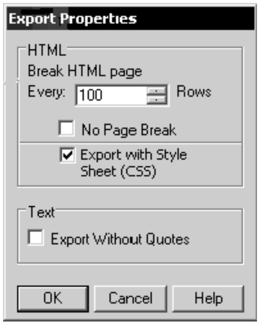
Export Properties

Before exporting data, use the Export Properties dialog box to specify export properties for data in the Results, Table, Pivot, and OLAP Query sections.

To specify export properties:

1 Choose **Format**→**Export Properties**.

The Export Properties dialog box appears.



Export Properties

HTML

Break HTML page

Every: 100 Rows

☐ No Page Break

☑ Export with Style Sheet (CSS)

Text

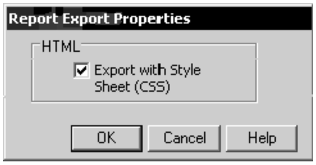
☐ Export Without Quotes

2. Select the desired export properties then click **OK**.

When exporting Results, Table, Pivot, or OLAP Query sections to HTML:

- Specify the number of rows to include on an HTML page before starting a new page (file). The default setting is 100 rows per HTML page.
- Specify whethet to export all the data to one HTML file. (Select No Page Break to export to one file.)
- Specify whether to create a Cascading Style Sheet file separate from the HTML file, or whether to embed the style sheet information in the HTML file itself . (Select Export with Style Sheet to embed the infomation.)

  When exporting information from the Report section to HTML, you can also specify whether to embed the Cascading Style Sheet in the HTML file, or whether to create it separately. The Export Properties dialog box for the Report section appears as follows:



☑ Export with Style Sheet (CSS)

- For data being 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.

Export 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 an EIS section to a JPEG file.

To export the current section to a file:

1. Choose **File**→**Export**→**Section**.

   The Export Section dialog box appears.

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 Choose a file format from the Save As Type drop-down list. Choose from:

- Excel (`.xls`)
- Lotus 1-2-3 (`.wks`)
- Tab delimited (`.txt`)
- HTML (`.htm`)
- Comma delimited (`.csv`)
- JPEG (`.jpg`) -
- PDF (`.pdf`)

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

Note Choose **Format**→**Export Properties** to set properties for files exported to text or HTML.

5 Click **Save** to export the data in the specified format.

You also can use the clipboard to cut and paste the data into another application.

If you are operating in a Macintosh environment, you can use Publish/Subscribe to export the Results set to an Edition file.

To export data with Publish/Subscribe *(Macintosh only):*

➤ Choose **Edit**→**Publish**.

The File Save dialog box prompts you to specify a name and path for the Edition file.

Export Document As Web Page

Use the Export Document As Web Page command to publish the contents of a Brio Intelligence document as a Web page. 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 appear in the Brio Intelligence document. Exportable document sections include: Results, Table, Pivot, Chart, 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, OLAPQuery, and EIS 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

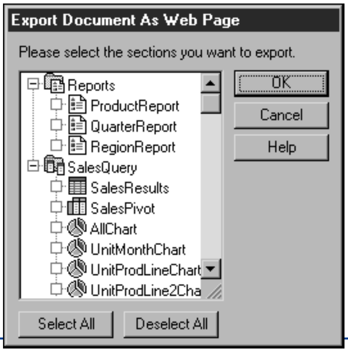
After these files are posted to a Web site, you can access the files individually or use the main HTML file to view the frameset, which lists the sections in the report. The exported Web pages include navigation buttons so you can scroll back and forth between pages, or jump to the beginning or end of 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 Brio Intelligence remain unchanged. However, we recommend that you use the Export As Web Page option.

To export a Brio Intelligence document as a Web page:

1 Choose **File**→**Export**→**Document As Web Page**.

The Export Document As Web Page dialog box appears.


Export Document As Web Page

Please select the sections you want to export.

2 Select the sections of the document that you want 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 appears.

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 Brio Intelligence (*.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, choose File→Open, browse to the location of the exported files, and open the *main* HTML page.

Browser and HTML Restrictions and Limitations

Browser and HTML restrictions may affect how graphics and formatting appear 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 appears 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. appears smaller in Brio Intelligence 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 do not appear 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 do not appear in reports in UNIX browsers.
- The right border of a table embedded in a report sometimes appears 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 appear with a certain font, it is because Brio Intelligence inserts a single blank into each empty cell. HTML requires a single value in each empty cell.
- HTML does not recognize vertical text.

Export To HTML Wizard

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

Note We suggest you use the Export Document As Web Page command rather than the HTML Wizard.

To use the HTML Wizard:

➤ Choose **Export→HTML Wizard**.

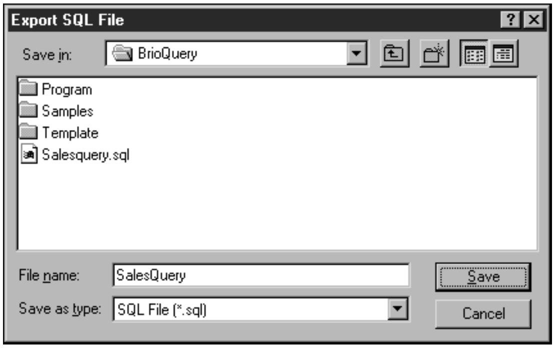
Export SQL

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

To export SQL:

1 Choose **File**→**Export**→**SQL**.

The Export SQL File dialog box appears.



Export SQL File

Save in: BrioQuery

Program
Samples
Template
Salesquery.sql

File name: SalesQuery

Save as type: SQL File (*.sql)

2 Specify the file name and location and click Save.

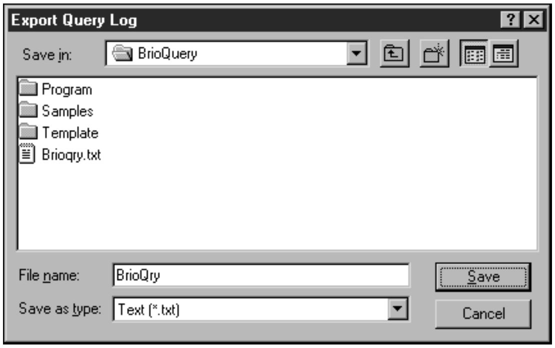
Export Query Log

When you process a query, Brio Intelligence 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 Choose **File**→**Export**→**Query Log**.

The Export Query Log dialog box appears.



Save in: BrioQuery

File name: BrioQry

Save as type: Text (*.txt)

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

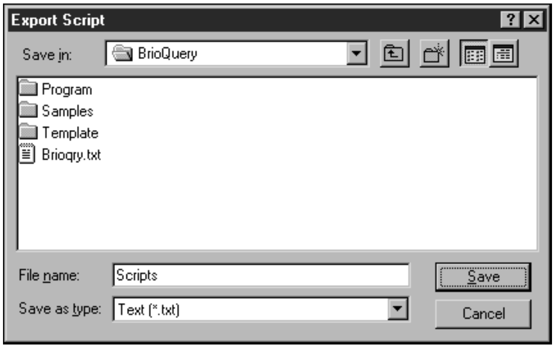
Export Script To Text File

Use the Export Script To Text File command to export JavaScript code and associated events contained in a BQY file into a text file (TXT). Brio Intelligence 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 Choose **File**→**Export**→**Script To Text File**.

The Export Script dialog box appears.



Save in: BrioQuery

File name: Scripts

Save as type: Text (*.txt)

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

Printing Documents

Brio Intelligence 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 onscreen.

Page Setup

Use the Page Setup command to specify the default printer for a Brio Intelligence document and to define default page properties for that printer.

➤ Choose **File**→**Page Setup**.

Print Preview

Use the Print Preview command to view onscreen a representation of the printed version of a finished report. The Print Preview command is available for all sections except EIS. 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:

Choose **File**→**Print Preview**.

A preview of the current section appears 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.

Choose **File**→**Print**. **[Ctrl+P]**

Emailing 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 Choose **File**→**Send**.

The Mail Document window appears.

2 Select the name of the document recipient.

You can send the document with or without the results. Sending a BQY 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 of reports created using the Report Designer. No further analysis is possible.

3 Select whether to copy other recipients or include additional remarks, then click **Send It**.

Quitting Brio Intelligence

To end your Brio Intelligence session:

➤ Choose **File**→**Exit**. **[Alt+F4]**

If documents remain open, Brio Intelligence prompts you to save changes to each document before it shuts down.

File Menu Command Reference

Table 3-1 provides a quick reference to the commands available on the File menu and list any related shortcuts. The commands listed are those available in Query and Results sections.

Table 3-1 File Menu Commands

Command	Description	Keyboard Shortcut
New	Opens the New File dialog box.	[Ctrl+N]
Open	Opens the Open File dialog box.	[Ctrl+O]
Close	Closes the active document.	[Ctrl+W]
Save	Saves the active document	[Ctrl+S]
Save As	Opens the Save As dialog box.	
Save Options	Allows you to specify save options for the active document such as saving result sets, compression, or password protection.	
Open From Repository	Opens the Open From Repository dialog box for the selected database, from which you choose a data model to use as the basis of a new document.	
Save To Repository	Saves data models to the repository for version-controlled distribution to networked Brio Intelligence users.	
Register For OnDemand Server	Register Brio Intelligence documents or repository data models to the OnDemand Server for use on the Web.	
Schedule	Schedules a document to the Broadcast Server for processing.	
Import Data File	Imports data stored in Microsoft Excel, tab-delimited, or comma-delimited file formats to the Results section for analysis and reporting.	
Export	Exports data to the selected format.	
Document Scripts	Opens the Script Editor for the selected object.	
Page Setup	Opens the Print Setup dialog box.	
Print Preview	Toggles Print Preview mode and the Print Preview menu. Displays a preview of the current section in the Content pane.	
Print	Prints the information displayed in the Content pane.	[Ctrl+P]
Send	Attaches a document to an email message.	
Exit	Shuts down Brio Intelligence.	[Alt+F4]

Using Edit Commands

The Brio Intelligence Edit menu contains standard editing commands. It also contains commands that allow you to work with document sections. (See "Working with Document Sections" on page 3-34 for detailed information on working with document sections.)

Table 3-2 provides a quick reference to the commands available on the Edit menu and lists any related shortcuts. The commands listed are those available in the Query and Results sections.

Table 3-2 Edit Menu Commands

Command	Description	Keyboard Shortcut	Shortcut Menu
Undo	Reverses the last command issued.	[Ctrl+Z]	
Redo	Re-applies the actions or commands on which you have used the Undo command.	[Ctrl+Y]	
Cut	Cuts the selected item.	[Ctrl+X]	✔
Copy	Copies the selected item.	[Ctrl+C]	
Paste	Paste the last copied item.	[Ctrl+V]	
Clear	Clears entry fields in dialog boxes.		
Select All	Select all items or elements, depending on the location of the cursor.	[Ctrl+A]	
Delete Section	Opens the Delete Section dialog box.		✔
Rename Section	Opens the Section Label dialog box.		✔
Duplicate Section	Duplicates the selected section and adds the copy to the Section pane with a sequenced number appended to the section name.		✔
Duplicatable	Makes the selected section duplicatable, which means it can be copied.		

Changing Workspace Views

The View menu allows you to toggle the display of Brio Intelligence interface elements, such as panes, toolbars, and so on. It also provides commands for working with sections. (See “Working with Document Sections” on page 3-34 for detailed information on working with document sections.)

Table 3-3 provides a quick reference to the commands available on the View menu and list any related shortcuts. The commands listed are those available in the Query and Results sections.

Table 3-3 View Menu Commands

Command	Description	Shortcut Menu
Section/Catalog	Toggles the display of the Section and Catalog panes.	
Section Title Bar	Toggles the display of the Section title bar.	
Toolbars	Toggles the display of the Standard, Format, and Section toolbars.	
Status Bar	Toggles the display of the Status bar.	
Console Window	Opens the Console window, which is used to display error messages and alert values generated by the JavaScript interpreter.	
Execution Window	Opens the Execution window, which is used to test script commands in the EIS section.	
Go To Section	Navigates to the section selected from the list of sections in the current document.	
Hide Section	Hides the active section.	
Unhide Section	Opens the Unhide Section dialog box which lists currently hidden sections.	
Query Log	Opens the Query Log dialog box, which displays the command language statement for the active query.	
Zoom	Changes the display of the active section to the selected zoom setting.	
Hide Request Item	Hides the selected Request item from view.	✔
Unhide Request Item	Opens the Unhide Columns dialog box, which lists currently hidden Request items.	✔

Table 3-3 View Menu Commands

Command	Description	Shortcut Menu
Custom SQL	Opens the Custom SQL window where you can edit the SQL statement for a query.	
Properties	Opens the Properties dialog box for the selected topic or topic item.	✔

Inserting Sections and Breaks

Use the commands on the Insert menu to insert new sections in your Brio Intelligence document. You can also insert page headers and footers for use when printing certain sections.

Table 3-4 provides a quick reference to the commands available on the Insert menu and list any related shortcuts. The commands listed are those available in the Query and Results sections.

Table 3-4 Insert Menu Commands

Command	Description
New Query	Inserts a new Query section.
New OLAP Query	Inserts a new OLAPQuery section.
New Table	Inserts a new Table section.
New Pivot	Inserts a new Pivot section.
New Chart	Inserts a new Chart section.
New Report	Inserts a new Report section.
New EIS	Inserts a new EIS section.
Page Header	Inserts a page header that is used when the section is printed.
Page Footer	Inserts a page footer that is used when the section is printed.

Note See "Working with Document Sections" on page 3-34 for detailed information on adding document sections and customizing the headers and footers in the document sections.

Formatting Text and Other Elements

Use the commands on the Format menu to change the formatting properties of text, numbers, borders, rows, columns, exported documents, and so on. Most of these commands can also be found on the Format toolbar (see page 3-4).

Table 3-5 provides a quick reference to the commands available on the Format menu and lists any related shortcuts. The commands listed are those available in the Query, Results, and OLAPQuery sections.

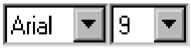


Table 3-5 Format Menu Commands

Command	Description	Keyboard Shortcut	Shortcut Menu
Font	Opens the Font page of the Properties dialog box.		✔
Style	Choose between Plain, Bold, Italics, Underline, Overline, Double Underline.		
Number	Opens the Number page of the Properties dialog box.		✔
Justify	Choose between Left, Center, Right and Top, Middle, Bottom.		✔
Column	By default, Results columns are evenly sized without regard to the length of data values. Numeric data that does not fit is replaced with pound signs (#). To manually resize a column, drag the right edge of the column to a new position.		
Auto-Size Width	Resizes the selected column to the width of the contents.	[Ctrl+E]	
Standard Width	Resizes the selected column to the standard column width.		
Row	Resizes all rows to the standard row height. To resize all rows in a table, drag the bottom edge of a row to a new position. All rows in the table are resized.		
Column Titles	Toggles the display of column titles.		

Table 3-5 Format Menu Commands *(Continued)*

Command	Description	Keyboard Shortcut	Shortcut Menu
Row Numbers	Toggles the display of row numbers. Row numbers are printed on reports, but are not copied to the clipboard or exported to a file.		
Text Wrap	Wraps text in a column.		✔
Suppress Duplicates	Suppresses duplicate values in a column. Use this feature if you want to display only the first instance of a duplicate value when individual database records include redundant information.		✔
Grid Lines	Opens the Gridlines page of the Properties dialog box.		
Border and Background	Opens the Border and Background page of the Properties dialog box.		
Spotlighter	Opens the Spotlight dialog box. Use to spotlight important values.		
Export Properties	Opens the Export Properties dialog box. Use to set the number of rows that should be included on an HTML page before the data breaks to another page, and to export data that does not contain any quote to a tab-delimited text file.		

☆ **Tip** To automatically size all columns so that the column width fits its contents, press [Ctrl+A] [Ctrl+E].

Note See "Working with Document Sections" on page 3-34 for detailed information on working with document sections.

Working with Document Sections

Working with document sections involves:

- Understanding Document Sections
- Adding Sections
- Viewing Sections
- Moving Between Sections
- Duplicating Sections
- Renaming Sections
- Adding Headers and Footers to Sections
- Deleting Sections

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 EIS 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.

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

Figure 3-1 shows some example document section displayed in the Section pane.

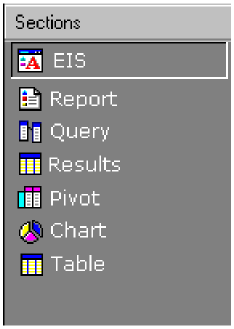


Figure 3-1 Document Sections in the Section Pane

Adding Sections

To insert a new section in your Brio Intelligence document:

➤ Choose **Insert**→**New *Section***.

For example, to insert a new Chart, choose Insert→Chart, to insert a new Table, choose Insert→Table.

Brio Intelligence 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 Brio Intelligence workspace. This allows you to concentrate on only those sections in which you want to work.

To hide a section:

1. In the Section pane, select the section that you want to hide.
2. Choose **View**→**Hide Section**.

To view a hidden section:

1. Choose **View→Unhide Section**.

   The Unhide Sections dialog box appears.

2. Select the **hidden section** that you want to view and click **OK**.

   The section appears 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.
- Choose View→Go To Section→*Section*.

Duplicating Sections

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

To make a section duplicatable:

➤ Select the section label in the Section pane and choose **Edit→Duplicatable**.

To duplicate a section:

➤ Select the section label in the Section pane and choose **Edit→Duplicate Section**.

Brio Intelligence 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

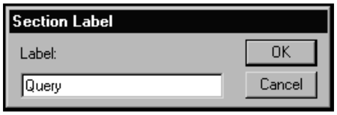
The first section that you create is given the default section name, for example, Query or Results. When you insert new sections of the same type as those that already exist, they are numbered sequentially, for example, Query2, Results2, and so on. To assign sections different or unique names based on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

To rename a section:

1. In the Section pane, select the section label.
2. Choose **Edit**→**Rename**.

   You can also click Rename Section on the Shortcut menu, or double-click the desired section.

   The Section Label dialog box appears.



Section Label

Label: Query

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.

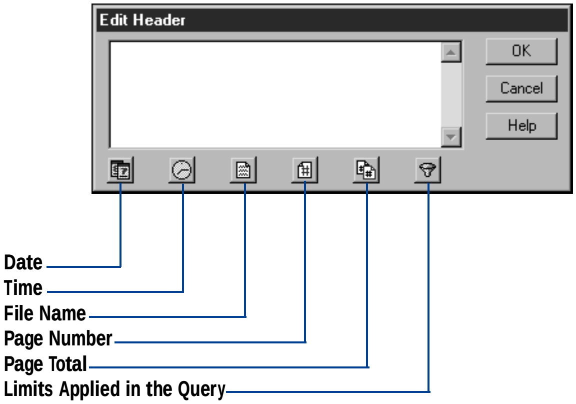
To add a header or footer:

1. Choose **File**→**Print Preview**.

   A preview of the current section appears in the Content pane.

2. Choose **Insert**→**Page Header** (or Page Footer).

   The Edit Header (or Edit Footer) dialog box appears.



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**.

Brio Intelligence adds the new header or footer to your report.

You can change the font properties and alignment of headers and footers, but you cannot add color.

To edit a header or footer:

➤ Double-click the header or footer you want to edit, make any desired changes, and then 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.

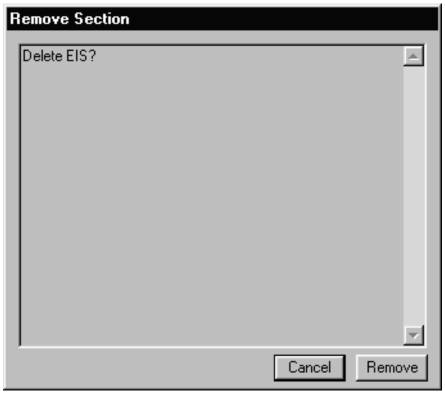
To delete a section:

1 In the Section pane, select the section label.

2 Choose **Edit→Delete Section**.

You can also select the section and click Delete Section on the Shortcut menu.

The Remove Section dialog box appears.



Delete EIS?

3 Click **Remove**.

Setting Brio Intelligence Options

The Tools menu provides commands that allow you to manage various Brio Intelligence options, such as job processing options, connections, default formats, and program options.

See the following sections for information on:

- Specifying Default Formats
- Selecting Program Options
- Customizing Menus
- Tools Menu Command Reference

Note See "Tools Menu Command Reference" on page 3-51 for an overview of all the Tools menu commands.

Specifying Default Formats

This section explains how to set up and change the way fonts, styles, numbers, currency values, and dates appear within the Brio Intelligence workspace.

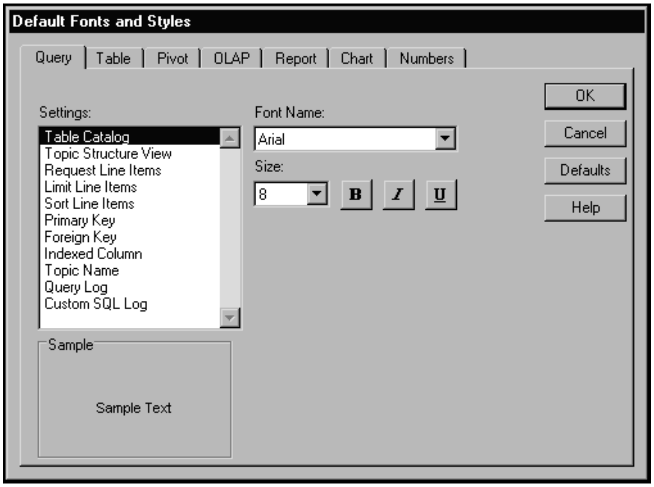
Default Fonts and Styles

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

To change default fonts and text settings:

1. Choose **Tools→Options→Default Formats**.

   The Default Fonts And Styles dialog box appears, with tabs that contain specific font settings for each section.



Font Name: Arial

Size: 8

2 Click the tab for the section whose fonts and styles you want to change.

3 Make the desired default 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 appear throughout Brio Intelligence, 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 3-6 lists the fields available on the Numbers page.

Table 3-6 Default Number Formats

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

Table 3-6 Default Number Formats

Field	Description
Month (For “Add Date Groups”)	Sets the default month format for the month used in 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 3-7 lists the numeric field options and definitions along with examples.

Table 3-7 Number Field Definitions

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

Table 3-7 Number Field Definitions *(Continued)*

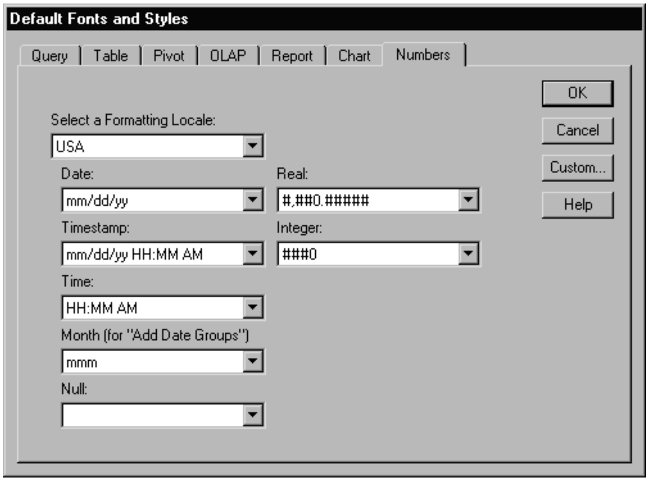
Option	Description	Examples
- /	Adds the respective character to date-coded values in the same position relative to variables.	Apply mm/dd/yy to show 06/23/99
HH MM SS	Displays hour, minute, and second in respective positions for date-coded information.	Apply HH:MM:SS to show 17:45:10
: AM PM	Adds the 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 number formats:

1 Choose **Tools**→**Options**→**Default Formats**.

The Default Fonts And Styles dialog box appears.

2 Select the **Numbers** tab.



Select a Formatting Locale: USA

Date: mm/dd/yy

Timestamp: mm/dd/yy HH:MM AM

Time: HH:MM AM

Month (for "Add Date Groups") mmm

Null:

Real: #,##0.######

Integer: ###0

3 Select the **country** associated with the default formats that you want to use.

4 Select a format for each data type.

5 If you are working with an imported file, select the Thousands, Decimal, or List separator that you want to apply for the format.

6 Click **OK**.

To add a new custom format:

1 Choose **Tools**→**Options**→**Default Formats**.

The Default Fonts And Styles dialog box appears.

2 Select the **Numbers** tab and click **Custom**.

3 Type the new format in the **Format String** field and click **OK**.

The format is added to the default options list.

Selecting Program Options

This section explains 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

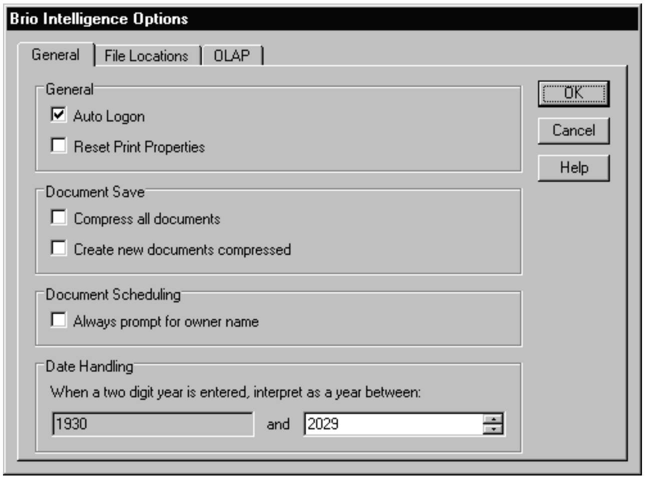
Use the General tab to globally enable or disable specific operating functions.

To select general options:

1 Choose **Tools**→**Options**→**Program Options**.

The Brio Intelligence Options dialog box appears.

2 Click the **General** tab.



General

☑ Auto Logon

□ Reset Print Properties

Document Save

□ Compress all documents

□ Create new documents compressed

Document Scheduling

□ Always prompt for owner name

Date Handling

When a two digit year is entered, interpret as a year between:

1930 and 2029

- Auto Logon – Maintains an existing connection whenever you create a new document. If you are currently logged on, Brio Intelligence 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.
- Compress All Documents – Specifies that Brio Intelligence save 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→Save As and changing the Save As type.
- Create New Documents Compressed – Specifies that Brio Intelligence compress only new documents.
- Always Prompt For Owner Name – Requires Brio Intelligence to prompt for an owner name of job scheduling repository tables whenever you schedule a document. Enable this feature if you schedule documents to more than one repository.

- 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, Brio Intelligence 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, Brio Intelligence 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 3/12/96, Brio Intelligence accepts the date as March 12, 1996.

  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 *19*99. 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 2001 instead of 01.

File Locations

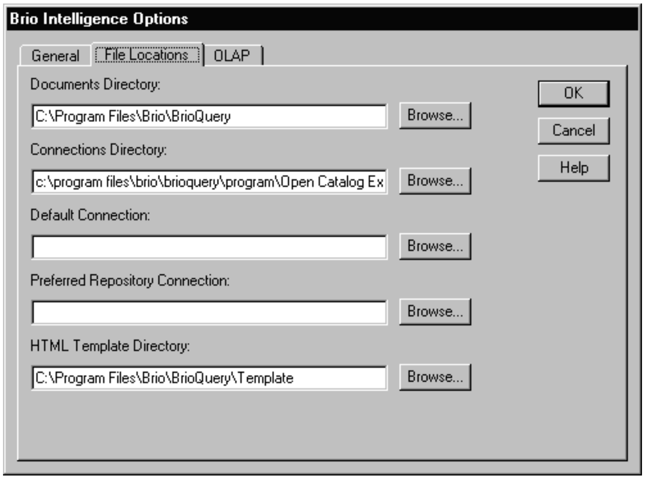
Use the File Locations tab to specify the default locations for Brio Intelligence documents and other necessary files.

To specify default file locations:

1 Choose **Tools**→**Options**→**Program Options**.

The Brio Intelligence Options dialog box appears.

2 Click the **File Locations** tab.



Documents Directory: C:\Program Files\Brio\BrioQuery

Connections Directory: c:\program files\brio\brioquery\program\Open Catalog Ex

Default Connection:

Preferred Repository Connection:

HTML Template Directory: C:\Program Files\Brio\BrioQuery\Template

- Documents Directory – The default directory where you want to save Brio Intelligence documents when the Save File dialog box appears. Brio Intelligence documents are saved in the default directory with the `.bqy` extension on Windows, and as Brio files on Macintosh.
- Connections Directory – A directory that contains the OCEs _used to connect to databases. The default Connections directory is `c:\brioqry\Open Catalog`.
- Default Connection – The OCE Brio Intelligence uses when no connection is specified, such as when you click the connection icon in a new document file.
- Preferred Repository Connection – The repository connection file you want the user to see in the Open Repository Connection drop-down list.
- HTML Template Directory – The directory of HTML templates used with the HTML Export Wizard.

OLAP Options

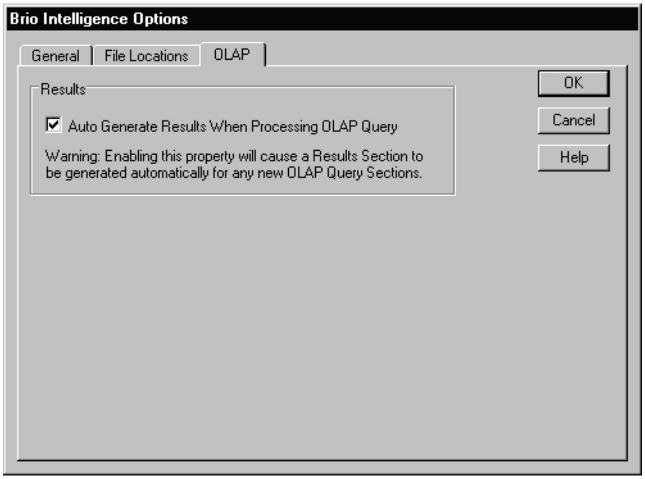
Use the OLAP tab to have Brio Intelligence automatically create a Results section when you click **Process** to process your OLAPQuery

To select OLAP options:

1 Choose **Tools**→**Options**→**Program Options**.

The Brio Intelligence Options dialog box appears.

2 Click the **OLAP** tab.



Results

☑ Auto Generate Results When Processing OLAP Query

Warning: Enabling this property will cause a Results Section to be generated automatically for any new OLAP Query Sections.

- Auto-Generate Results Section When Processing an OLAP Query – Automatically create a Results section for any future OLAPQuery section when that OLAPQuery section is first processed.

Customizing Menus

You can use JavaScript to customize Brio Intelligence menus. Add scripted menu items to the menu bar to:

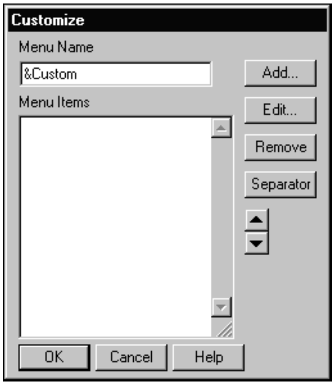
- Run commonly used scripts
- Launch separate applications
- Export sections to a different file format with a single click

Note Since version 6.0 of Brio Intelligence, JavaScript is used as the script-editing tool instead of the Brio Intelligence 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 Choose **Tools**→**Customize**.

The Customize dialog box appears.

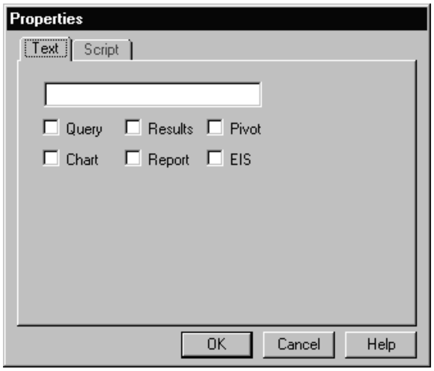


&Custom

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 appears and displays the Text page.



4 Type a **name** for the menu item.

Select one or more check boxes to indicate the sections where the menu item should appear.

5 Click 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.

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

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

7 Click **OK** when you are finished to close the Customize dialog box.

8 If you want 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 Macintosh, the `bqtools.ini` file is located in the `System:Preferences` folder.
- For UNIX, the `.bqtools.ini` file is located in the `user home` directory.

Tools Menu Command Reference

Table 3-8 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 3-8 Tools Menu Commands

Command	Description	Keyboard Shortcut
Process Query	Processes the current query, all queries in the document, or a customized selection of queries.	
Connection	Allows you to select, log on, log out, modify, or create a connection.	
Connections Manager	Opens the Connections Manager dialog box.	[F11]
Save Connection	Saves a connection file with a Brio Intelligence document.	
Administer Repository	Opens the Administer Repository dialog box, where you can modify object descriptions or groups.	
View Job List	Shows run time and status details for each job submitted to a job repository.	
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.	

What Next

This chapter provides a reference for many of the menu commands used throughout Brio Intelligence. Now you are ready to learn core query techniques. Proceed to Chapter 4, “Querying Relational Databases,” which explains how to build a relational query by downloading prebuilt data models from a repository, and how to create your own data model and use it to build a relational query.

4

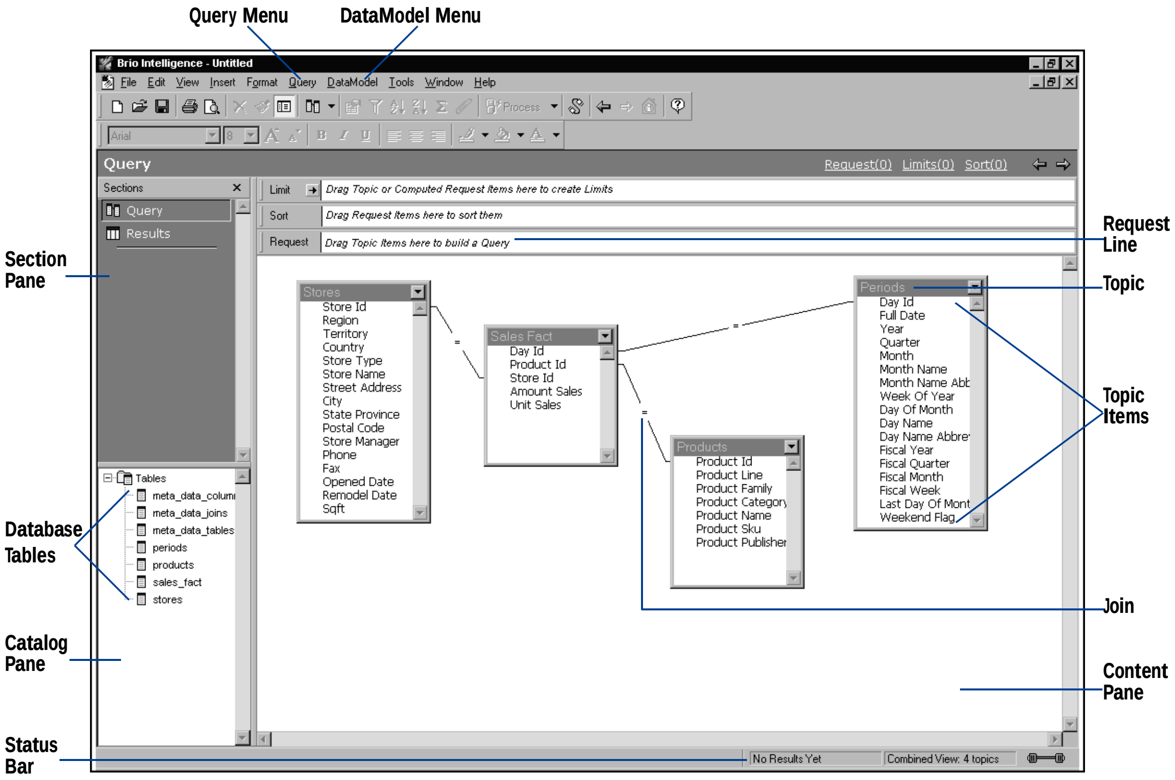
Querying Relational Databases

This chapter explains how to use Brio Intelligence to connect to and query a relational database. It contains:

- About the Query Section
- Building Queries
- Processing Queries
- Saving Queries
- Building Subqueries
- Working with Query Section Data
- Query Menu Command Reference

About the Query Section

The Query section is the foundation of any Brio Intelligence document and is automatically created (along with its complementary Results section) whenever you open a new Brio Intelligence document. It provides an intuitive interface to a relational database server .



A Brio Intelligence document 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 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 Brio Intelligence tool, 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 appear 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:

➤ Choose **DataModel→Table Catalog**.
[F9]

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, Brio Intelligence prompts you for your user name and password.

To create a data model:

➤ In the Catalog pane, select a table from the Table catalog, and choose **Query→Add Request Item(s)**.

Note For detailed information on data models, see Chapter 2, “Data Modeling” in the *Brio Intelligence 6.6 Administrator’s Guide*.

Building Queries

You build queries by adding topics from the data model in the Content pane to the Request line.

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 choose Query→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 is processed and displayed.

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 [Del].

If you have not yet processed the query, Brio Intelligence removes the item from the Request line. If you have previously processed the query, you are informed that your results no longer match the items requested.

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

Hiding Request Items

You can hide items that appear 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→Hide Request Items.**

To show a hidden requested item:

1 Complete one of the following actions:

- Click in the Request line and click **Unhide** on the shortcut menu.
- Choose **View→Unhide Request Items.**

The Unhide Columns dialog box appears.

2 Select the items you want to view and click **OK**.

When you have identified the items you want to include in your query, you can perform a number of other operations before processing the query. You can add limits or computed items to the Request line, or you can use a Request line item to specify a sort order.

For more information about limits, see Chapter 7, "Using Limits." For more information about computed items, see Chapter 8, "Working with Computed Items." For more information about sorts, see Chapter 9, "Applying Sorts."

Processing Queries

After you build your query and apply limits, 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, use one of the following options:

- Click the Process button on the standard toolbar. (Click the right-arrow to select a process option.)
- Choose Tools→Process Query and select the desired process option.

Since a document 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.
- **Process Custom –** Opens the Process Custom dialog box so that you can indicate which queries to process by selecting a query's check box.

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

Saving Queries

After you have processed a query, your data is available until you close the document. Saving your document saves the current formatting and layout of all Brio Intelligence sections in the document.

To save your query, choose File→Save. For a complete discussion of save options, see “Saving Documents” on page 3-11 and “Saving Results Sets” on page 5-12.

Cancelling Queries

To cancel a query, use one of the following options:

- For Windows, press and hold [Alt+End] until the query is cancelled.
- On the Macintosh, press [Command+.] (period), or choose Results→**Cancel Process**.
- For Solaris, press and hold the [Ctrl] and [End] keys at the same time until the query is cancelled.

Note If the database server has to perform computations before sending the data across the network, you can cancel the query only if you are using an asynchronous API.

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 then filters its rows based on the rows retrieved by the subquery. Brio Intelligence uses two types of subqueries: regular and correlated.

Regular Subqueries

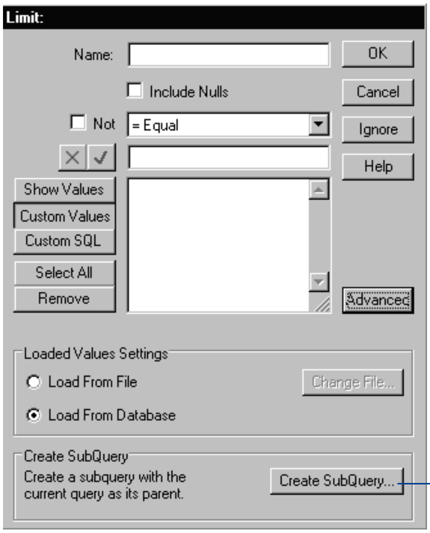
A regular subquery executes the inner and outer queries once and returns the values from the inner query to the outer query. For example, you might need to find out who sold more than the average of all sales representatives in April. You first use a subquery to define what was the average sales amount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

To build a subquery:

1. Select an existing parent query or build a parent query.
2. Use one of the following options to select an item on which you want to set a limit:
   - Double-click the topic item
   - Drag the item from the Content pane to the Limit line
   - Right-click and item and select **Limit** on the shortcut menu.

   The Limit dialog box appears.
3. Click **Advanced**.

   The Advanced button toggles the Create Subquery button.



□ Include Nulls

□ Not = Equal

○ Load From File

● Load From Database

Create a subquery with the current query as its parent.

Click here to create a subquery.

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 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 limits, set a limit 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.

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 an existing parent query or build a parent query.

2 Use one of the following options to select an item on which you want to set a limit:

- Double-click the topic item
- Drag the item from the Content pane to the Limit line
- Right-click and item and select Limit on the shortcut menu.

The Limit dialog box appears.

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 limits, set a limit inside a subquery as a variable, and define data functions and computations as needed.

6 If the parent query you want to work with does not appear in the Catalog pane, choose **Show Queries** from the shortcut menu in the Section pane.

7 Drag the parent query into the Content pane.

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

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

The Select Correlation Column dialog box appears and displays 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 you want to join the subquery topic item and click **OK**.

The topic item is added to the Limit 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.

Working with Query Section Data

You can further enhance the data requested in the Query section by working with the data. Working with Query Section Data involves:

- Processing Results to a Database Table
- Estimating Query Size
- Displaying Database Remarks
- Preaggregating Data Using Functions
- Appending Queries
- Using Local Results
- Using Stored Procedures
- Setting Query Properties

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

Note See "Query Menu Command Reference" on page 4-24 for a complete list of the commands available on the Query menu.

Processing Results to a Database Table

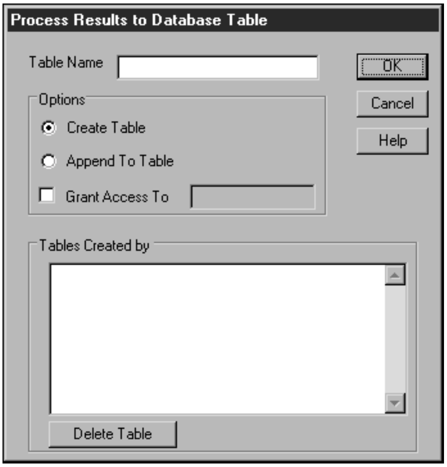
Instead of retrieving data to the Results section, you can instruct Brio Intelligence to create a table in the database to store your results set. Items on the Request line become the column headings of the new table, and you can append new columns to the table and query it as needed.

Note The connection file and database to which you are connected determine whether or not you can use this feature. You must have *Create* and *Insert* privileges on the database in order to process results to a database table.

To process results to a database table:

1 Choose **Query→Process Results To Table**.

The Process Results To Database Table dialog box appears.



2 Specify the information requested.

- Table Name – 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 – Appends Request items as new columns to an existing 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 limited 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 Choose **DataModel→Table Catalog**. **[F9]**

The Table catalog expands in the Catalog pane.

2 Choose **Refresh** on the shortcut menu.

The table you created appears in the list of database tables.

Brio Intelligence tracks tables you have created 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 Choose **Query→Process Results To Table**.

The Process Results to Database Table dialog box appears. Tables created under your user ID appear 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:

➤ Choose **Query→Estimate Query Size**. **[Ctrl+I]**

Brio Intelligence queries the database and counts the number of records that will be retrieved 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:

- Choose Query→Show Remarks.
- Click Show Remarks on the shortcut menu.

Preaggregating Data Using Functions

Depending on how you plan to view your data, you can choose 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.

Note Preaggregating data in your query assumes that you have a clear idea of the data you want to look at and a good conceptual understanding of relational databases. If you are unsure that you want to preaggregate, process the query without applying data functions. If you find that it would be better to preaggregate, you can always 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. Table 4-1 lists the prebuilt data functions that you can apply to items in the Request line.

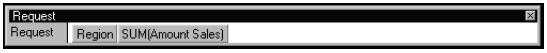
Table 4-1 Query Section Data Functions

Data Function	Description
None	Returns unaggregated values as stored in the database. This is the default option in the Query section.
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.
Count Distinct	Returns the number of distinct values in a column. This function is not supported by all database servers.
Standard Deviation	Returns standard deviation of values. This function is not supported by all database servers.
Variance	Returns variance of values. This function is available through Oracle servers only.
Weight	Use for computing weighted items in pivot tables. See *Data Analysis and Reporting with Brio Intelligence 6.6* for more information.

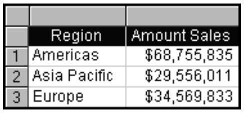
When using data functions, remember that with the exception of counts, data functions are applied almost entirely to numeric data items and the results are computed with respect to dimensional, nonnumeric items on the Request line, such as name and date items as in the following examples.

Example 1

Query 1 includes only items State and Units_Delivered. The data function Sum is applied and the data returned consists of one row for each state with an aggregate sum for that state in the Units column.



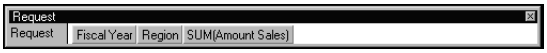
Request: Region | SUM(Amount Sales)



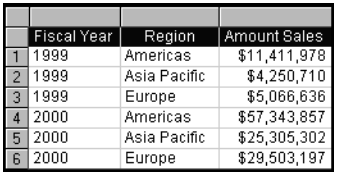
	Region	Amount Sales
1	Americas	$68,755,835
2	Asia Pacific	$29,556,011
3	Europe	$34,569,833

Example 2

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.



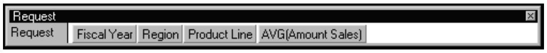
Request: Fiscal Year | Region | SUM(Amount Sales)



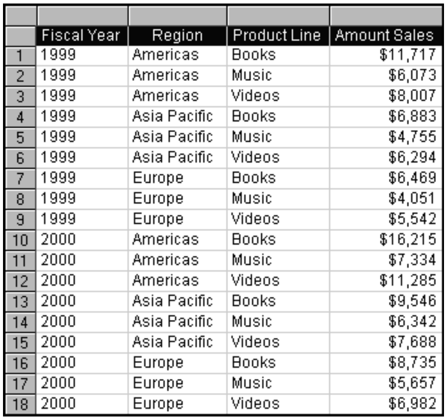
	Fiscal Year	Region	Amount Sales
1	1999	Americas	$11,411,978
2	1999	Asia Pacific	$4,250,710
3	1999	Europe	$5,066,636
4	2000	Americas	$57,343,857
5	2000	Asia Pacific	$25,305,302
6	2000	Europe	$29,503,197

Example 3

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: 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 Choose **Query**→**Data Function** and select the desired function. (See Table 4-1 for a list of the available data functions.)

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 whose data function you want to remove.

2 Choose **Query**→**Data Function**→**None**.

Appending Queries

When you need to view and merge multiple queries in a combined Results set, Brio Intelligence provides four query operators that allow you to merge two or more separate queries. The operators and their functions are:

- 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.”)

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, then 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 limits, data functions, and computations to the query as needed.

3 Choose **Query**→**Append Query**.

An Operator drop-down list and a second query tab is added below the Request, Limit, 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 you want to use from the Operator drop-down list.

6 Click **Process**.

You can have Brio Intelligence automatically generate the join path required by the context of the query by using the automatic join path feature (Brio Intelligence only). This feature eliminates the need for you to predefine any join paths, since Brio Intelligence 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 a Brio Intelligence document.

To use local results:

➤ Click anywhere in the Catalog pane and choose **Local Results** on the shortcut menu.

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

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:

- Limits, 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 limit
  - ❑ Time limit
  - ❑ Auto-process
  - ❑ Custom Group by
- Query limits on Local Results Topic Items
- More than one Limit Local Join
- Limit 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.

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. Brio Intelligence treats stored procedures as locked standard queries and does not allow you to modify the procedures.

You can use Brio Intelligence 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**. Brio Intelligence 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 OCE must specify the database as "ODBC" rather than "Oracle" to work properly.

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. Brio Intelligence 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 Choose **Query**→**Stored Procedures**.

The Stored Procedures dialog box appears.

2 Choose 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 limits. 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 Brio Intelligence adds items to the Request line.

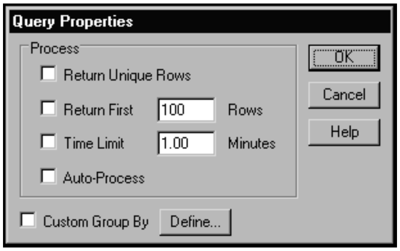
Setting Query Properties

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 Choose **Query**→**Query Options**.

The Query Properties dialog box appears.



Query Properties

Process

□ Return Unique Rows

□ Return First 100 Rows

□ Time Limit 1.00 Minutes

□ Auto-Process

□ Custom Group By

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

- Return Unique Rows – Eliminates duplicate rows from the data set retrieved by the query.
- Return First ___ Rows – Limits the number of database rows retrieved to the number entered.
- Time Limit ___ Minutes – Limits the amount of time the query is allowed to run to the number entered. Seconds are entered as a decimal number. Time limits 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.

Query Menu Command Reference

Table 4-2 provides a quick reference to the commands available on the Query menu and lists any related shortcuts.

Table 4-2 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 Limit(s)	Allows you to create a limit 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 existing topic item.		✔
Data Functions	Applies a prebuilt data function to the selected Request item.		✔
Variable Limit	Designates the selected Limit item as variable, which causes Brio Intelligence to prompt the user for limit values when the query is processed.		✔
Customize Limit	Allows you to control access to the features on the Limit 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 limits, and so on.		

What Next

This chapter taught you about the primary building block—the Query section. Now that you have built and processed a query, proceed to Chapter 5, “Working with Query Results,” to learn how to work with the results of your query.

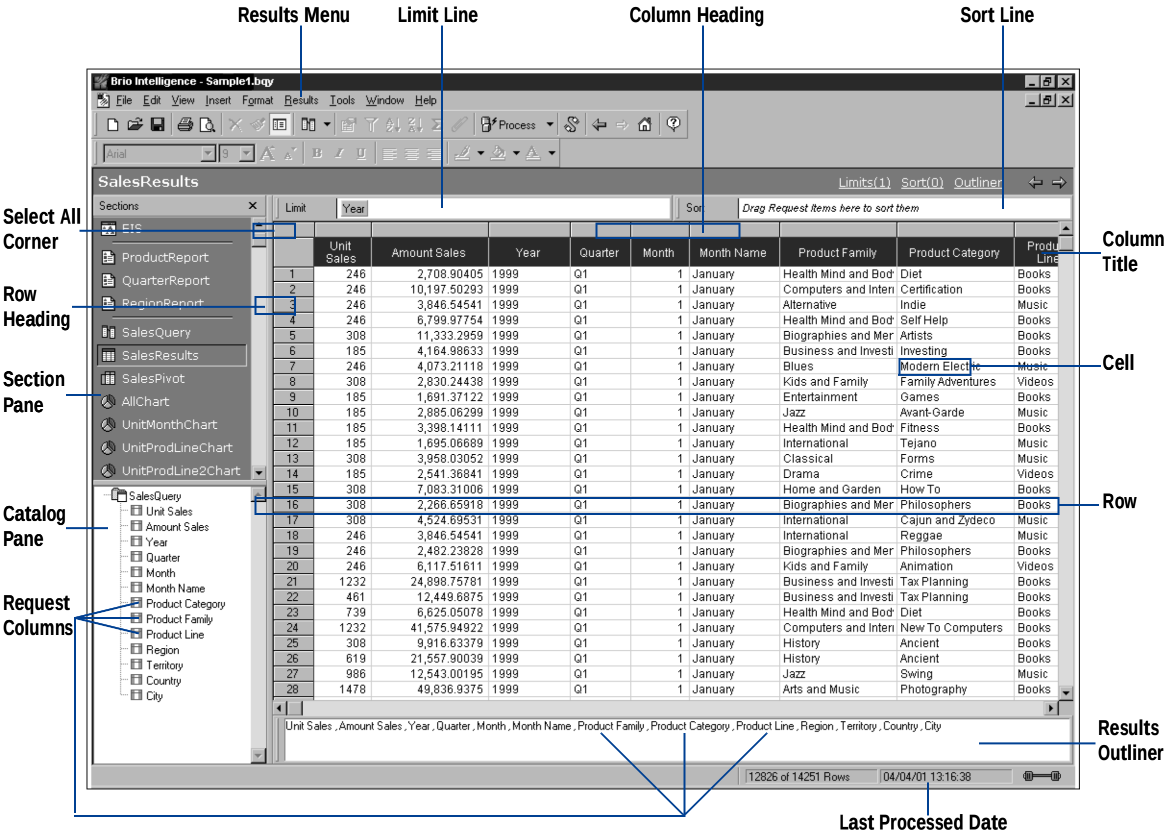
5 Working with Query Results

This chapter explains how to work with the results sets obtained from your relational database query or data import. It contains:

- About the Results Section
- Enhancing Your Results Set
- Working with the Results Table
- Saving Results Sets
- Exporting Result Sets
- Results Menu Command Reference

About the Results Section

When you process a query or import data, Brio Intelligence 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 appears 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).



	Unit Sales	Amount Sales	Year	Quarter	Month	Month Name	Product Family	Product Category	Produ Line
1	246	2,708.90405	1999	Q1	1	January	Health Mind and Bod'	Diet	Books
2	246	10,197.50293	1999	Q1	1	January	Computers and Inter	Certification	Books
3	246	3,846.54541	1999	Q1	1	January	Alternative	Indie	Music
4	246	6,799.97754	1999	Q1	1	January	Health Mind and Bod'	Self Help	Books
5	308	11,333.2959	1999	Q1	1	January	Biographies and Mer	Artists	Books
6	185	4,164.98633	1999	Q1	1	January	Business and Investi	Investing	Books
7	246	4,073.21118	1999	Q1	1	January	Blues	Modern Electric	Music
8	308	2,830.24438	1999	Q1	1	January	Kids and Family	Family Adventures	Videos
9	185	1,691.37122	1999	Q1	1	January	Entertainment	Games	Books
10	185	2,885.06299	1999	Q1	1	January	Jazz	Avant-Garde	Music
11	185	3,398.14111	1999	Q1	1	January	Health Mind and Bod'	Fitness	Books
12	185	1,695.06689	1999	Q1	1	January	International	Tejano	Music
13	308	3,958.03052	1999	Q1	1	January	Classical	Forms	Music
14	185	2,541.36841	1999	Q1	1	January	Drama	Crime	Videos
15	308	7,083.31006	1999	Q1	1	January	Home and Garden	How To	Books
16	308	2,266.65918	1999	Q1	1	January	Biographies and Mer	Philosophers	Books
17	308	4,524.69531	1999	Q1	1	January	International	Cajun and Zydeco	Music
18	246	3,846.54541	1999	Q1	1	January	International	Reggae	Music
19	246	2,482.23828	1999	Q1	1	January	Biographies and Mer	Philosophers	Books
20	246	6,117.51611	1999	Q1	1	January	Kids and Family	Animation	Videos
21	1232	24,898.75781	1999	Q1	1	January	Business and Investi	Tax Planning	Books
22	461	12,449.6875	1999	Q1	1	January	Business and Investi	Tax Planning	Books
23	739	6,625.05078	1999	Q1	1	January	Health Mind and Bod'	Diet	Books
24	1232	41,575.94922	1999	Q1	1	January	Computers and Inter	New To Computers	Books
25	308	9,916.63379	1999	Q1	1	January	History	Ancient	Books
26	619	21,557.90039	1999	Q1	1	January	History	Ancient	Books
27	986	12,543.00195	1999	Q1	1	January	Jazz	Swing	Music
28	1478	49,836.9375	1999	Q1	1	January	Arts and Music	Photography	Books

Unit Sales , Amount Sales , Year , Quarter , Month , Month Name , Product Family , Product Category , Product Line , Region , Territory , Country , City

12826 of 14251 Rows 04/04/01 13:16:38

Use the Results section to:

- Verify that your query returned the correct information.
- Refine and extend the data set by applying limit 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 Brio Intelligence 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 a 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.

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 limits 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, it is best to 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 limits, sorts, and computations.

Enhancing your results set involves:

- Limiting Results
- Sorting Results Data
- Adding Computed Items to Results
- Applying Data Functions to Results
- Adding Grouping Columns
- Automatically Adding Columns
- Breaking Out Dates

Limiting Results

Local limits applied in the Results section enable you to temporarily screen out portions of data for reporting purposes, without eliminating then from the data set.

Local limits are discussed in “Server versus Local Limit Processing” on page 7-2.

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 9, “Applying Sorts.”

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 Compute Item command enables you to build equations to compute totals, or to apply functions to existing values. Computations are performed on the desktop by Brio Intelligence 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.

In the Results section, reference items are limited to the items that appear on the Request line of the original query. Also, the scalar functions used to compute items are provided by Brio Intelligence, rather than the RDBMS.

For more information about computed items, see Chapter 8, “Working with Computed Items.”

Applying Data Functions to Results

In the Results section, you can only use data functions 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. 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. Table 5-1 lists the data functions that you can use with break totals and grand totals.

Table 5-1 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 Column Totals

To calculate a column total:

Σ ➤ Select the column to be totaled and click the summation icon on the Standard toolbar.

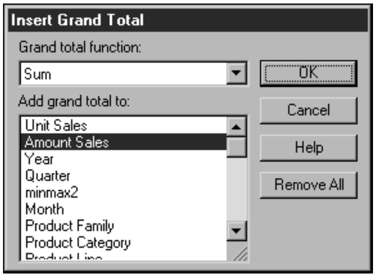
Brio Intelligence adds a row labelled Total to the bottom of the table and displays the total as the last entry in the selected column.

Inserting Grand Totals

To apply a grand total to a column using a data function:

1 Select a column and choose **Results**→**Grand Total**.

The Insert Grand Total dialog box appears.



Insert Grand Total

Grand total function: Sum

Add grand total to: Amount Sales

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**.

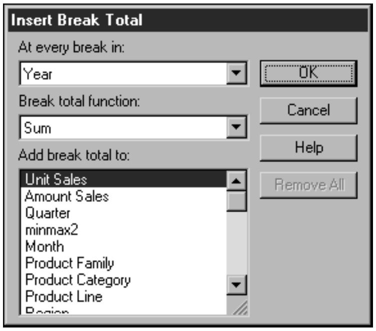
The total and any subtotals in the column are computed to reflect the new data function.

Inserting Break Totals

To apply a break total (subtotal):

1 Select a column and choose **Results**→**Break Total**.

The Insert Break Total dialog box appears.



Insert Break Total

At every break in: Year

Break total function: Sum

Add break total to:

Unit Sales
Amount Sales
Quarter
minmax2
Month
Product Family
Product Category
Product Line
Region

2 Select a **break column** from the At Every Break drop-down list.

3 Select the **data function** you want to apply from the Break Total Function drop-down list.

4 Select one or more columns on which to display the break total and click **OK**.

Adding Grouping Columns

Grouping columns, like computed items, create new data in your results set by grouping data from an already existing column. You can use grouping columns to consolidate nonnumeric 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 Results section and are available for use in report sections.

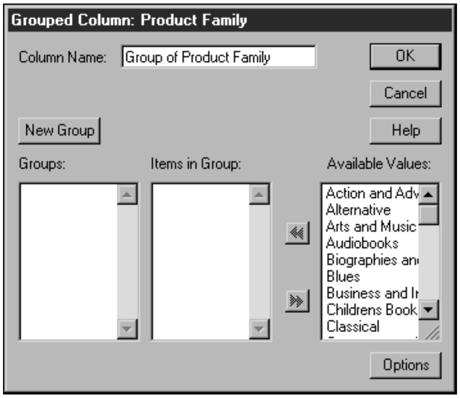
For example, your company sales database may contain the items: State, Sales Region, and Country, which allow you to aggregate data on different levels in reports. However, suppose you are looking to track sales by subregion, or you want to see data for one state versus an average for all other states 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 grouping column.

2 Choose **Results**→**Add Grouping Column**.

The Grouped Column dialog box appears. Use the column values to build the grouping categories for the new item.



Grouped Column: Product Family

Column Name: Group of Product Family

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, and then 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.
- Specify options for ungrouped values as follows:
  - ❑ **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.
  - ❑ **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 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.

You can modify a grouping column to change the group structure.

To modify a grouped column:

➤ Select the grouped column and choose **Results**→**Modify Column**.

Automatically Adding Columns

By default, Brio Intelligence retrieves data to your desktop and displays it in the Results section as columns. You can manually add request items by having Brio Intelligence return an empty Results set. This allows you to add columns as you need them.

This feature allows you to display only the rows with which you want to work. You can still sort or create limits using columns not displayed in the Results section.

To toggle AutoAdd columns:

➤ Choose **Results**→**AutoAdd Columns**.

- If the AutoAdd Columns feature is selected, then all requested items are displayed in columns.
- If the AutoAdd Columns feature is not selected, then 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. Choose **Results**→**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

Brio Intelligence 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 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 Outliner):

Choose **Results**→**Remove**. **[Del]**

If an item is removed from the Content pane, it is completely removed from the Outliner 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. For more information on formatting options, see “See “Working with Document Sections” on page 3-34 for detailed information on adding document sections and customizing the headers and footers in the document sections.” on page 3-31.

Saving Results Sets

After you have processed a query, your data is available until you close the document. Saving your document saves the current formatting and layout of all Brio Intelligence sections in the document.

You have the option to save the results set with the document. You also have the option to save any computed column expressions as a *snapshot*. Your decision in this selection depends largely on how you want to use the information in the document, and on what information needs to be recalculated.

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.

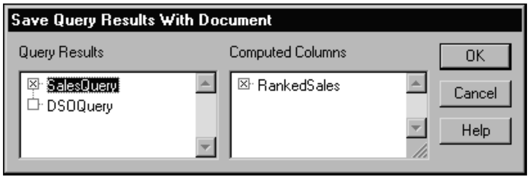
Note If you intend to work with a document which includes a Report Designer section, you *must* save your results with the document. If you do not save results with the document, the Report Designer section is not available.

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.

To save results with your document:

1 Choose **File**→**Save Options**→**Save Query Results With Document**.

The Save Query With Results Document dialog box appears and displays all of the query sections contained in your document.



Save Query Results With Document

Query Results
☒ SalesQuery
☐ DSOQuery

Computed Columns
☒ RankedSales

2 Select the check box for the query results you want to save and decide whether to save computed columns as snapshots, then click OK.

The query results and snapshots for computed columns that you selected 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 shapshots, but they take less time to open.

If you want 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 file 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.

Table 5-2 lists the selection options and effects for saving query results and snapshots of computed columns with documents.

Table 5-2 Effects of Save Query Results with Document Options

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

Note: You cannot choose to save computed columns as snapshots unless you first choose to 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 from Brio Intelligence, the most common being into file formats such as Excel or Lotus.

Brio Intelligence 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 datacube and is preaggregated. Scripts created by the Brio Intelligence JavaScript engine can be saved to a text file.

Export options are discussed in detail in “Exporting Data” on page 3-18. 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-delimted text files.

Results Menu Command Reference

Table 5-3 provides a quick reference to the commands available on the Results menu and lists any related shortcuts.

Table 5-3 Results Menu Commands

Command	Description	Keyboard Shortcut	Shortcut Menu
Limit	Opens the Limit 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.		✔

Table 5-3 Results Menu Commands *(Continued)*

Command	Description	Keyboard Shortcut	Shortcut Menu
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 Outliner item).	[Del]	✔
Break Total	Opens the Insert Break Total dialog box.		✔
Σ 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 Column	Automatically adds columns in the Content pane for all requested items. If not selected, no columns appear in the Content pane. Turn this option off to manually add columns for requested items.		

What Next

This chapter described the features and tools used for working with results from a relational query. Proceed to Chapter 6, “Querying Multidimensional Databases,” to find out how to create OLAP queries.

For more information about the features used in the Query and Results section, read Chapter 7, “Using Limits,” Chapter 8, “Working with Computed Items,” and Chapter 9, “Applying Sorts.”

When you are finished formatting and fine-tuning your results, see *Data Analysis and Reporting with Brio Intelligence 6.6* for assistance with using the Brio Intelligence application’s powerful reporting and data analysis features.

6

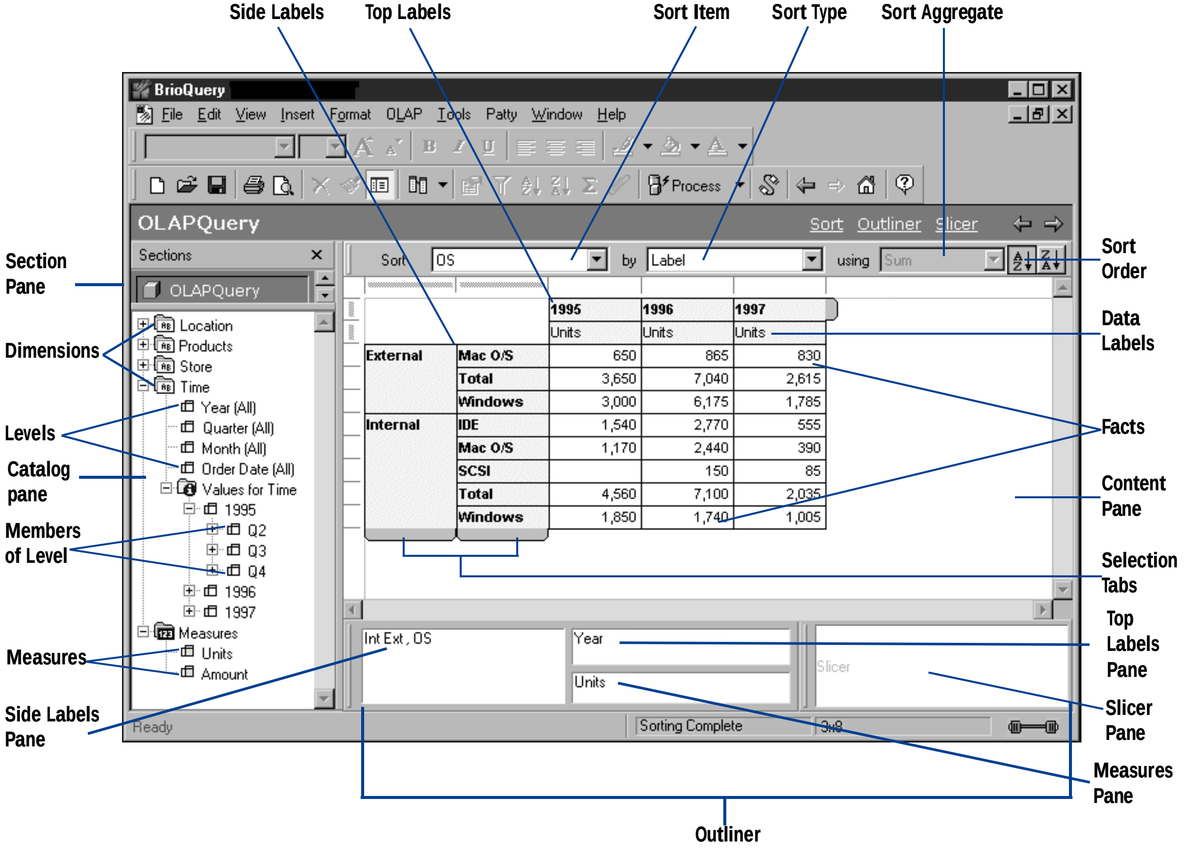
Querying Multidimensional Databases

This chapter explains how to use Brio Intelligence to connect to and query a multidimensional database. It contains:

- OLAPQuery Section
- Defining OLAPQuery Options
- Building OLAP Queries
- Refining OLAPQuery Data
- Processing OLAP Queries
- Applying Filters and Limits
- Changing Data Views
- Formatting OLAPQuery Items
- Drilling Through from a Multi-Dimensional Database to a Relational Database
- OLAP Menu Command Reference

OLAPQuery Section

Use the Brio Intelligence OLAPQuery section to query multidimensional or OLAP databases. When you connect to a multidimensional database, the document opens an OLAPQuery section.



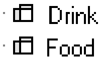
		1995	1996	1997
		Units	Units	Units
External	Mac O/S	650	865	830
	Total	3,650	7,040	2,615
	Windows	3,000	6,175	1,785
Internal	IDE	1,540	2,770	555
	Mac O/S	1,170	2,440	390
	SCSI		150	85
	Total	4,560	7,100	2,035
	Windows	1,850	1,740	1,005

The user interface is similar to the Brio Intelligence 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.)

Level (All)

- **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.



Drink
Food

- **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 Outliner.

- **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

  A property can be dragged into the Outliner with its corresponding level, but cannot be dragged into the Slicer pane, the Measures pane, or have a filter applied 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.** Brio Intelligence 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, Brio Intelligence creates an *Attribute Calculation* Dimension. An Attribute Calculation Dimension appears 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 Outliner.

Defining OLAPQuery Options

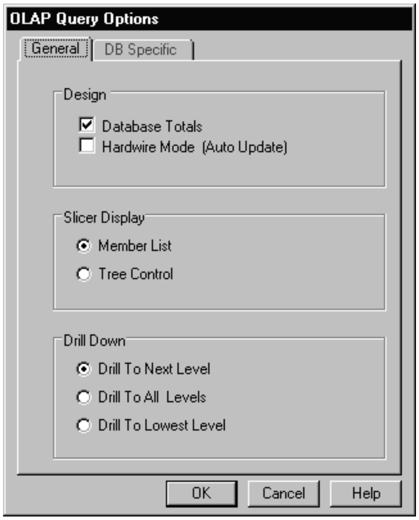
Brio Intelligence allows you to set options that control various properties in the OLAPQuery section. You can define both general and database-specific options.

To define OLAPQuery options:

➤ Choose **OLAP→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 appears as follows:



Design

☑ Database Totals

□ Hardwire Mode (Auto Update)

Slicer Display

● Member List

○ Tree Control

Drill Down

● Drill To Next Level

○ Drill To All Levels

○ Drill To Lowest Level

Table 6-1 General OLAP Query Options

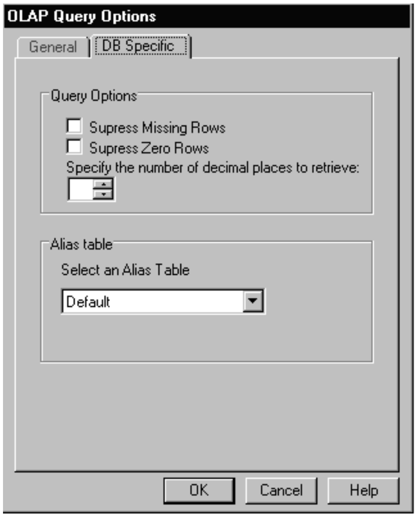
Design Options	Defines how Brio Intelligence retrieves database totals and requeries the database. ■ **Database Totals** – Retrieves database totals when the query is processed. ■ **Hardwire Mode** – Defines whether to requery the database when changes are made in the Outliner. □ If you select Hardwire mode, Brio Intelligence automatically requeries the database every time you add an item to or remove an item from the Outliner and instantaneously retrieves the data. You do not have to click Process. □ If you do not select Hardwire mode, you must click Process to requery the database whenever you make a change.
Slicer Display	Defines how to display slicer values. ■ **Member List** – Selects from a list of all members at the same level as the previously selected member. ■ **Tree Control** – Displays parent-child slicer value relationships. With Tree Control, you can select multiple values from a dimension as long as your database supports this function. (Databases such as Hyperion Essbase and IBM DB2 OLAP allow this).
Drill Options	Defines what level of data is the next level displayed when you drill down in an OLAPQuery. ■ **Drill to Next Level** – Automatically displays data for the next level below the selected member. For example, in a dimension with levels of *Year*, *Quarter*, *Month*, and *Date*, double-clicking on a Year level member name automatically displays all the data for the Quarter level belonging to that year. ■ **Drill to All Levels** – Automatically displays all possible levels of data below the selected member. For example, in a dimension with levels of *Year*, *Quarter*, *Month*, and *Date*, double-clicking on a Year level member name automatically displays all the data for the Quarter, Month, and Date levels belonging to that year. ■ **Drill to Lowest Level** – Automatically displays data for *only* the lowest level belonging to the selected member (intermediate member levels are not shown). For example, in a dimension with levels of *Year*, *Quarter*, *Month*, and *Date*, double-clicking on a Year level member name automatically displays all the data for the Date level belonging to that year.

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 appears as follows:



Query Options

☐ Supress Missing Rows

☐ Supress Zero Rows

Specify the number of decimal places to retrieve:

Alias table

Select an Alias Table

Default

Table 6-2 Display Options for Essbase Databases

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.

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, Brio Intelligence 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 for OLAP

The tab for Microsoft/OLE DB for OLAP database appears as follows:

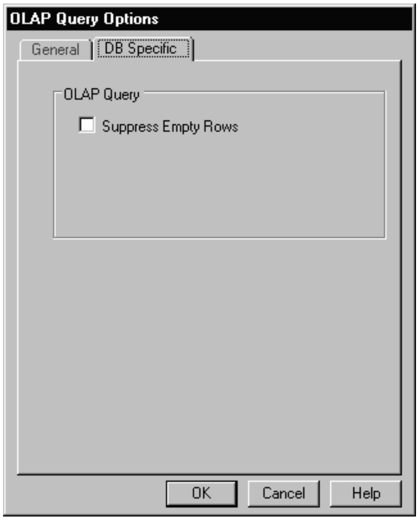


Table 6-3 Display Options for Microsoft/OLE DB for OLAP Database

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

Building OLAP Queries

Brio Intelligence uses the Outliner to manage multidimensional information. The Outliner enables you to plot, view, and manipulate dimensions. The three Outliner panes are:

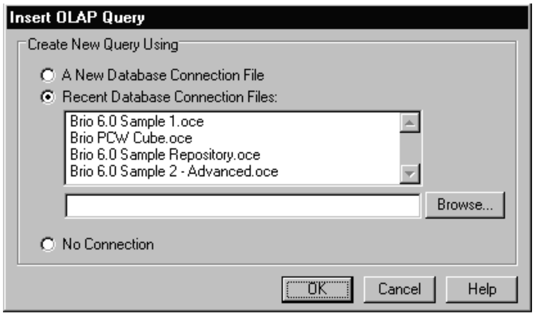
- Side Labels – Contains nonquantifiable dimensions or members.
- Top Labels – Contains nonquantifiable dimensions or members.
- Measures – Contains quantifiable dimensions or measures. The Measures pane in Outliner holds the cube's values.

Note Brio Intelligence treats time and date dimensions as nonquantifiable values.

To build an OLAP query:

1 Create an OLAP connection file, or if you already have an OLAP connection file, choose **Insert→New OLAPQuery**.

The Insert OLAP Query dialog box appears.



Insert OLAP Query

Create New Query Using

○ A New Database Connection File

● Recent Database Connection Files:

Brio 6.0 Sample 1.oce
Brio PCW Cube.oce
Brio 6.0 Sample Repository.oce
Brio 6.0 Sample 2 - Advanced.oce

○ No Connection

See Chapter 1, "Understanding Connectivity" in the *Brio Intelligence Administrator's Guide* for information on creating connection files.

2 Specify an OCE to use to connect to the multidimensional database; enter your user name and password if prompted.

The Brio Intelligence document creates and switches to an OLAPQuery section. The Catalog pane displays the hierarchy of your multidimensional database.

3. If the Outliner is not visible, click **Outliner** on the Section title bar.
4. In the Catalog pane, select one or more measures (such as units or amounts) and choose **OLAP**→**Add Fact/Measure** to add the item(s) to the Measures pane in Outliner.
5. In the Catalog pane, select one or more levels or members and choose **OLAP**→**Add Side Labels** to add the item to the Side Labels pane in Outliner.
6. In the Catalog pane, select one or more levels or members and choose **OLAP**→**Add Top Label** to add the item to the Top Labels pane in Outliner.

To refresh the dimension values in the Catalog pane:

➤ Choose **OLAP**→**Retrieve Dimensions**.

When you drag items from the Catalog pane to the Outliner, only the level names appear. For example, if you drag CA into the Outliner, the Outliner displays State. Level names appear with a ⊽ 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.

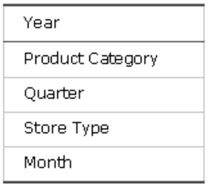
OLAPQuery Section Outliner Rules

The following rules apply to the OLAPQuery section Outliner:

- 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, Brio Intelligence automatically moves all levels of the same dimension.
- To pivot data in the OLAPQuery section, move the items in the Outliner panes. You cannot use the label handles to pivot data.

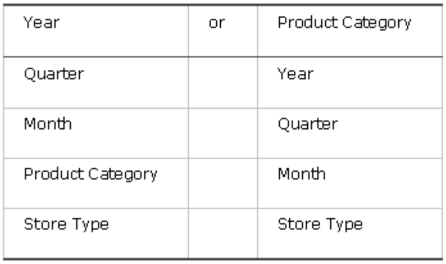
- Levels from the same dimension must be grouped together in both the Side Labels and Top Labels panes.

  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).



Year
Product Category
Quarter
Store Type
Month

  Instead, the Outliner requires that you use this order (*Year*, *Quarter*, and *Month* are all from they same dimension so they are grouped together).



Year	or	Product Category
Quarter		Year
Month		Quarter
Product Category		Month
Store Type		Store Type

- The hierarchy of a dimension cannot be broken. For example, *Year* must come before *Quarter*, which must come before *Month*.
- **OLE DB only –** If you retrieve dimensional level properties from the database, you can drag each property into the Outliner after you add its corresponding level.

  For example, if the Outliner contains Country, Year and you drag Manager, (a property of Country) into the Outliner, 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 Outliner, 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 OCE.

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 then select San Francisco (which is a city), your query retrieves all states and San Francisco.

Table 6-4 shows the results of different queries.

Table 6-4 Query Results

Outliner 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
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

Note If the same query is run against a MetaCube database, an intersection of the two data sets occurs, and not a union. For example, if you select the Regions (All) level and then select San Francisco (which is a city in California), you query retrieves the Western region and the city of San Francisco.

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 limits, 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

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 "Automatically Processing OLAP Queries" on page 6-19), 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 find a specific item that you want to learn more about, such as a product line, you can drill down into the item label. You can drill down on more than one item as well as drill down on all items at the same time.

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, you show only the data for that label value. When you select the dimension tab for a level, you show all the members of that dimension level.

Use one of the following 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→Drill Down.

☆ Tip You can specify what level of data is the next level displayed when you drill down in an OLAPQuery. See "Drill Options" in Table 6-1 on page 6-5 for information.

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 together. 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, then 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.
- Choose **Drill Up** on the shortcut menu.
- Choose OLAP→Drill Up.

Adding Computed Items

Computed items allow you to create a new column by building an expression to compute measures, or by applying functions to existing measures. Computed items are like normal data measures and can be included in reports or reused to compute other measures.

Computed items appear 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.

To compute or modify a measure:

1. Choose **OLAP→Add Computed Item**.

   The Modify Item dialog box appears.

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 keyboard 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 Function dialog box changes to accommodate the selected function. For more information about functions, see "Using OLAPQuery Functions" on page 6-16.

3 When the expression is complete, click **OK**.

The new measure name is added to the Outliner.

For more information on computed items, see Chapter 8, "Working with 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:

- Brio Intelligence 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. Brio Intelligence supports a number of MDX functions. (For more information on MDX functions, consult your MDX documentation.)

Table 6-5 lists the functions available in the OLAPQuery section.

Table 6-5 OLAPQuery Functions

Function	Type of Function	Description
% of Column	Brio Intelligence	Calculates the value of the specified measure as a percentage of the total for the column.
% of Row	Brio Intelligence	Calculates the value of the specified measure as a percentage of the total for the row.
% of Total	Brio Intelligence	Calculates the value of the specified measure as a percentage of the total for all rows and columns.
% Change	Brio Intelligence	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.
Absolute Change	Brio Intelligence	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	Calculates the linear regression of a dataset and returns the value of “b” in the regression line y = ax + b.
Linregpoint	MDX	Calculates the linear regression of a dataset and returns r2 (the coefficient of determination).
Linregr2	MDX	Calculates the linear regression of a dataset and returns the value of “a” in the regression line y = ax + b.
Linregslope	MDX	Calculates the linear regression of a dataset and returns the variance that fits the regression line y = ax + b.
Linregvariance	MDX	Returns the maximum value of the selected measure evaluated over the specified dataset.

Table 6-5 OLAPQuery Functions *(Continued)*

Function	Type of Function	Description
Max	MDX	Calculates the median value of the selected measure evaluated over the specified dataset.
Median	MDX	Returns the minimum value of the selected measure evaluated over the specified dataset.
Min	MDX	Calculates the standard deviation of the selected measure evaluated over the specified dataset.
Stdev	MDX	Calculates the sum of the selected measure evaluated over the specified dataset.
Sum	MDX	Calculates the variance of the selected measure evaluated over the specified dataset.
Variance	MDX	Calculates the average of the selected measure evaluated over the specified dataset.

To apply a data function:

1 Select the Brio Intelligence or MDX **function** that you want use from the Functions list.

A description of the selected function appears below the Functions list and explains the type of calculation the function performs.

2 Select the **measure** to which you want to apply the function.

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.

3 Define the **dataset** by which to evaluate the function and click **OK**.

To add a member to the dataset from the Cube Hierarchy list, select a member or level and click **Add**. The member or list is added to the Evaluated By box.

To remove a selected member or level from the Evaluated By box, select a member or level and click **Remove**.

Processing OLAP Queries

After you build your OLAP query and apply limits, 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:

➤ Choose **Tools**→**Process Query**→***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.

Brio Intelligence sends the query to the database and retrieves 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.

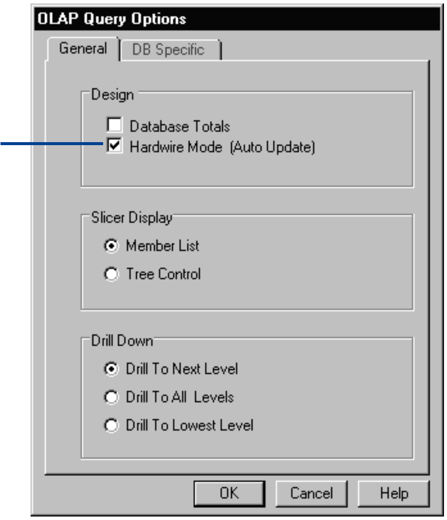
Automatically Processing OLAP Queries

If you select to run in Hardwire mode, Brio Intelligence automatically requeries the database everytime you add an item to or remove an item from the Outliner 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:

➤ Choose **OLAP**→**OLAP Query Options** and select **Hardwire Mode** in the Design section of the General tab.



OLAP Query Options

General | DB Specific

Design

☐ Database Totals
☑ Hardwire Mode (Auto Update)

Slicer Display

● Member List
○ Tree Control

Drill Down

● Drill To Next Level
○ Drill To All Levels
○ Drill To Lowest Level

Select Hardwire Mode to automatically process your OLAPQuery everytime you add or remove an item.

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 Brio Intelligence application’s 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:

➤ Choose **OLAP**→**Download To Results**.

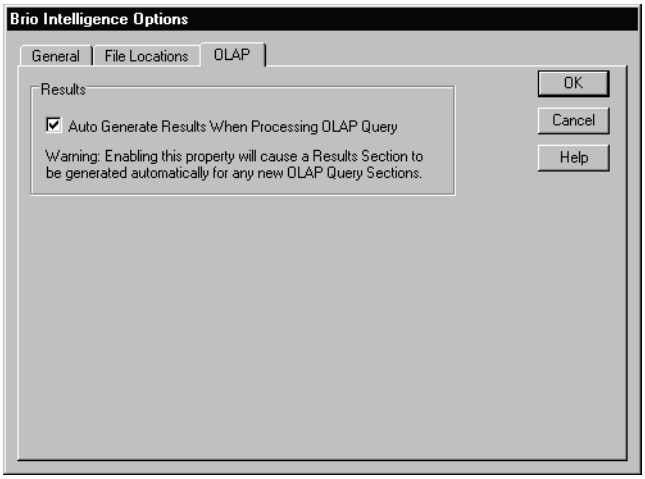
An OLAPResults section is created for the query. You can use the OLAPResults data set to insert a new chart, pivot, or other report.

Automatically Creating a Results Section

You can have Brio Intelligence automatically create a Results section when you click Process to process your OLAPQuery. This eliminates the need to select OLAP→Download to Results (see “Working with an OLAPQuery Offline”).

To automatically create a Results section when you click Process:

1 Choose **Tools**→**Options**→**Program Options** and click the **OLAP** tab.



Brio Intelligence Options

Results

☑ Auto Generate Results When Processing OLAP Query

Warning: Enabling this property will cause a Results Section to be generated automatically for any new OLAP Query Sections.

2 Select **Auto-Generate Results Section When Processing an OLAP Query**.

When you select this option, Brio Intelligence automatically creates a Results section for any OLAPQuery section you create in this session when that OLAPQuery section is first processed.

Applying Filters and Limits

Filters enable you to define and apply limits 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.

Applying a MetaCube Filter

A MetaCube filter is a customized limit defined on the database. For example, a filter can be created called *West*, which is defined as the values of *Oregon*, *California*, and *Washington* for the state level. If you place the *West* filter on a query, data for only those three states is retrieved. In this example, the filter behaves exactly as if you had manually selected the three states as limiting criteria on the state column.

The two types of MetaCube filters are:

- **Parameterized filter –** Prompts you for a specific value when the query is run. For example, if the *West* filter is a parameterized filter, Brio Intelligence prompts you to select the states to be included when the query is processed.
- **Nonparameterized filter –** Contains hard-coded values inside of the filter's definition. When a filter is set, data is returned from the database only if it meets the conditions of the filter.

Note You can have multiple filters in a query, but each dimension can have only one filter.

To apply a MetaCube filter:

1 In the Query Parameters dialog box, select a level where a parameter is required to satisfy the filter syntax.

The Message box at the top of the dialog box shows you exactly how many parameters are required to satisfy all filters included in the query.

2 Enter a value or operand for the filter in the Value field and click **OK**.

If you do not know the value to enter, click Show Values to display members for the selected level from the database. Select the member value from the Choice list and use the right arrow to move it to the Values field.

To remove a value, select the value and click the left arrow.

Applying Member Selection Limits

Use a member selection limit to filter data retrieved from the server cube. A member selection limit is similar to a slicer, except that the member selection limit introduces the member value in your report, and multiple members may be selected from a single hierarchy.

To apply a member selection limit:

➤ Drag individual member values from the Catalog pane to the Outliner.

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 limits.

To apply a member selection limit using an expression:

1 Drag a level into the Outliner and double-click the level name.

The Filter dialog box appears.

2 Specify the information requested in the fields in Table 6-6, and then click **OK**.

Table 6-6 Options in the Filter Dialog Box

Operator Type	Specifies the selection method for retrieving items from the database. The selections shown on this list depend on the database to which you are connected. See “Operator Types and Data Operators” on page 6-29.
Members	Displays the members that can be used as limits.
Set As Variable	Allows you to specify a variable limit that can be defined at runtime after you click Process.

Table 6-6 Options in the Filter Dialog Box *(Continued)*

Applied Filters	Shows the members to apply as limits. Multiple limits can be set at the same time on a level. For example, you can specify the top 10 states and specifically CA. You can also specify the top 10 and bottom 10 states and so on.
Drilldown Values	Displays members that have been drilled into using the Drill Down feature. To use these members as a constraint, use the Drill Up feature to return to the original level before applying Member Selection limits.
Show Values	Displays the members that belong to the level from the multidimensional database. You can also use this option to read limit values for a Metacube OLAPQuery from an external file. To do this: ■ Select the operator type **Load Values From External File** from the Operator Type field. ■ Click **Show Values**. A browse/select dialog box appears. ■ Select the appropriate tab-delimited file from which to read limit values. After you select the file, Brio Intelligence populates the Members pane with the values from the external field. ■ Select the values to be used as limit criteria and move them to the Applied Filters pane.

Additional parameters for selected multidimensional databases include:

Table 6-7 Additional Parameters for Selected MDD Databases

Search String (Essbase Only, Match Member)	Specify the search string where the limited item equals the specified value(s).
User Defined Attributes (Essbase Only)	When selected, move the attribute to include from the list of user defined attributes for the dimension to the Applied Filters box. The attribute is preceded with the tag UDA (for example, UDA: Summer).

Applying Measure Limits (Essbase and MetaCube Only)

Data retrieved from the server cube can be limited using a measure limit, which is similar to a member selection limit. A measure limit 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 limit:

1 Double-click a **measure** in Outliner.

The Filter dialog box appears.

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 6-29.

3 Specify any database specific **parameter** requests, such as a column index (that is, the column on which to apply the measure limit) or value.

4 Move the member(s) to the **Applied Filters** list and click **OK**.

Applying Variable Limits

A variable limit is a limit you specify when you process a query. You can use variable limits for standardized documents that you distribute to many users, or to automatically reset limits when you need new conditions every time you run a particular query. A (v) next to an item indicates it has a variable limit.

You place a variable limit on an item using the Filter dialog box, which is accessed in either the Slicer pane or the Outliner, depending on your database. Table 6-8 specifies how various OLAP databases access the Filter dialog box.

Table 6-8 Filter Dialog Box Access Points for Variable Limits

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

Member Variable Limits

To set a *member* variable limit:

1 In the Outliner, double-click the item you want to define as a member limit.

The Filter dialog box appears.

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 6-29.

3 Specify any database-specific **parameter** requests.

4 Click **Show Values** to display values in the database.

5 In the Members field, select the **member(s)** that you want to define as a **limit**.

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 limit** in the Members field.

10 Move the member(s) as a variable limit to the **Applied Filters** list and click **OK**.

Measure Variable Limits

To set a *measure* variable limit:

1. Double-click the item to define as a **measure limit** in the Outliner.

   The Filter dialog box appears.

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 6-29.

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. Choose a measure variable limit 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 limit to the **Applied Filters** list and click **OK**.

Slicer Variable Limits

To set a *slicer* variable limit:

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 that you want to use to filter data.

   The Slicer dialog box appears.

4. Select the member for which you want to filter data.

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 limit** and click **OK**.

Essbase Substitution Variables

In Essbase, a substitution variable acts as a dynamic limit. Substitution variables are defined on the server using Essbase's 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 Outliner.

The Filter dialog box appears.

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) that you want to use as a **variable limit**.

6 Move the member(s) you want to use as a variable limit 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 6-9)
- MetaCube Operator Types and Data Operators (see Table 6-10 and Table 6-11)
- Essbase Operator Types and Data Operators (see Table 6-12)

Note Not all providers support all operators.

Table 6-9 OLE DB Operator Types and Data Operators

Operator Type	Description
Select Members from DB	Creates a filter based on a member(s) retrieved from the database.
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 limited item equals the specified value(s).
< > Not Equal	Retrieves only records where the limited item does not equal the specified value(s).

Table 6-9 OLE DB Operator Types and Data Operators *(Continued)*

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

Table 6-10 MetaCube Operator Types and Data Operators

Operator Type	Description
Select Members	Creates a filter based on a member(s) retrieved that you specify.
Select Members from DB	Creates a filter based on a member(s) retrieved from the database.
Load Values From External Files	Allows you to read limit values from a tab delimited external file. To use this operator type, select the Load Values from External Files operator type and click Show Values. A standard windows browse/select dialog displays so that you can select the appropriate tab delimited external file. After you select the file, the Members pane is populated with the values from the external file. You can then select the values to be used as limit criteria in the query and move them to the "Applied Filters" pane.
Data Operator	**Description**
= Equal	Retrieves only records where the limited item equals the specified value(s).
Not Equal	Retrieves only records where the limited item does not equal the specified value(s).

Table 6-11 MetaCube Data Operators for Measures

Data Operator	Description
= Equal	Retrieves only records where the limited item equals the specified value(s).
< > Not Equal	Retrieves only records where the limited item does not equal the specified value(s).
< Less than	Retrieves only records where the limited item is less than the specified value(s).
< = Less than or Equal to	Retrieves only records where the limited item is less than the specified value(s).
> Greater than	Retrieves only records where the limited item is greater than the specified value(s).
> = Greater than or Equal to	Retrieves only records where the limited item equals, or is greater than the specified value(s).
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.
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 (the smallest number possible) values such that their sum is at least the specified Index value.

Table 6-12 Essbase Operator Types

Operator Type / Data Operator	Definition
Equal	Retrieves only records where the limited item equals the specified value(s).
Match Member	Retrieves only records where the limited item does not equal the specified value(s).
Data Operator	**Description**
= Equal	Retrieves only records where the limited item equals the specified value(s).

Table 6-12 Essbase Operator Types *(Continued)*

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

Changing Data Views

In the OLAPQuery section, there are numerious ways to change the way you view the data. Changing data view involves:

- Suppressing Rows
- Adding Totals
- Adding Data Functions
- Showing 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:

➤ Choose **OLAP**→**OLAP Query Options** and click on the **DB Specific** tab.

The options that appear on the DB Specific tab depend on the database to which you are connected. See “Database-specific OLAPQuery Options” on page 6-6 for more information.

Adding Totals

Brio Intelligence allows 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 on the Brio client.

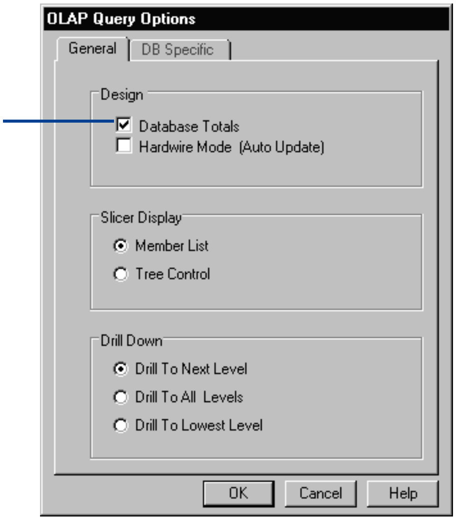
Adding Database Totals

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 Choose **OLAP→OLAP Query Options** and click **Database Totals** in the Design section of the General tab.



Design
☑ Database Totals
□ Hardwire Mode (Auto Update)

Slicer Display
● Member List
○ Tree Control

Drill Down
● Drill To Next Level
○ Drill To All Levels
○ Drill To Lowest Level

Click here to retrieve database totals when you process the query.

2 Click **Process** to add the totals to the data.

Brio Intelligence totals the data and displays the result as the *first* item at each level of the dimension.

Note If you enable database totals in the OLAPQuery section, Brio Intelligence copies them 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 checkmark 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**.

Brio Intelligence 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. **[Del]**

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 the summation icon on the Standard toolbar to calculate the total.
3. Select a column of a particular measure.
4. Choose **OLAP**→**Data Function** and select a function.

Table 6-13 lists the data functions available in the OLAPQuery section.

Table 6-13 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.
% 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 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:

➤ Choose **OLAP**→**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. For more information on formatting options, see “Working with Document Sections” on page 3-34.

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→California→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

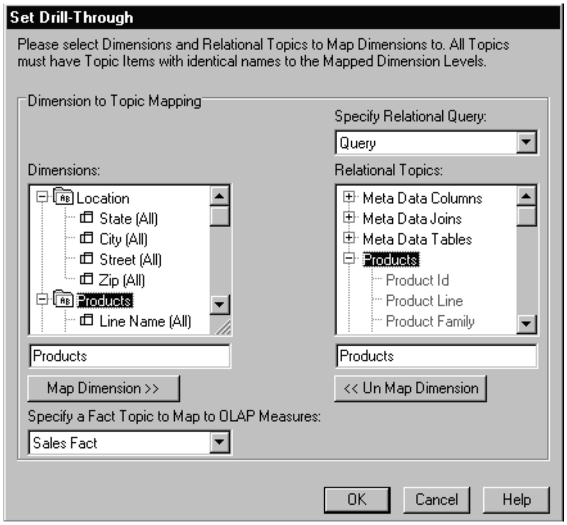
Setting Drill-through Options

Drill-through options define the mapping between a multi-dimensional database and a relational database.

To set drill-through options.

1 Choose **OLAP**→**Set Drill-Through**.

The Set Drill-Through dialog box appears.



Set Drill-Through

Please select Dimensions and Relational Topics to Map Dimensions to. All Topics must have Topic Items with identical names to the Mapped Dimension Levels.

Specify Relational Query: Query

Dimensions: Products

Relational Topics: Products

Specify a Fact Topic to Map to OLAP Measures: Sales Fact

2 Fill in the options in the Set Drill-Through dialog box and click **OK**.

Table 6-14 Options in the Set Drill-Through Dialog Box

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 be used 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 appears.
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 be used 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, Brio Intelligence stores an internal link between the dimension and the selected topic.

Table 6-14 Options in the Set Drill-Through Dialog Box *(Continued)*

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 appears. The topics that appear are the topics available in the selected query.

☆ **Tip** In order 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

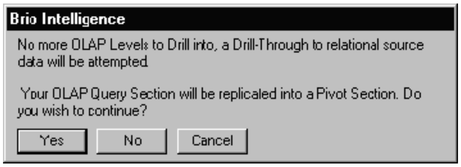
To drill through from a mulit-dimensional database to a relational database:

1 Select a dimension and drill-down to the lowest level.

2 Do one of the following:

- Double-click the dimension.
- Select OLAP→Drill Through.

The Brio Intelligence Drill-Through dialog box appears letting you know that there are no additional OLAP levels to drill into.



Brio Intelligence

No more OLAP Levels to Drill into, a Drill-Through to relational source data will be attempted

Your OLAP Query Section will be replicated into a Pivot Section. Do you wish to continue?

3 Click **Yes** to drill through to the relational data source.

Brio Intelligence creates a new Pivot Section (along with associated Query and Results sections) for the relational data.

Note Brio Intelligence ignores slicers when drilling through to a relational database. If a slicer is present, a message appears 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 an existing pivot section with the new dimension data.

If you update an existing 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 existing Pivot.

OLAP Menu Command Reference

Table 6-15 provides a quick reference to the commands available on the OLAP menu and lists any related shortcuts.

Table 6-15 OLAP 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. (The availability of this option depends on the database to which you are connected.)		✔
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.		

Table 6-15 OLAP Menu Commands *(Continued)*

Command	Description	Keyboard Shortcut	Shortcut Menu
Remove Total	Removes local totals from selected dimensions.	[Del]	✔
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.		

What Next

For more information about features used in the OLAPQuery section, read Chapter 7, “Using Limits,” Chapter 8, “Working with Computed Items,” and Chapter 9, “Applying Sorts.” When you are finished formatting and fine-tuning your results, see *Data Analysis and Reporting with Brio Intelligence 6.6* for assistance with using Brio Intelligence’s powerful reporting and data analysis features.

7 Using Limits

Sometimes you do not need to work with all of the data in a particular category. This chapter explains how to use limits in the Query and Results sections to refine your data for more intelligent analysis and filter away the data you do not need. It contains:

- About Limits
- Setting Simple Limits
- Setting Compound Limits
- Setting Variable Limits
- Customizing Limit Options

About Limits

When building a query, you usually do not want to see information associated with 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 limit 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 your most important customers who spend more than $400,000 per year, or who buy gardening products in the Midwest. By applying a limit 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 limit 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 limit expression.

Similarly, the limit "> 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 Limit Processing

You apply limits in the Query or Results sections in basically the same way. The difference is in whether the limit is applied at the server or on your desktop.

Server Limits in the Query Section

A limit 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 limits, it is best to apply limits 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 limits in the Query section is that you can apply a limit to any topic item in the data model, even if the item is not on the Request line. You can also set limits on computed items.

Local Limits in the Results Section

You can apply limits to columns to locally filter the data set in the Results section. Since the other reporting sections reference the results set, these local limits are disseminated to these sections.

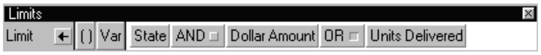
Local limits are useful for managing your data set. If you decide you do not need all the information retrieved by your query, you can use a local limit to exclude data from view. Local limits are a good way to filter the data set for temporary or hypothetical situations. You can always suspend or delete the limit to view the data and make it available for reporting.

Note You can only apply local limits to items on the Request line.

Limit Line

Setting limits involves dragging an item from the Content pane to the Limit line and then setting the desired limits in the Limit dialog box.

The Limit line is a drag and drop command line similar to the Request line and Sort lines. Limited items are displayed on the Limit line. The Limit line can be moved, sized, docked, and hidden.



Limit indicators appear on the Limit line and next to the item in the topic. The Limit line expressions available in the Query section Limit line are:

- () – Encloses suboperations.
- Var – Indicates a variable.
- AND – Retrieves data that meets both condition.
- OR – Retrieves data that satisfies either of two conditions.

Limit Line Functionality

In the Query section, the Limit line includes special functionality:

- The Limit line is an interface for building compound limits. Compound limits are multiple limits linked together to form complex limit equations. (See “Setting Compound Limits” on page 7-9 for more information.)
- The Limit line enables you to convert an existing limit to a variable limit, which prompts the user of the document to select limits as the query is processed. (See “Setting Variable Limits” on page 7-10 for more information.)
- When placing a limit on a request item computed with a data function, a divider appears on the Limit line, and the Limit icon is placed to the right of the divider. The divider indicates that the limit is applied in the SQL Having clause.
- Subqueries and correlated subqueries use the result from an inner query as the value of a limit in an outer query.

Limit Line Syntax

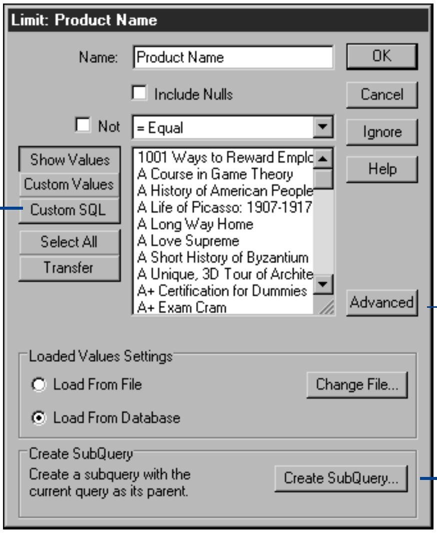
The following rules of syntax apply to all Limit line expressions.

- 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 limit 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 limit 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.

Limit Dialog Box

All limits are applied using the Limit dialog box. When applying a limit, you supply (or select from a list) data values associated with a data item and use mathematical logic to apply the values as constraints.

Custom SQL appears only if you access the Limit dialog box from the Query section.



Limit: Product Name

Name: Product Name

□ Include Nulls

□ Not = Equal

1001 Ways to Reward Emplc
A Course in Game Theory
A History of American People
A Life of Picasso: 1907-1917
A Long Way Home
A Love Supreme
A Short History of Byzantium
A Unique, 3D Tour of Archite
A+ Certification for Dummies
A+ Exam Cram

Loaded Values Settings

○ Load From File

● Load From Database

Create SubQuery

Create a subquery with the current query as its parent.

Displays as *Advanced* in the Query section and *Options* in all other sections.

Create Subquery appears only if you access the Limit dialog box from the Query section.

Table 7-1 Options in the Limit Dialog Box

Name	Descriptive name for the limit.
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 limit expression. Values that pass the comparison test will be included. For detailed information about the operators that are available when defining limits, see the online help for the Limit dialog box.
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 limit definition.Click the "x" to erase the contents of the Edit field. **Note:** The Edit field appears *only* if you are entering a custom value.

Table 7-1 Options in the Limit Dialog Box *(Continued)*

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 limits on computed, aggregate, or computed metatopic items. For detailed information about showing values, see the online help for the Limit dialog box.
Custom Values	Lists potential values saved with the limit 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 limit. For detailed information about defining custom values, see the online help for the Limit dialog box.
Custom SQL	Displays the custom SQL dialog box for coding Limits directly in SQL. **Note:** The Custom SQL button appears *only* if you access the Limit dialog box from 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 limit without deleting it.
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 allow you to specify the file name.If you read values from a text file, vertical tabs or paragraph markers must delimit each value. Use Show Values to display the file contents. ■ **Create Subquery** – Creates a subquery where the current query is the parent. (For information about subqueries, see “Building Subqueries” on page 4-8.) **Note:** If you access the Limit dialog box from the Results section, the *Advanced* button appears as *Options* and the Create Subquery option does *not* appear.

Note Limits apply only to the chosen data item. You can combine limits into more complex Compound Limits using the Limit line.

Setting Simple Limits

Simple limits are applied in basically the same manner in the Query and Results sections. If you are working with a distributed document, some of the buttons in the Limit dialog box may not be available.

Limiting Queries

The simplest way to limit a query is to apply limits individually to topic items.

To limit a query:

1 Select an item in the data model and choose **Query→Add Limit(s)**. **[Ctrl+L]**

The Limit dialog box appears.

2 Select an arithmetic or logical operator from the drop-down list.

3 Define the potential limit 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 previously saved custom list or values loaded from a file.
- **Custom SQL** – Supplies an empty text box for typing an SQL clause to be included in the query statement. Choose View→Custom SQL to check the complete statement logic.

4 Select the values to include in the limit 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 limit 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 limit is applied to the topic item and an icon is added to the Limit line.

In the Query section, the limit is applied when you process the Query.

Note When placing a limit on an aggregate item (computed with a data function) in the Query section, a limit divider appears and the limit icon is placed to the right. The divider indicates the limit is located in the SQL Having clause. You need to drag the aggregate item from the Request line to the Limit line to create a limit on an aggregate item.

To remove a limit on a query:

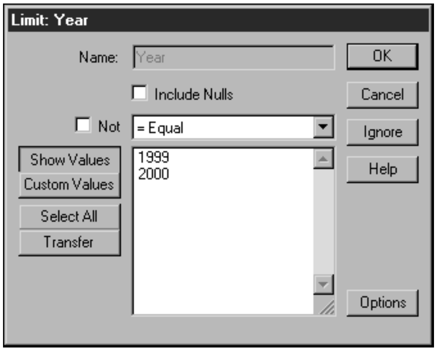
➤ Select an item on the Limit line and choose **Remove** on the shortcut menu. **[Del]**

Limiting Results

To limit the display of data in the Results section:

1 Select a column (click the column heading) and choose **Results→Limit**. **[Ctrl+L]**

The Limit dialog box appears.



Limit: Year

Name: Year

□ Include Nulls

□ Not = Equal

1999
2000

2 Select an arithmetic or logical operator from the drop-down list.

3 Define the potential limit 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 limit 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 limit definition.

5 When the correct values are highlighted in the Values list, click **OK**.

An indicator is added to the Limit line and the limit is immediately applied to the data set.

To remove a local limit:

➤ Select an item on the Limit line and choose **Results**→**Remove**. **[Del]**

Setting Compound Limits

In some cases, you may want to set two distinct limits on the same Request item or create compound constraints using more than one item.

Use the Limit line to build compound limit expressions. The Limit line enables you to apply more than one limit to a single item, or create compound conditions dependent on more than one constraint.

Drag an item to the Limit 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 Limit line twice, and set two separate limits. An AND operator is placed between the icons on the Limit line and the data set is constrained by both conditions.

Similar logic can be applied using two different limits and substituting the OR operator. For example, to retrieve customers located in the Cleveland area and those who have purchased more than $100,000 worth of goods, you can place a limits 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 Limit line displays a “_2” next to the item name.

To create compound Limit line expressions:

1 Add two or more items to the Limit line and apply individual limits using the Limit dialog box .

An AND operator appears between each item on the Limit line.

2 In the Query section only, click the small arrow at the left edge of the Limit line.

The Limit line is adjusted to display the Limit line control buttons.

3 On the Limit line, select limit controls to complete the equation.

- To switch Boolean operators AND and OR, double-click the operator you want 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 Limit line, click **Var** to make the limit variable.

When using compound limits, verify that the expression delivers the correct results.

Setting Variable Limits

A variable limit is a limit in the Query section that is resolved only when a query is processed. At that time the user is prompted to select or enter limit values and complete the constraint. You can use variable limits in standardized documents and distribute them to users to supply different limit values for each process.

Variable limits work particularly well with custom lists. If a custom list has been saved with a variable limit, the user can respond to the prompt by simply selecting a value from the custom list.

For example, you may have a document you use monthly to monitor inventory levels. Each time you use the document, you run it separately for each product line you carry. You can accelerate the process by making the limit variable on the product line item, and create a custom values list. Each time you process the document, you can select a new product line without redefining limits.

Note Variable limit functionality is only available if you are connected to OnDemand Server, working with a registered OnDemand Server document, and the document has query and analyze privileges.

To set a variable limit:

1 If necessary, add a topic item to the Limit line and define a limit.

If you are providing a custom list, add all the values you want to make available to the list of values.

Note You need to select at least one value in the list to save the limit, 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 limit, which offers all values in the database or in the custom list.

2 Select a topic item on the Limit line.

3 Choose **Query**→**Variable Limit**.

The Limit item is displayed with a V(1) beside the item name to indicate it is a variable limit.

If other items are set to variable limits, they are displayed with V(2), V(3), and so on, to indicate the order in which the user is prompted to respond to the Variable Limit dialog box when the query is processed.

Customizing Limit Options

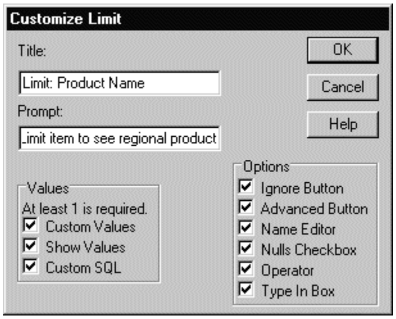
You can control access to the features available in the Limit dialog box, which can be especially useful when you distribute documents to end users. For instance, if you plan to distribute a document to novice users who need to set a simple variable limit , it may be preferable to disable or even remove such features as Include Null or the Custom Values buttons.

⟹ **Note** Customizing affects a single limit. Setting global limit preferences that restrict the options available throughout a distributed data model is an advanced feature not covered in this book.

To customize limit options:

1 Select an item on the Limit line and choose **Query**→**Customize Limit**.

The Customize Limit dialog box appears.



Title: Limit: Product Name

Prompt: Limit item to see regional product

Values
At least 1 is required.
☑ Custom Values
☑ Show Values
☑ Custom SQL

Options
☑ Ignore Button
☑ Advanced Button
☑ Name Editor
☑ Nulls Checkbox
☑ Operator
☑ Type In Box

2 Customize the limit as follows:

- Title – Add a title or text to the limit dialog to instruct users on how to set the limit 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 option.
- Options – Disable elements, lock in operators , or disable inputting of custom values .

Click **OK**.

8 Working with Computed Items

This chapter describes how to use calculations to compute new data items in the Query, Results, and reporting sections. Such calculations are important for supplementing the information already stored in the database. It contains :

- About Computed Items
- Adding Computed Items
- Using Functions in Brio Intelligence
- Common Computed Item Examples
- Scalar Function Examples

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 Brio Intelligence to compute it for yourself.

Computing New Data

Computations can be performed by your database server or by Brio Intelligence 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.

Brio Intelligence 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.

Computed Items in Brio Intelligence 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 Brio Intelligence 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. Brio Intelligence 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.

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.

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 Brio-supported data types, see Table 8-1 .

Table 8-1 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 Brio Intelligence 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, Brio Intelligence 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 Brio Intelligence, 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 limited to the items that appear 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 Outliner. 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.

Using Surface Values

Brio Intelligence allows 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 appear 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 By default, the Surface Values feature is deactivated.

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.

Adding Computed Items

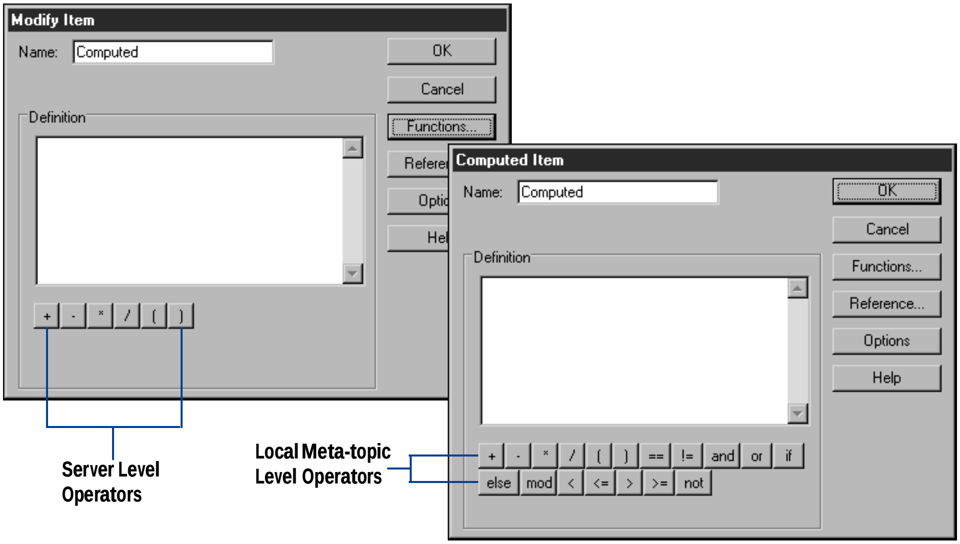
In the Query section, a computed item is a new data item derived from calculations. In the Results, OLAPQuery, Pivot, Chart and Report Designer sections, you add computed items by building equations to compute data items, or by applying functions to existing data items.

Computed items are like normal data items, and can be included in reports or reused to compute other data.

To create a computed item:

1 Choose **Add Computed Item** from a Section menu (for example, Query, Results, and so on).

The Modify Item or Computed Item dialog box appears.



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, Brio Intelligence 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 Outliner and appears as a column in the results set.

In the Query section, the computed item appears on the Request line with its new name. In the Results section, compute items appear in the Outliner in blue type.

Operators

Click the Operator buttons 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.
- All text string constant values entered in expressions must be enclosed in single quotes. All date constant values entered in expressions must be enclosed in quotes. Numbers can be entered without quotes.

- To join items with a space or other character, simply 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 8-2 Arithmetic Operators

+	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.

Table 8-2 Arithmetic Operators *(Continued)*

++	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.

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.

Note The comparison operators in Table 8-3 are only available at the local metatopic level.

Table 8-3 Comparison Operators

==	Returns true if the operands are equal.
!=	Returns true if the operands are not equal.
<	Returns true if the left operand is less than the right operand.
<=	Returns true if the left operand is less than or equal to the right operand.
>	Returns true if the left operand is greater than the right operand.
>=	Returns true if the left operand is greater than or equal to the right operand.

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 8-4 Statements

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.

Note The logical operators in Table 8-5 are only available at the local metatopic level.

Table 8-5 Logical Operators

AND (&&)	Connects two conditional expressions and retrieves records only if each expression is true.
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.
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.

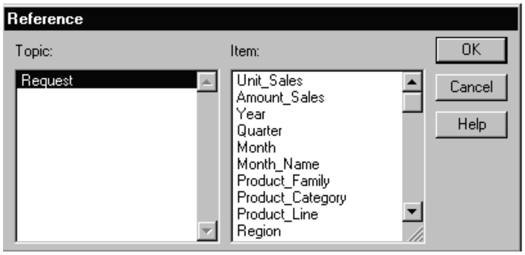
Reference Items

Use the Reference dialog box to insert Reference items into equations in the Computed Item (or Modify Item) dialog box. You can insert Reference items directly into an equation, or add them as arguments to scalar functions.

To place a Reference item in an equation:

1 In the Computed Items dialog box, click **Reference**.

The Reference dialog box appears.



2 Select a topic and an item, and then click **OK**.

The Reference item is inserted in the Definition text box.

Note The Reference dialog box differs between the Query, Results, and Pivot and Chart sections. In the Query section, any topic item in the data model is a possible reference item, including Request items. In the Results section, only columns or items in the Outliner are Request items. In the Pivot and Chart sections, only items in the Outliner are Request items.

Inserting Scalar Functions

Use the Functions dialog box to insert standard mathematical, logical, and text-based scalar functions in computed item expressions.

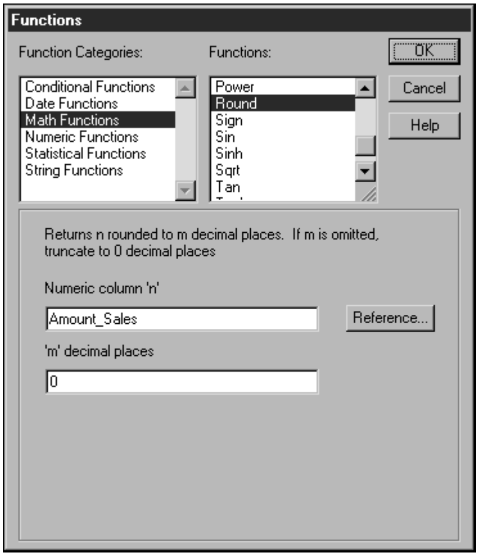
Scalar functions (in contrast to data functions) do not aggregate data or compute aggregate statistics. Instead, scalar functions compute and substitute a new data value for each value associated with a data item.

Note The Functions dialog box differs between the Query section and the Results, Table, Chart, and Pivot sections. In the Query section, your RDBMS software provides the operators and functions. In the Results, Table, Chart and Pivot sections, scalar functions and additional operators are provided by Brio Intelligence.

To apply a scalar function:

1 In the Computed Item dialog box, click **Functions**.

The Functions dialog box appears.



Returns n rounded to m decimal places. If m is omitted, truncate to 0 decimal places

Numeric column 'n'

Amount_Sales

'm' decimal places

0

2 Select a category from the Function Categories list.

Function categories include Conditional, Date, Math, Numeric, Statistical, and String.

3 Select a function from the Functions list.

The dialog box expands to display the arguments specific to the function, which may include Reference items and possibly numeric or text values. Most arguments are required, but a few are optional.

4 Enter the requested information in the fields provided and click **OK**.

Using Functions in Brio Intelligence

This section lists the functions available in Brio Intelligence, including:

- Scalar Functions
- Teradata Version 3 OLAP Functions
- Methods for Returning the Day of the Week

Scalar Functions

Tables 8-6 through 8-11 list the scalar functions available in Brio Intelligence.

Note In the tables for Conditional, Date, Math, and String 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.

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.

Table 8-6 Conditional Scalar Functions

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*.
Nvl (c,exp1,exp2)	Returns *exp2* if null, and *exp1* otherwise.

Table 8-7 Date Scalar Functions

AddMonths (d,n)	Adds *n* months to date *d*.
DayOfMonth (d)	Returns the day of month for date *d*.
LastDay (d)	Returns date of the last day of the month containing date *d*.
MonthsBetween (d1,d2)	Returns the number of months between dates *d1* and *d2* as a real number (fractional value).

Table 8-7 Date Scalar Functions *(Continued)*

NextDay (d,s)	Returns the date of the first weekday *s* after date *d*. If s is omitted, add one day to *d*.
Sysdate (c)	Returns the current system date and time for each record in item *c*.
ToCha (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. 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.
ToMonth (d)	Returns a numeric month value for each value of *d*. You can change the value to display as a month string by adding and applying a *mmm* date format.
ToQtr (d)	Returns a string quarter value for each value of *d*.
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.

Table 8-8 Math Scalar Functions

Abs (n)	Returns the absolute value of number *n*.
Atn (n)	Returns arc tangent of number *n* radians.
Ceil (n)	Returns the smallest integer value greater than or equal to number *n*.
Cos (n)	Returns cosine of number *n* radians.
Cosh (n)	Returns hyperbolic cosine of number *n* radians.
Count (c)	Returns the number of row values in *c* (including nulls).
Exp (n)	Returns *e* (2.718) raised to exponential power *n*.
Max (a,b)	Returns the larger of items *a* and *b* for each new value.

Table 8-8 Math Scalar Functions *(Continued)*

Min (a,b)	Returns the smaller of items *a* and *b* for each new value.
Mod (n,m)	Returns the integer remainder of number *n* divided by number *m*. If *m* is larger, the default value is *n*.
Power (n,m)	Returns number n raised to exponential power *m*.
Round (n,m)	Returns number *n* rounded to m decimal places. The default value for *m* is 0.
Sign (n)	Returns indicator of -1, 0, or 1 if number *n* is variously negative, 0, or positive.
Sin (n)	Returns sine of number *n* radians.
Sinh (n)	Returns hyperbolic sine of number *n* radians.
Sqrt (n)	Returns square root of number *n*.
Tan (n)	Returns tangent of number *n* radians.
Tanh (n)	Returns hyperbolic tangent of number *n* radians.
Trunc (n,m)	Returns number *n* truncated to number m decimal places. The default value for *m* is 0.

Table 8-9 Numeric Scalar Functions

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.
AvgNonNull (numbers, break_col, break_value)	Returns the average (arithmetic mean) of values in a number column, excluding null values.
chr (n)	Returns string converted from ASCII numeric code *n*.
ColMax (numbers, break_col, break_value)	Returns the largest value in a column of numbers.
ColMin (numbers, break_col, break_value)	Returns the smallest value in a column of number.
Count (numbers, break_col, break_value)	Counts and returns the number of rows in a column.
CountDistinct (numbers, break_col, break_value)	Counts and returns the number of values in a column.

Table 8-9 Numeric Scalar Functions *(Continued)*

CountNonNull (numbers, break_col, break_value)	Counts the number of rows in a column.
CountNull (numbers, break_col, break_value)	Counts the number of rows in a column that contains null values.
Cume (numbers, break_col)	Returns a cumulative running total for each value in a column of numbers.
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.

Table 8-10 Statistical Scalar Functions

Median (numbers, break)	Returns the median of a column of numbers.
Mode (numbers, break_col)	Returns the most frequently occurring value in a column of numbers.
Percentile (numbers, n, break_col)	Returns the *n*th percentile of values in a column of numbers in ascending order.
Rank (numbers, break_col)	Returns the rank of a number in a column of numbers.
RankAsc (numbers, break_col)	Returns the rank of a number in a column of numbers in ascending order.
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.

Table 8-10 Statistical Scalar Functions *(Continued)*

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.
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.
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.

Table 8-11 String Scalar Functions

Ascii (s)	Returns an ASCII numeric representation of string *s*.
Concat (s1, s2)	Returns text strings *s1* and *s2* concatenated.
Initcap (s)	Returns string *s* with the first letter of each word capitalized, and remaining characters in lower case.
Instr (s1,s2,n,m)	Returns position of *m*th 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.
Length (s)	Returns character count of string *s*.
Lower (s)	Returns string *s* in lower case.
Ltrim (s1,s2)	Trims string *s1* from the left, up to the first character not included in string *s2*.
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*.'
Rtrim (s1,s2)	Trims column string *s1* from the right, up to the first character not included in string *s2*.

Table 8-11 String Scalar Functions *(Continued)*

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.
Translate (s1,s2,s3)	Returns string *s1*, with each character contained in string *s2* replaced by the corresponding characters in string *s3*.
Upper (s)	Returns string *s* in upper case.

Teradata Version 3 OLAP Functions

Brio Intelligence supports a number of Teradata version 3 OLAP and system functions, which dramatically reduce query time. See Tables 8-12 through 8-22 for complete descriptions of these functions.

Table 8-12 CSum (Cumulative Sum) Function

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)
CSum	Returns the cumulative (or running) sum of a column expression. The CSum function assumes the rows are sorted by 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).

Table 8-13 MAvg (Moving Average) Function

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).

Table 8-14 MDiff (Moving Difference) Function

Description	Returns the moving difference between the current row-column value and the preceding *n*th 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.
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).

Table 8-15 MSum (Moving Sum) Function

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).

Table 8-16 MLingreg (Multiple Linear Regression) Function

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.

Table 8-17 Quantile Function

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.

Table 8-18 Rank Function

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).

Table 8-19 Current_Timestamp Function

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

Table 8-20 Qualify Function

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

Table 8-21 Sample Function

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.

Table 8-21 Sample Function *(Continued)*

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.

Table 8-22 SampleID Function

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 appear either as part of a SELECT clause or in an ORDER BY clause.

Methods 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, or if a user with Brio Intelligence Navigator needs to use the day of the week to perform analysis on data which does not already include this information, 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→Number**.

The Number page of the Properties dialog box appears.

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 T*uesday* 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→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:

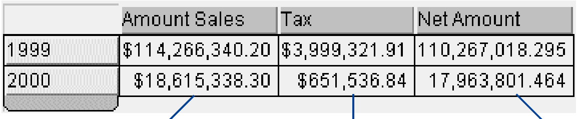
: decode((NEXT_DAY (<*MyDate*>, 'Sunday')) -<*MyDate*>, 7,'Sunday', 6,'Monday', 5,'Tuesday', 4,'Wednesday', 3,'Thursday', 2,'Friday', 1,'Saturday', 'Error!')

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

A mathematical equations 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

1. Create a query to retrieve Amount Sales for each year.

2. Create a computed item in the Pivot section:
`=Amount_Sales x 0.035`

3. Create a computed item in the Pivot section:
`=Amount_Sales – Tax`

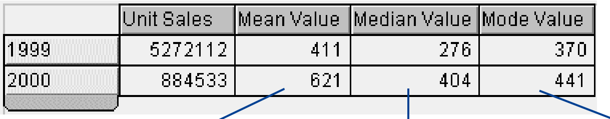
Measure of Central Tendency

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.

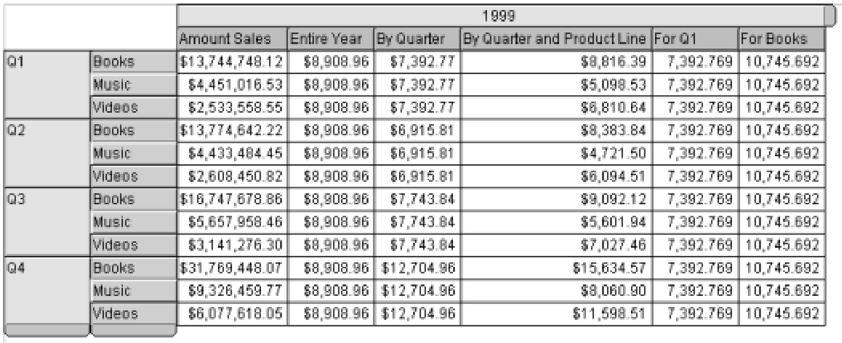


	Unit Sales	Mean Value	Median Value	Mode Value
1999	5272112	411	276	370
2000	884533	621	404	441

`= Avg (Unit_Sales, Year)` `= Median (Unit_Sales, Year)` `= Mode (Unit_Sales, Year)`

Average Numeric

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.



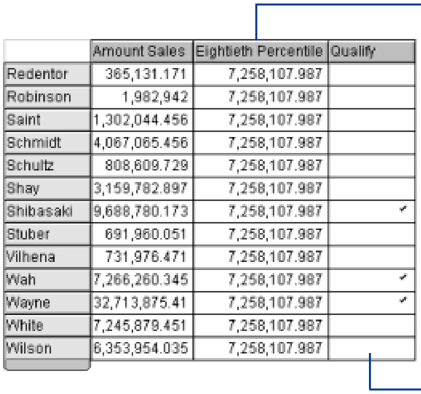
		1999					
		Amount Sales	Entire Year	By Quarter	By Quarter and Product Line	For Q1	For Books
Q1	Books	$13,744,748.12	$8,908.96	$7,392.77	$8,816.39	7,392.769	10,745.692
	Music	$4,451,016.53	$8,908.96	$7,392.77	$5,098.53	7,392.769	10,745.692
	Videos	$2,533,558.55	$8,908.96	$7,392.77	$6,810.64	7,392.769	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

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

The Percentile function returns the *n*th percentile of values in a distribution range that is equal to or below the percentile value. You might use the Percentile function to define a threshold of acceptance in a range of values.

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	

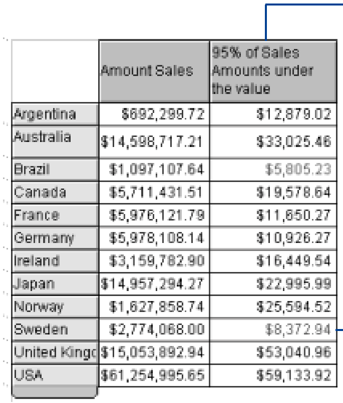
The Eightieth Percentile column calculates the 80th percentile value for all Sales managers:
`=Percentile (Amount Sales, .8)`

Important! Surface values must be used in this type of report.

The Qualify column determines whether a Sales Manager is within the 80th percentile:
`=if (Amount_Sales >= Eightieth_Percentile) {‘a’}`

The letter ‘a’ formats as a check mark using the Marlett font.

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 Kingc	$15,053,892.94	$53,040.96
USA	$61,254,995.65	$59,133.92

Calculated by finding the 95th percentile value within each country. Country is used as the break column.
`=Percentile (Amount Sales, .95, Country)`

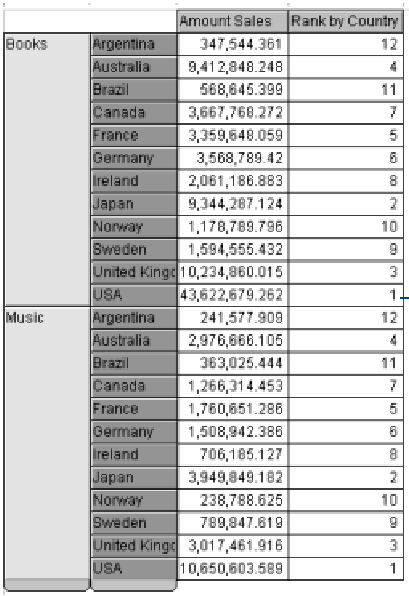
Important! Calculations must be done using the underlying values instead of the surface values.

Percentile values under $10,000.00 are spotlighted in red to identify the countries that have such sales transactions 95% of the time.

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.

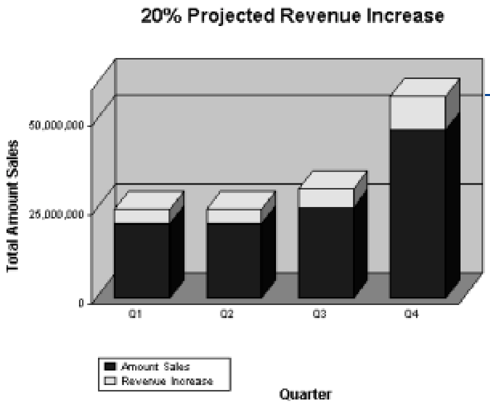


		Amount Sales	Rank by Country
Books	Argentina	347,544.361	12
	Australia	9,412,848.248	4
	Brazil	568,645.399	11
	Canada	3,667,768.272	7
	France	3,359,648.059	5
	Germany	3,568,789.42	6
	Ireland	2,061,186.883	8
	Japan	9,344,287.124	2
	Norway	1,178,789.796	10
	Sweden	1,594,555.432	9
	United Kingc	10,234,860.015	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 Kingc	3,017,461.916	3
	USA	10,650,603.589	1

Countries ranked by their Amount Sales.
Country is the break column.
`=Rank (Amount Sales, Country)`

Projected Sales

This example shows a calculation for a 20% increase in sales projections for each quarter, based on Amount Sales for 1999.



This bar represents the
Revenue Increase values.
`Amount_Sales * 2`

Scalar Function Examples

This section provides 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 Brio Intelligence scalar functions.

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.

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)
```

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.

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’)
```

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.

In this example, the ColMax function is used on the numeric column. The results are shown in the Computed column.

```
ColMax(Amount)
```

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

IIColMin

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.

IIn 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)
```

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	180
NY	Rochester	180	180

ICount

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.

In this example, the Count function is used on the numeric column. The results are shown in the Computed column.

`Count (Amount)`

State	City	Amount	Computed
AZ	Tucson	112	9
CA	Burbank	240	9
CA	Glendale	97	9
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

IICountDistinct

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 column named “OS Operating Systems” has one hundred rows and shows data by Windows, Macintosh, and UNIX systems, the `CountDistinct` function counts only the three operating systems and not the number of rows.

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)
```

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.

In this example, the `CountNull` function is used on the numeric column. The results are shown in the Computed column.

```
CountNull(Amount)
```

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.

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’)
```

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.

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)
```

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

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.

In this example, the Sum function is used on the numeric column. The results are shown in the Computed column.

```
Sum(Amount)
```

State	City	Amount	Computed
AZ	Tucson	112	1556
CA	Burbank	240	1556
CA	Glendale	NULL	1556
FL	Palmetto	70	1556
MD	Laurel	97	1556
MI	Detroit	208	1556
MN	Eagan	159	1556
NY	New York	490	1556
NY	Rochester	180	1556

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.

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.

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.

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.

In this example, the `Mode` function is used on the numeric column. The results are shown in the Computed column.

```
Mode(Amount)
```

State	Amount	Computed
AZ	10	10
CA	10	10
FL	0	10
MD	40	10
MI	50	10

Percentile

The `Percentile` function returns the *n*th 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 *n*th percentile to the following: 0.25 for first quartile, 0.5 for second quartile, 0.75 for third quartile.

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 *n*th 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 *n*th percentile of values (.25 in this case).

```
Percentile (Units, .25)
```

State	Units	Computed 1	Computed 2
WI	50	50	96
AZ	70	50	96
CA	96	50	96
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.

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)
```

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
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.

IIn 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)
```

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 your data represents the entire population, then compute the standard deviation using `StdDevp`.

The standard deviation is calculated using the *nonbiased* or *n-1* method.

If a result set contains one row of data or less, the `StdDev` function should return an error.

In this example, the StdDev function is used on the numeric column. The results are shown in the Computed column.

```
StdDev (Amount)
```

State	City	Amount	Computed
AZ	Tucson	112	128.11
CA	Burbank	240	128.11
CA	Glendale	97	128.11
FL	Palmetto	70	128.11
MD	Laurel	97	128.11
MI	Detroit	208	128.11
MN	Eagan	159	128.11
NY	New York	490	128.11
NY	Rochester	180	128.11

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`.

The standard deviation is calculated using the *biased* or *n* method.

In this example, the `StdDevp` function is used for the numeric column. The results are shown in the Computed column.

```
StdDevp (Amount)
```

State	City	Amount	Computed
AZ	Tucson	112	120.79
CA	Burbank	240	120.79
CA	Glendale	97	120.79
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

IVar

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`.

IIn 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)
```

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
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`.

In this example, the `Varp` function is used on the numeric column. The results are shown in the Computed column.

```
Varp (Amount)
```

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

9

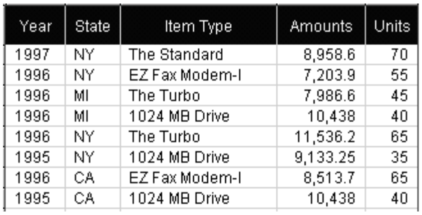
Applying Sorts

This chapter discusses features that enable you to sort data in various Brio Intelligence sections, including sort lines, single and nested sorts in the Query and Results section, and complex sorting by values and by labels in the Brio Intelligence report sections. It contains:

- Sorting Data
- Simple Sorts
- Sort Lines
- Complex Sorting

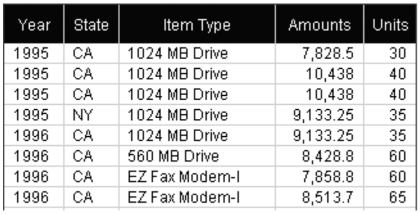
Sorting Data

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

Unsorted Results



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

Sorted Results

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

Brio Intelligence 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 Query, Results, Table, and Report Designer sections, sort lines are drag and drop command lines similar to Request and Limit lines.
- In the Chart, Pivot, and OLAPQuery 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

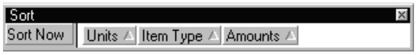
In addition to performing simple sorts, you can use Brio Intelligence to perform complex sorting. Review the following sections for information on:

- 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 Designer 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 Limit lines.



Using this feature, you can drag items to the sort line and request them to be sorted in sequence to yield nested sort 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.

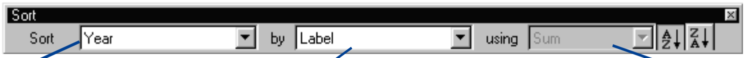
The bottom line effect is the same whether you apply sort conditions locally in the Results or Table section or on the database server in the Query Section. Depending on the situation and the needs of your business, one method or the other may be preferable.

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.

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.



Sort Items

Reference Items

Functions

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 9-2 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, Brio Intelligence 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, Brio Intelligence initially 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 the ascending or descending Sort button on the Sort line.

   The Sort line stores a sort condition for each dimensional item included in the report.

Glossary

Adaptive Report Level The level of privilege that a user can interact with a Brio Intelligence document. The adaptive report levels are View; View and Process; Analyze; Analyze and Process; Query and Analyze; and Data Model, Query, and Analyze. When a document is registered to the OnDemand Server, groups of users are granted access to it with specific Adaptive Report level privileges.

ADR (also called as sync) See *Automatic Distributed Refresh*.

Aggregate Limit Limits placed on aggregated request line items or aggregated meta topic items.

API Application Programmer Interface

API Socket (DaAPISock) The API (such as ODBC, SQL Net, and so on) or protocol level information used when connecting to a database that has been abstracted in an API Socket. Brio has one API socket for each supported API/communication protocols.

Application Server A framework for developing applications that provides fundamental capabilities required by many applications such as session and resource management, and security.

Auditing The monitoring of Brio Intelligence Repository objects to determine usage patterns.

Automatic Distributed Refresh The process of synchronizing locally saved documents and the version in the repository.

Auto-Join An administrator can configure a connection file to cause joins to occur automatically for users using the Auto-Join feature. Auto-Join can be configured one of three ways: Best Guess, Custom, and Server-Defined.

Auto-Process The automatic processing of a query.

Axes Straight lines on a chart used for measurement and categorization. Typically the X-axis and the Z-axis are both used for categories while the Y-axis is used for quantification (for example, Facts/Values). Pie charts only use the X-axis for its categories and the Y-axis for its Facts/Values.

BCS See *Broadcast Server*.

Brio Repository, Brio Intelligence Repository A specific group of tables created using Brio Intelligence on a database server and used to store Brio Intelligence document objects, including data models and standard queries. Administrators can use Brio Intelligence Designer to upload these objects to the repository for end-users to download, providing a template for query and report building. Documents built from repository objects can be version-controlled or audited through a link to the repository.

Brio Intelligence Designer A client/server-based tool that delivers query, analysis, and reporting capabilities and centralized solution administration for developers, database administrators, and system administrators.

Brio Intelligence Explorer A client/server-based tool that delivers query, analysis, and reporting capabilities for power users who need to directly access data sources—or to explore the information organized in prebuilt data models stored in the repository.

Brio Intelligence Navigator A client/server-based tool that provides simplified query, analysis, and reporting for analysts who need more structured access to the data stored in the repository, but need full analysis and reporting functionality.

Broadcast Server A Brio application, residing on a server, that enables users to schedule queries to process on a regular basis or during convenient off-hours, and then distribute the results across a network or to an intranet.

Business Intelligence Organizations utilize information from various operational systems, data marts and/or data warehouses in order to improve business processes, enhance customer service, accelerate sales and increase profitability.

Catalog A collection of database tables and local results. This is the information the user can use in his/her data model or query.

Catalog Pane Shows a list of elements available to the active section. For example, if Query is the active section, the Catalog pane displays a list of database tables. If Pivot is the active section, the Catalog pane displays a list of results columns. If EIS is the active section, the Catalog pane displays a list of embeddable sections, graphic tools, and control tools.

Categories Groupings by which the data is organized (for example, month).

CGI See *Common Gateway Interface*.

Chart A graphical representation of data. Users create charts to convert raw data into eloquent, visual information.

Chart Section With a varied selection of chart types, and a complete arsenal of OLAP tools like group and drill-down, the Chart section is built to support simultaneous graphic reporting and ad-hoc analysis.

Client/Server A network architecture in which each computer or process on the network is either a client (requests information) or a server (delivers requested information).

Clustered Bar Charts Occurs when the categories are viewed side by side within a given category, useful for side by side category analysis. Clustering is only done with vertical bar charts.

Common Facilities One of the four categories of objects defined by the Object Management Architecture (OMA). The Common Facilities is a collection of services that many applications may share, but the services are not as fundamental as Object Services. For example, Print Facility and Mobile Agent Facility.

Common Gateway Interface A native executable launched by the Web server. The role of our CGI is to pass requests from Web clients to ODS and results from ODS back to Web clients.

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 Brio Intelligence in the Results section. They are calculations of new data based on functions, data items and operators provided in the dialog box. They can be included in reports or reused to calculate other data.

Connection File See *Open Catalog Extension Files*.

Controls Folder Contains prebuilt items that can be added to your EIS section, such as list boxes, radio buttons, and command buttons. You can then attach scripts to the controls embedded in the EIS section to execute actions.

Correlated SubQueries Subqueries that are evaluated once for every row in the parent query. A correlated sub query is created by joining a topic item in the subquery with one of the topic items in the parent query.

Cross Join Creates a query where none of the tables is joined. Every row in one table is joined to every row in another table.

Cube The data in OLAP databases (MS Plato, Essbase, MetaCube) is stored in *3-dimensional* cubes, which is different from standard relational databases (*2-dimensional*). Cubes are made up of dimensions and measures. A cube may have dozens of dimensions.

DaConnect (aka DaConn) Brio Intelligence's main connectivity class. Contains all the connectivity specific information. Instance of a connection class is made whenever Brio Intelligence needs to connect to any database.

Daemon-mode BQ The Brio client/server executable. It is spawned in daemon mode by node JREs. The role of the daemon-mode BQ is simply to execute database queries and return results.

DaSession When a client needs to perform an operation such as select, insert, update on a database, a session is created. The life time of a session synchronizes with the operation's duration. Sessions are created when a client needs to perform operations on the database and are destroyed when the operation is considered complete.

Data Function Compute aggregate values, including averages, maximums, counts and other statistics, which 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 totals and items in the other analysis sections.

Data Model A representation of a subset of actual database tables that acts as a menu for the query-builder. Data models are the sources for building the Request, Limit, and Sort lines in a query section. You create a data model by dragging database tables from the Catalog pane Table Catalog into the Content pane. The data model displays database tables graphically as topics when they are in the Content pane.

Data models can be distributed through the Brio repository and used by end users to create their own queries.

Database Function A predefined formula in a database.

Database Server A computer that stores database management system software (DBMS, for example, Oracle, Sybase, Essbase), and a database shared by a network of computer clients. Most databases are used in a client/server environment.

By storing data on a single, powerful machine on a network, the data is centralized and accessible to many users. The server ensures that the data is maintained correctly and serves as a traffic cop to regulate client machine access to the data. The server's computing power is also useful in computing and filtering data from the database before it reaches your workstation. For small or localized databases, your own computer can act as both database server and client.

Datatype The type of data stored in a specific column in a database. For example, data can be stored as a numeric datatype.

Date Group A feature in the Results and Table sections that separates a date into Year, Quarter and Month columns. The display format for the new Month column is automatically set to *mmm* so that the month names sort chronologically (as opposed to alphabetically) in the report sections. Quarters are based on the calendar year, beginning January 1.

Design Mode A toggled environment used to build and modify EIS sections. In Design mode, you build EIS sections with an array of prebuilt control items and graphic items.

Designers can switch out of design mode to test EIS features and deploy them to end users. Design mode is only available in EIS sections. Brio Intelligence Navigator users cannot switch to EIS Design mode.

Detail View Displays a topic as a database table. When Detail view is selected by the user, the database returns ten sample rows from the associated table. Each topic item is displayed as a database field. Detail view enables users to browse a sample of the raw data, which is useful when unfamiliar with the data model or the underlying data. Users cannot view a meta topic in Detail view.

Dimension In an OLAP database cube, categories of information are called dimensions. Examples of dimensions may be Location, Products, Stores, and Time.

In Brio, related, nonquantifiable items in a topic are also referred to as dimensions, such as Contact or Store Name.

Dimension (Legend Dimension) The current axis categorization or grouping method. This can be set to the X, Y, or Z-axis for most chart types. For line and area charts it can only be set to the Y or Z-axis. For pie charts it is always set to the X-axis. When you change the Legend Dimension the color is distributed along the new axis you change to.

Dimension Tab In the Pivot section, the tab that enables you to pivot data between rows and columns.

Dimension Table Consists of numerous attributes about a specific business process. Each row in a dimension table is unique.

Drill Anywhere This feature enables a user to drill into and add items to pivot reports residing in the Results section, without having to return to the Query section or trying to locate the item in the Catalog pane. Drill Anywhere items are broken out as new pivot label items.

Drill to Detail This feature enables a user to retrieve items from a data model that are not in the Results section, without having to rerun the original query. This feature provides the ability to interactively query the database and filter the data that is returned. Drill to Detail sets a limit on the query based on the user's selection and adds the returned value as a new pivot label item automatically.

Drilldown Allows you to progressively narrow your focus on a selected chart category. Very useful when you have too many categories on a particular axis.

EIS See *Executive Information System*.

EIS Home This button returns you to the EIS section that has been designated as the EIS Home section. If you have only one EIS section, then EIS Home will return to that section. If you have several EIS sections, the default EIS Home will be the top EIS section in the Catalog pane. In Design mode you can specify another EIS section to be the EIS Home section.

Embedded Section A Brio Intelligence section that is embedded in a Smart report or other Brio Intelligence section. All embedded sections maintain live data content and formatting from the original section. Reprocessing the query, or modifying the original section will automatically update the display of that embedded section.

Executive Information System Allows users to build and deploy analytic applications. Analytical applications focus on delivering significant prepackaged business content that is encapsulated within an application. With Brio Intelligence, customers and independent software vendors can leverage their knowledge of industry specific best practices and assemble their own analytical applications for deployment to end users using the Web. Combining embeddable Report sections with enhanced EIS controls into a visual workspace and hooking up interactivity between these controls and the native Brio Technology application, an application designer can quickly create a variety of robust applications.

Expression Line Displays displays the JavaScript syntax for each item displayed in a report. Use this line to build equations in the Report section. For ease of use, it can be undocked and resized.

Extranet An intranet that is partially accessible to authorized outsiders. Extranets are secured by user names and passwords.

Fact Table Created to store business activity measures. Most fact tables are extremely large. Each row in a fact table contains numeric measures (fully additive measures, nonadditive measures and/or semiadditive measures) and foreign keys to each dimension table.

Facts The numeric values that are broken up in the body of the Pivot section. To add facts to the Pivot, put Results columns from the Catalog pane into the Outliner's Facts pane. Facts are the numeric values in a relational database that are available to analyze. In an OLAP Query, they are called measures.

Facts/Values The data that is being visually represented, usually a numeric amount (for example, $15,000)

File Server A computer and storage device dedicated to storing files.

Filters In the OLAPQuery section, filters enable you to define and apply limits to the query once Top or Side Labels have been added to the query. You set a filter by applying comparison operators on the values for a specific member. Additional server-specific functions are available based on the selected OLAP database cube. Filters are built to include or exclude data according to specific criteria. Filters can be set on any level in a dimension. However, they must be one of the labels in the Outliner. Filters can be made of members of one of the dimensions or selected by a measure. OLE DB for OLAP has other filter operator types (Top N, Top Sum, Top N %, Bottom N, Bottom N %, Bottom Sum.) Essbase and MetaCube have their own operator types. In Essbase and MetaCube, filters can be done on measures. In OLE DB for OLAP, filters cannot be placed on measures.

Foreign Key A database column or set of columns included in the definition of a referential integrity constraint.

Fully Additive Measure Attributes in a table that can have their values added together across any dimension.

Grain The level of detail at which measures in a table are recorded is referred to as the grain.

Grouping Columns This feature, in the Results and Table sections, creates a new column in a dataset by grouping data from an already existing column. Grouping columns consolidate nonnumeric data values into more general group values and map the group values to a new column in the dataset.

Hardwire Mode In hardwire mode whenever the OLAP Query is changed, the database is queried to fetch new cube data. In contrast, process mode is manually controlled. You add or remove several items to the Outliner, and then press Process to query the database. When determining whether to use hardwire mode or process mode, consider the size of the cube in which you are working.

Hierarchy In an OLAP database cube, a hierarchy organizes a dimension’s levels and corresponding members into parent and child relationships. For the levels in a Location dimension, the hierarchy would have Country as the parent of the child City and City as the parent of the child Address.

Home Abstract base interface that some Lightning interfaces derive from. Provides basic methods to manipulate and create Bean objects.

HTML See *Hypertext Markup Language*.

Hypertext Markup Language A programming language used to create World Wide Web pages, with hyperlinks and tags that explain how to format the information on the screen.

Icon View Icon View shrinks a selected topic to an icon in the Content pane. The topic remains part of the data model, but is deactivated and can not be accessed by the query. Associated items are removed from the Request line when a topic is made into an icon, and the topic is not recognized as joined to other topics. Icon view is helpful in restricting the use of server time when a topic is infrequently used, and does not have to be active at all times.

Imported Files Excel, Text, or CSV files imported into Brio Intelligence. The information in the file goes into a table or a results object. Imported results can be used in other queries (like regular results) for local join purposes.

Indexes Indexes are created in a database to increase the performance of data retrieval. Just as book indexes help to locate specific information faster, database indexes provide a faster access path to table data. Indexes are created on one or more columns of a table.

Insight A Web plug-in that delivers query, analysis, and reporting functionality for intranet, Internet, or extranet access to information. Based on user profiles or report-level security, the client environment adapts in six stages from full query, analysis, and reporting with data refresh to static report viewing.

Integrity Constraints Constraints on tables that guarantee the data adheres to certain business rules. Integrity constraints are defined with a table and are stored as part of the table definition, central to the database data dictionary, so that all database applications adhere to the same set of rules.

Internal Function Internal functions are built-in formulas, defined in the Brio application.

Internet A global network connecting millions of computers. Unlike online services, which are centrally controlled, the Internet is decentralized by design. Each Internet computer, called a host, is independent.

Interval Equal subdivisions within a given scale. The interval can be set manually or to best fit.

Intranet A network belonging to an organization, usually a corporation, accessible only by organization members, employees or other authorization users. Intranet Web sites look and act just like any other Web site, but the firewall surrounding an intranet fends off unauthorized access.

ISAPI This is a shared library that implements HTTP using Microsoft's Web server plug-in API (ISAPI). Thus it is appropriate for customers with Microsoft Web servers. Functionally it is equivalent to our CGI. Its advantage is that it is far more scalable than a CGI executable.

Item An item is a visual representation of a database column and is a member of a topic in the Query section. Items are used to create queries and reports. For example, the Customer Topic may have items including Name, Address, and Phone. You select items from data model topics to build the Request, Limit, and Sort lines in the query section.

Java Server The OnDemand server classes packaged in a single JAR file (`ODSClasses.jar`). This component is responsible for managing, routing, and queuing client requests.

JavaScript The scripting language for Brio Intelligence products. Brio Intelligence 6.x includes the Netscape JavaScript interpreter (version 1.4.) JavaScript and Brio's Object Model allow application developers to use the full functionality of the industry-standard scripting language to control Brio Intelligence applications.

Job Repository A set of database tables which store a queue of scheduled jobs. There can be multiple job repositories in an organization. Job repositories are polled periodically by a Broadcast Server, which downloads and processes jobs when they are due to run.

Join A relational database concept indicating a link between two topics. A join typically occurs between identical or similar items within different topics. Joins allow row records in different tables to be linked on the basis of shared information in a column field. For example, a row record in the Customer table is joined to a related record in the Orders table when the Customer ID value for the record is the same in each table. This allows the order record to be linked with the record of the customer who placed the order. If you request items from unjoined topics, the database server has no way to correlate the information between the two tables and leads to awkward datasets and run-on queries.

Brio displays joins visually in the workspace between topics to indicate joins between database tables. Users can also create new joins which are not already specified in the database.

Join Path A predetermined join configuration for a data model. Administrators create join paths for users to simply select the type of data model needed, in a user-friendly prompt, upon processing a query. Join paths ensure that the correct tables in a complex data model are being used in a query.

JRE (Java Runtime Environment) This is the Java interpreter used to run the Java Server. We ship this as part of our ODS product. Internally we often use the term “JRE” to refer to a running instance of our Java Server.

Legend Box An informative box containing color-keyed labels to identify the data categories of a given dimension.

Level Similar types of members in an OLAP database cube are grouped at the same level. For example, using the members listed in a Location dimension, France, the USA, and Japan belong to the Country level. Paris, Palo Alto, and Tokyo belong to the City level. 35 Main Street belongs to the Address level.

Limit Constraints placed on topic items or request line items to limit them to a certain set of values. Limits appear on the limit line in Brio Intelligence. For example, although the database may display worldwide sales figures for all stores, you may only want to see sales for stores in Germany. Limits make data sets retrieved through a query more efficient and manageable by filtering out unnecessary information.

Limit Joins Joins that are created between a database table and a local results object. The topic item being joined is limited by the values of the column being joined in of the local results object. A limit join is one of the options that can be chosen in a Modify Join operation between a topic item and a local results item.

Linked Data Model Documents that are linked to a master copy in a repository. When changes are made to the master, users are automatically updated with the changes when they connect their duplicate copy to the database.

Load Balancing A technique used to improve the scalability of the OnDemand Server.

Local Computed Meta Topic Items Computed item definitions evaluated by Brio Intelligence results engine. Local computed items are created to be meta topic items. They can be dragged to the request line like regular topic items. The only difference is that the results engine evaluates these items as opposed to the database.

Local Joins A join between a local results object and a database table or another local results object. Brio Intelligence performs the actual join in this case.

Local limits Limits placed on the local dataset in the Results section, as opposed to the Query section. Limits in the query section restrict the data retrieved by the query to the desktop. Local limits screen data from view in the Results set; although it's still there, you cannot see the data that has been excluded or use it in reports unless the limit is removed.

Local Results Results of other queries within the same data model. These results can be dragged into the data model to be used in local joins. Local results are displayed in the catalog when requested.

Locked Data Model Data Models that are locked cannot be modified by a user.

Manager See server.

Master Data Model A Data Model that exists independently and has multiple queries that reference it as a source. When using a master data model, the text "Locked Data Model" appears in the Content pane of the Query section. This means that the data model is linked to the master data model displayed in the Data Model section, which may be hidden by an administrator.

MDX Multi Dimensional eXpressions is the language used to give instructions to OLE DB for OLAP- compliant databases (MS Plato), as SQL is the language used for relational databases. When you build the OLAP query section's Outliner, Brio Intelligence translates your requests into MDX instructions. When you process the query, MDX is sent to the database server. The server returns a collection of records to your desktop that answer your query. *See* SQL.

Measures Numeric values in an OLAP database cube that are available for analysis. Measures may be margin, cost of goods sold, unit sales, budget amount, and so on. *See* Facts.

Members In an OLAP database cube, members are the content values for a dimension. In the location dimension, they could be Palo Alto, Paris, Tokyo, 35 Main Street, USA, France, Japan, and so on. These are all member values for the location dimension.

Metatopic A customized, virtual topic, built from regular topics that reflects the exact topic and item structure of database tables. Metatopics allow items from disparate topics to be consolidated in a single topic, simplifying its appearance and reducing its conceptual resemblance to the underlying database structure. You can choose to view a data model in terms of its original topics, metatopics, or a combination of both.

Metadata Data about data. Stored in database tables, metadata describes the history, content, and function of database tables, columns and joins in understandable business terms. Metadata can overcome the awkward names or ambiguous abbreviations often used in a database. For example, in a table named CUST_OLD, metadata may use a descriptive business name, such as Inactive Customers.

Mime Type A browser mapping of a file type to either a helper application or a plug-in. When a browser attempts to open a file of a particular mime type, it either loads the associated plug-in or launches the associated helper application. A file's mime type is determined either by a) the file extension or b) the HTTP header. Plug-ins tell browsers what mime types they support and what file extensions correspond to that mime type.

Brio Web clients support the following mime types: *application/x-brioquery* mime type (for `.bqy` files). This is the default mime type our Web clients support and are ordinary Brio files. *application/x-brioquerydata* (for `.bqd` files). These are data files in text or Excel format, whose extension has been changed to `.bqd`. When a Brio Web client is launched to open a BQD file, it imports the data and executes any BrioScript or JavaScript the file contains.

Morphing Mechanism by which Brio Web clients provide document security. See *Adaptive Report Level*.

Multidimensional Database A database that stores data in a format often referred to as a cube, such Essbase, MS OLAP, MetaCube, and so on. *See also* Relational database and OLAP database.

Nonadditive Measure Attributes in a table that cannot be added across any dimension, such as a percentage value (for example, margin rate).

NSAPI This is a shared library that implements HTTP using Netscape's Web server plug-in API (NSAPI). It is equivalent to our CGI. Its advantage is that it is far more scalable than a CGI executable.

Null Value A null value is absent of data.

OCE See *Open Catalog Extension Files*.

OCE Wizard Wizard or set of screens used to create a new OCE or modify an existing OCE.

ODS See *OnDemand Server*.

OLAP Database A database that stores its information in cubes. Cubes contain dimensions and measures. A cube may have dozens of dimensions. Cubes are built to hold aggregated data, which anticipate how users think about business models. Cubes deliver this information efficiently and quickly.

OLAPQuery Section Analyzes and interacts with data stored in an OLAP cube. When you use Brio Intelligence to connect to an OLAP cube, the document immediately opens an OLAP Query section. The OLAPQuery section displays the structure of the cube as a hierarchical tree in the Catalog pane. Queries are built by dragging measures and dimension levels or members directly into the Outliner panes.

OnDemand Server A Brio server application that enables users to view and select from a list of available documents over the Web, as well as to build and process new queries.

OOA Object Oriented Analysis.

OOA&D Object-Oriented Analysis & Design.

Open Catalog Extension Files Encapsulate and store connection information used to connect Brio applications to a database. OCE files specify the database API (ODBC, SQL*Net, etc.), database software, the network address of the database server, and your database user name. Once created, a user can specify the OCE file and database password and logon. An OCE file is required for a Brio Intelligence document to use a database. the file extension is `.oce`.

Open Client DBLib API to connect to Sybase, Redbrick, SQL Server, and so on.

Open Metadata Interpreter The Tables, Columns, Joins, Lookups and Remarks tabs available in the connection wizard when you edit a custom metadata source. These tabs allow Brio administrators to specify a customer source of metadata that can be accessed through SQL statements, and provided to end users with data models.

Outliners Drag-and-drop command lines used in the Pivot, Chart, OLAPQuery and Report sections. Each Outliner pane corresponds to a specific layout element of the report. When an item is dragged to an Outliner pane, the item assumes the layout attributes of the respective report element. Data appears simultaneously in the Content pane with the appropriate formatting.

Pivot Dimension A row or column of labels that corresponds to an item in the Catalog pane.

Pivot Section The Pivot section is used to create crosstab reports and analyze data.

Pivot Table Analytical tools that resemble spreadsheets or crosstabular reports. A pivot table overlays a dynamic datacube, which allows data to be sliced and diced for ad-hoc, interactive, and multidimensional analysis.

Pivoting In the Pivot section, pivoting is the ability to change a label from a top to a side (or a side to a top) orientation with a simple click and swing of the label's Dimension tab.

Plot Area The area bounded by the X-, Y-, and Z-axis. For pie charts, it is the rectangular area immediately surrounding the pie.

Plug-in A special application file placed in the browser Plug-in directory. Plug-ins add seamless functionality to a Web browser, enabling the browser to open particular plug-in file types.

Plug-in / Helper The two implementations we support for our Web clients. The terms are short for browser plug-in and helper application respectively. The plug-in implementation is a shared library which the browser loads when the user requests a document of the application/x-brioquery mime type. The helper implementation is an application launched by the browser when that same mime type is requested. The plug-in is generally considered a *better* implementation because it runs within the browser. Brio provides helper applications on all platforms (Windows, Mac and Unix) and plug-ins on Windows only.

Predefined Drill Paths Enables a user to drill directly to the next level of detail, as defined in the data model.

Primary Key A database column or set of columns included in the table definition of the PRIMARY KEY constraint. Primary key values uniquely identify the rows in a table. Only one primary key is defined per table.

Query A query is set of database instructions to return an answer set to a specific question. Each row returned in the Results section of a document is an answer to the question posed in the Query section.

Query Computed Items Item definitions created by the user. This can include other request line items or topic items and or database functions. The definition is sent to the database and the database evaluates them.

Query Log Log of all SQL statements sent to the database (also referred to as SQL Log).

Quickview A Web plug-in that offers simplified report viewing and data refresh for users who need to view published, formatted reports within their browser.

Relational Database A database that stores its information in tables that are related or joined to each other by common pieces of information called keys. Relational databases store information in tables. A table is subdivided into column fields. Related information is grouped in column fields. Column fields have parents and children. For example, the Customer table may have columns including Name, Address, and ID number. Each table contains row records that describe information about a singular entity, object, or event, such as a person, product, or transaction. Row records are segmented by column fields. Rows contain the data that you retrieve from the database. Database tables are linked by Joins. (*See also Join*)

Report Group In the report section, embedded reports and tables are grouped by other data items. Items placed in the Groups Outliner break information into these dimensional groupings. For example, your table may include the name, contact information, and sales for each of your distributors. This table gains in clarity when broken into groupings that classify the stores by geographical region, year, or both.

Report Section A dynamic, analytical report writer, that provides users with complex report layouts and easy to use report building tools. Pivot tables, tables, and charts can be embedded in a report. The report structure is divided into group headers and body areas, with each body area containing a table of data. Tables are created with dimension columns and fact columns. These tables are elastic structures. Multiple tables can be ported into each band, each originating from the same or different result sets.

Repository Central location used to store data models, queries and queries with reports. Repository is usually a database chosen by the user.

Request Line Holds the list of items requested from the database server and that will appear in the user's results.

Request Line Items Columns listed in the request line.

Results Section A section in a Brio Intelligence document that contains the dataset derived from a query. Data is massaged in the Results section for use in the report sections.

Run Mode A toggled environment used to test EIS sections. It simulates a Navigator user's view of the section. In Run mode, you can not add any features, but you can use features that are part of the deployed EIS sections. *See also Executive Information System* and *Design Mode*.

Scale The range of values that allows one to gauge how much each category represents. This range can be either at equal intervals or at logarithmic interval. The scale can be set manually or to best fit.

Script A series of instructions for a computer. Scripts are activated when an event occurs, such as clicking a button or selecting an item from a drop down list. Brio Intelligence's scripting language is JavaScript.

Section Pane Lists all the sections that are available in the current Brio Intelligence document.

Section Title Bar A navigational aid under the toolbars that provides a means of moving between sections and toggling section-specific tools and gadgets, such as the outliner.

Semiadditive Measure Attributes in a table that can be summarized across some dimensions, but not all.

Server Administrator A stand-alone executable for administering the Brio OnDemand Server and the Brio Broadcast Server.

Server Components The components that make up the OnDemand Server. These include the Web Broker, the ODS Manager, the ODS Node(s), and the ODS Repository (ies).

Server Computed Meta Topic Items Metatopic item definitions created by the user. These can use any of the database functions available. These items can also use any of the other topic items in the data model. These items are evaluated by the database.

Session Socket (DaSessionSocket) Session information specific to each database or API is abstracted in a session socket. We have session sockets for each of the APIs we support.

Simple Join Retrieves rows to create a query where the values in joined columns match.

Slicer An axis that filters the data in an OLAPQuery. Only individual members can be used in a slicer. A slicer can be thought of as a third axis in a OLAP Query. The other axis are the Side Labels and the Top Labels. Every dimension folder contains a members subfolder named “Values for…” that dimension. This subfolder contains the members that are eligible for the slicer.

Snapshot A read-only table snapshot is a local copy of table data that originates from one or more remote master tables.

Sort Conditions placed on request line items to sort the results in ascending or descending order. These are displayed in the sort line in Brio Intelligence.

SQL *See Structured Query Language.*

SQL Net Oracle’s native API to connect to an Oracle database.

Stacked Charts A chart where the categories are viewed on top of one another for visual comparison. This type of chart is useful for subcategorizing within the current category. Stacking can be used from the Y- and Z-axis in all chart types except pie and line. When stacking charts the Z-axis is used as the Fact/Values axis.

Stored Procedure Precoded queries in languages other than SQL. This is a feature available in some database software. Brio Intelligence can run stored procedures and retrieve the Results.

Structure View Displays a topic as a list of component items allowing users to see and quickly select individual data items. Structure view is the default view setting.

Structured Query Language The language used to give instructions to relational databases. When you build the query section’s Request, Limit, and Sort lines, Brio Intelligence translates your requests into SQL instructions. When you process the query, the SQL instructions are sent to the database server. The server returns a collection of records to your desktop that answers your query. This reply is displayed as the Results section. You can look at the SQL generated by a query in the Query Log, or code a query directly in SQL using the Custom SQL window.

Subquery A query embedded within another query.

Surface Values A setting in the Pivot section to base aggregate calculations on the values in the report, rather than the values in the Results section.

Synonym An alias for a database table or view. It is a direct reference to a table view.

Table The basic unit of data storage in a database. Database tables hold all of the user-accessible data. Table data is stored in rows and columns.

Table Catalog Displays tables, views, and synonyms to which users have access. Users drag tables from the Table catalog to the Content pane to create data models in the Query section.

Table Section Used to create tabular-style reports. It is identical in functionality to the Results section, including grain level (table reports are not aggregated). Other reports can stem from a Table section.

Top and Side Labels In the Pivot section, labels are the column and row headings on the top and sides of the pivot. These define categories by which the numeric values are organized.

Topic A topic is a visual representation of a database table in the Content pane. Topics are part of data models displayed in the Query section and can contain one or more items.

Topic Items Individual items in a topic or metatopic.

Topics Visual representation of tables in the database, related by joins that link certain items in each topic. Each topic title bar displays the topic's name. The topic shows a list of items, one for each column in the database table.

Underlying Values Another name for Results values. When Use Surface Values is disabled in a Pivot section, aggregate calculations are based on values in the Results section.

Union/Intersection/Minus Queries Queries created to perform set operations such as Union, Intersection, and Minus. These queries are created by the Append Query option.

Variable Limits Limits that prompt users to enter or select limit values before the queries are processed on the database.

View A custom-tailored presentation of the data in one or more database tables. Views do not actually contain or store data; rather, they derive their data from the tables on which they are based, referred to as the base tables of the views.

Web Clients The Web-enabled counterparts of our client/server application (Brio Query). See *Insight* and *Quickview*.

Web Components Web server components of the OnDemand Server are Web Broker, HTML pages without ODS tags, and the Zero Administration Web client installers. See *OnDemand Server*.

World Wide Web A system of Internet servers that support specially formatted documents. The documents are formatted in a language called HTML (HyperText Markup Language) that supports links to other documents, as well as graphics, audio, and video files.

WWW See *World Wide Web*.

Zero Administration Installs Brio Web applications (Insight and Quickview) automatically without the help of an administrator when a user logs on to the OnDemand Server.

Index