



How'd You Do That?

4th Edition

Brio Intelligence™
Tips and Tricks
for
Pivots, Charts and Reports

Mark Ostroff
Principal Systems Consultant
Brio Software™

About the speaker

Mark Ostroff has over 20 years experience in the computer industry. He began programming medical research data acquisition systems in Assembly Language and Fortran at the Johns Hopkins School of Hygiene and Public Health using a PDP-12 mini-computer. He also helped develop microcomputer device interfaces and database systems for use in the process control industry. Mark then moved to building PC database applications using dBASE and Clipper. These systems performed systems configuration management for the U.S. Navy, and project tracking at IBM's COS Division.

Mark moved to the vendor side of the software industry when he joined Borland in 1989. Mark worked at Borland as a Systems Engineer for nearly seven years. While employed at Borland, Mark presented several times at the annual Borland Developer's Conference, at Comdex and at numerous Delphi Client/Server seminars. In addition, he was one of the volunteers who helped create the original Paradox-based "InTouch System" for the Friends of the Vietnam Veteran's Memorial.

Mr. Ostroff has written numerous articles for the Paradox, Delphi and Web Informant magazines. Mark is a certified Delphi developer and trainer. He is also a co-author of the book, *Delphi in Depth*, from Osbourne/McGraw-Hill.

Mark started at Brio Software in June 1997. He is currently employed as a Systems Consultant working primarily with Federal Government accounts. He was the original author of the Tips and Tricks column for the BrioCast, Brio's original electronic customer newsletter. Mark was also a speaker at all previous Brio Users Conferences. He presented the original Tips and Tricks cookbook for Pivots and Charts in 1999, followed by yearly updates to the Tips Cookbook as well as other numerous sessions. These included the Planning a Smooth Migration to v6.x and the BQY Performance Optimization white papers.

Each year, Mark has expanded his "Tips and Tricks Cookbook" to include more tips. A summary of this progression follows.

Brio Conference	Brio Intelligence Versions	Number of Tips
1999	v5.0 through v5.5.6	13
2000	v6.0 through v6.2.3	22
2001	v6.0 through v6.5.0	34
2002	v6.0 through v6.6.1	47

1 Introduction

These tips are designed to help you get the most out of Brio Intelligence's analysis features. They are based solely on Brio Intelligence™ v6.0 and higher, and so will not work with v5.x Brio clients. For tips specific to older clients, please see 1999's Advanced Pivots and Charts presentation.

Some of the v5.x tips are now replaced with newer features in Brio Intelligence v6.x that make them easier or more flexible. However, some of these v5.x tips still apply to Brio Intelligence v6.x. Updated versions of these older tips are included in this cookbook for completeness. The v5.x tips that still apply include:

- Creating Range-based Analysis
- Creating multi-contingency grouping / range columns
- Spotlighting in a Chart
- Charting Portion within a Category
- Demographic Analysis
- Delta Trends
- Counting the number of stores in each state and the sum of all state revenues.

In some cases, a v5.x tip will still work, but newer features in the v6.x product provide another way to accomplish the same result. In these cases, both methods will be listed.

In this "Tips and Tricks" paper, you will find a set of basic concepts, followed by specific tips laid out as a cookbook of individual "recipe" items. I used this cookbook format to let you bounce around and look for specific items you may need at a given moment.

2 Basic Concepts

The idea behind this session is to provide you with more than just a set of working tips. I also hope to impart some of the basic thought processes that went into these tips. That way, you will be able to expand on what is presented here and create your own tips and enhancements.

Be Creatively Lazy

Find ways to let the computer do a lot of the work. Think globally. Think about creating reusable “components”. And think about what is the best processing location for the particular functionality you need to use.

Plan Your Course

Before you start developing pivots, charts or reports themselves, take a step back and think about what you are trying to accomplish. You may change the way you develop your reports entirely once you have gone through this process first.

Take the 30,000 Ft. View

Look at your analysis requirements from a business perspective, rather than the initial specifics of the pivot or chart you want to end up with. Business analytics involves gaining business value, not just reproducing existing management reports. Your design planning should take into account *why* the existing management reports are inadequate, and provide a solution that solves those problems as well as providing the actual reports. Use this opportunity to add value, not just recreate the same paper reports in an electronic form.

Static versus Analytic Reports

Part of the difference between older style solutions and a Brio solution is the ability to provide live, analytic reports. As such, you might need to look at some of the new capabilities in Brio Intelligence v6.x that give you more flexibility in this area.

For example, do you want to take advantage of the new drill-to-detail and incremental drilling features? Do you want to create a live interactive data dashboard of Key Performance Indicators to replace or enhance the existing management reports? Will the new SmartReport enhancements enable you to use embedded charts or pivots in a Report section to produce something you've always wanted to do but couldn't in prior versions? The answers to these types of questions will drastically affect how you develop your charts and pivots.

User Navigation Control

How much control do you want your user to have in navigating through these charts, pivots and reports? You might want to control how users can drill in the chart or pivot. Will they use drill anywhere or a pre-defined drill path? Can they drill into further detail? Do you want to provide double-click incremental drilling in your charts?

If you want to exercise even more control over your users, then you might want to embed these items in an EIS or Report section and hide the original charts and pivots altogether. Embedded items need to be designed slightly differently than items that will be directly manipulated by the user.

Do you want to use an EIS section to limit the value choices a user can select, or to provide a customized limit interface? You may need to make slight modifications to the way you build your query sections to support that design.

What do you need for the Trip?

Remember you are taking your users on a data journey. Like any good trip, you need to make sure you “pack” everything you’ll need. Once you have figured out the general usage plan for your reports, you need to analyze the specific BQY requirements to produce those results. Several specific areas you should consider are covered here.

Data Requirements

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

Raw versus “Crafted” Data

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

Creating reusable “components”

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

Cross-row comparisons

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

Grouping the data to provide ad-hoc organization

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

Transforming the data into a more useful form

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

Preparing for expanding the data's use

For example, say you want to create a Pivot with a count of the number of transactions for each item. You could drop the Customer_ID field into the data items area in the Pivot Outliner, right-click on the column and change the data function to be Count. Now, you want to add the dollar amounts and then expand the use of the Customer_ID count to calculate the average per transaction. In building the formula for the average, the Pivot will revert to the default data function of summing the Customer_ID as if it were a number to be added instead of an ID to be counted. This will not yield the correct average.

Back up a step to the Results section instead. Use a computed item to "translate" the Customer_ID into a true count. Simply define the formula as a 1. Then you can use this numeric field directly in the pivot to compute the correct number of transactions *and* as part of the formula to yield the *correct* average per transaction.

Where to Consolidate

The proper use of Computed Items in the Results section depends on understanding how you want to consolidate your information. If the consolidation is based on the "raw" data, use the power of the Results section.

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

Local versus Server Processing

People often overlook the power of their database to serve their reporting and analysis purposes. For example, one retailer wanted to graph the number and dollar amount of transactions per department on an hour-by-hour basis. They wanted to use this chart to estimate required staffing levels throughout the day based on actual sales history.

The original query processing and chart generation took over 2 hours. This kind of response was clearly unacceptable. Two factors contributed to the amount of processing time. First, of course, they were fetching every single record of sales data for the given period, amounting to more than 100,000 rows. Then, Brio had to process all those rows to produce the graphical consolidation that was the desired end result.

As a general rule, anything you do on the Query section is processed at the database. Actions performed on any other section are executed by Brio itself. I showed them how they could speed up the processing by having their database perform the consolidation of the data instead of having Brio do it. They only had to make a minor change to the query.

By default, Brio creates a simple Select statement from the Request line in the Query section. You can change this behavior by selecting the value(s) on the Request Line you want to consolidate, right-clicking and selecting the Data Function item from the speed menu which appears.

In this case, I had the retailer select the Sum function. This time, the entire process returned 52 rows and was finished in less than one minute.

Data Presentation Requirements

You also need to look at how you will present the data to your users. Drilling options will determine how you build the data model. Human factors engineering concepts affect which pivot or chart display options you use.

Drilling Requirements

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

Flattening the Chart

Sometimes you can influence the data's interpretation by the way in which it is presented visually to the user. A basic charting concept to leverage is to realize the visual difference between a 2-D and 3-D graphical presentation. While 3-D mode is the default for Brio charts, certain analysis tasks can be performed by first creating a 3-D chart, then flattening the chart to 2-D to create a layering effect.

The human mind will interpret the flattened result in a far different manner. You can create inferences about the data in 2-D that aren't easily seen (if at all) in 3-D. Last year's tips cookbook showed how you can use this difference to create spotlighting in charts and to show portions of the whole in multiple categories at once.

Embedding Requirements

Charts and pivots intended for embedding should be designed differently than items intended for direct manipulation. The unique requirements of each type of embedding need to be accounted for in your design.

Embedding in an EIS section

Items to be embedded in an EIS section need to use short descriptions to provide meaningful information in a smaller display. Charts should not use auto-sizing, since your EIS item's size should control the scaling of the chart elements. Users will not be able to change the size of items drilled to in embedded live pivots, so label length is of key importance in these pivots. You may want to use a combination of computed items in your data model along with pre-defined drill paths to keep the display readable.

Embedding in a Report section

Items used for SmartReport embedding in a Report section will probably be designed with exactly the opposite characteristics in mind. Realize that a chart or pivot embedded into a Report will become context-sensitive. It will only display the information that pertains to the report grouping within which the item is embedded. Thus, you will want to create a "global" pivot or chart with many more elements than you would display in a directly displayed item. Don't worry about what the raw chart or pivot looks like. Design for what it will narrow down to when embedded in your report.

Scripting Requirements

Do you plan to use an EIS section to control what gets displayed? If so, you may need to design things a bit differently to support your scripting requirements. For example, it is easy in a chart or pivot to display only the top 5 items. But if you want the user to select how many items will display, you may want to provide a simple EIS screen to automate the process. To pull this off, you will need to create a Table section and build your chart or pivot off that Table section.

Why do you need a Table section? To code how many items get displayed, you need to be able to set a limit on the Rank. However, Brio doesn't let you create a limit on a column based on an aggregate function. The Table section lets you convert that aggregate column into a plain numeric one. Thus, the Table section will be required to create a limit on the rank. It's the only way to provide programmatic control of how many of the top items get displayed.

Prepare Your Data

As was mentioned before, you will probably need to craft your data in some way before building your charts and pivots. You need to select which data crafting operations you will perform in which of three possible section types - the Query section, the Results section and (occasionally) one or more Table sections. Each type of section is appropriate for performing different types of data crafting tasks.

The Query Section

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

The Results Section

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

- **Default Labels**
The column labels that appear in the Results section will be the default labels for all charts, pivots and reports built from this data. You can set these global default labels in either the Query or Results section.
- **Default Formats**
Like column labels, the display format used in the Results section will be the default formatting for all charts, pivots and reports built from this data. These global default formats can only be set in the Results section.
- **Global Computed Items**
Computed Items that need to appear in more than one section should be defined here in the Results section.
- **Global Comparisons**
Like global computed items, comparisons that need to apply across multiple sections should be defined once in the Results section.
- **Global Groupings**
Groupings that apply across multiple sections are just like other global items. They should be defined in the Results section.

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

Table Sections

The Results section is actually a special, global type of Table section. The new Table section provides a way to built items that are less global than ones built within the Results section. It can also be used as a data staging area to enable various functions not possible before.

Semi-Global Items

A Table section can be used to create semi-global items using the same techniques used in the Results section to create global items. Items created in a Table section will only apply to those sections built from that Table section. In general, sections built using a Table section are isolated from certain changes that occur in the Results section that feeds the Table section.

Data Staging

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

Top N / Bottom N Analysis

Allowing users to easily select how many items to display usually involves creating a limit. However, Brio will not allow a limit to be created on a Computed Item based on an aggregate calculation. Rank and Sum are such aggregate calculations. Enter the Table section. When a Computed Item column is added to a Table section, it is converted to the actual value derived from the computation. Since it is now a simple numeric value, a limit can now be applied to the Rank or Sum value in the Table section.

Report Section Refresh Control

The older v5.x Detail section carried its own version of a data cache specific to that report. Thus, v5.x documents with multiple Detail sections carried a lot of redundant data with them. To reduce the size of BQY files that contained multiple reports, the new Report section is designed to work off the Results or Table sections. However, since they do not carry their own data cache, Report sections do not offer any Refresh Control options. You can restore this Detail section feature by using a Table section as the basis for your Report. You then create a dummy limit on the Table section to artificially start with zero rows in the display. When you want to refresh the Report section, simply set the dummy limit to "Ignore" (either manually or in JavaScript code).

Specific Design Considerations

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

Design Considerations for Charts

Available design choices for chart sections include the following:

- **Global vs. Local Computed Items**
Think about the best place to create Computed Items before starting the Chart design. Remember that items computed locally in the Chart section only have available to them those fields that have been added to the Chart Outliner. Globally computed items can even include database server fields that are not included in the query itself (by using server-based computed items).
- **Starting “raw” vs. starting with a Pivot**
Some charts may be easier to create by starting with a pivot and then selecting the “Chart this Pivot” menu item from the Insert menu.
- **2-D vs. 3-D**
Weigh the different effects each chart type has on the interpretation of the data.
- **Direct use vs. Embedded use**
Remember that charts destined for embedding should be designed with that purpose in mind. If you need two similar charts, one for embedding and one for direct use, create one chart, then duplicate it.

Design Considerations for Pivots

Design choices for pivot sections include the following:

- **Global vs. Local Computed Items**
The same issues that pertain to computed items for charts apply to pivot sections as well.
- **Starting “raw” vs. starting with a Chart**
Like pivots, starting with another section can be an easy way to create charts. In addition to making the creation of charts easier, it also allows for the creation of multiple “views” on the same data. This facilitates a broader range of data interpretation possibilities.
- **Using base values vs. using Surface values**
By default, a pivot section reaches back to the Results or Tables section from which it was created for all computations. This can sometimes create what may look like strange values for certain computations.

For example, if you add the same fact a second time, you can right-click on the duplicate column and change the data function from “Sum” (the default) to something else, like “Max”. By default, the pivot will reach back to the original data rows and display the maximum single row value. This may not be what you wanted displayed. If you instead wanted to display the maximum sum as displayed on the pivot section itself, you will need to switch the pivot to “Use Surface Values” mode. Aggregate calculations will then look at the summarized data as it appears in the pivot section to perform the calculations.

- **Direct use vs. Embedded use**
The same considerations that apply to charts apply to pivots as well.

Design Considerations for Reports

Use your imagination to think beyond simple reports. Available design choices for report sections include the following:

- **Table vs. Free-form reporting**
Different design techniques are used to create these two fundamentally different report styles.
- **Global vs. Local Computed Items**
In addition to the considerations mentioned for chart and pivot sections, report sections add a wrinkle all their own. In migrating from the v5.x Detail section, Brio customers wanted a more powerful reporting tool. Brio delivered the requested power in the new v6.x Report section.

However, to provide the requested power, some fundamental changes were made in the way fields operated in the report designer. If you click on a data field in a report and look in the Expression bar, you will see that all items in the report section are actually short embedded JavaScript routines. This provides report designers a huge amount of power, but it means that computed items defined in the report section will require a different syntax than globally computed items that reside in the Query, Results or Table sections. You will want to weigh the relative merits of simplicity versus power when deciding where to define computed items to be used in report sections.
- **Creating tables from scratch vs. Embedding a Table section**
One easy way to create table-oriented reports with easily created computed items is to embed an existing Table section instead of designing a table object via the report section Outliner.
- **Single Column vs. Multiple Columns**
The new report section supports creation of multi-column reports.
- **Headers & Footers for Reports, Pages, Groups and Sub-Groups**
Remember that only headers for Groups and Sub-Groups are created automatically. You will need to specify all other headers and footers if you want to use them.

- **Using the full power of the SmartReports feature**

The SmartReports feature can do a whole lot more than embed charts and pivots into a report. Let your imagination run wild with the possibilities. Some ideas include:

- Printing multiple charts and pivots on 1 page
- Creating context-sensitive charts and pivots
- Alternate reporting styles
- Alternate ways to create computations

- **Consider using an EIS Section as a “power report”**

An EIS section with embedded charts, pivots and tables can provide an interactive reporting environment where users can select various parameter values to easily change the data view in the “report”.

Locking in the Knowledge

In addition to learning these techniques yourself, you’ll want to implement a plan of action to spread these ideas throughout your entire Brio user community.

Teach the Thought Process

Most tips result from the application of few simple basic techniques. Help your users understand the logic behind these techniques, not just the specific solutions themselves.

Take a Step Back

Help them focus on the basic design process first. For example, you want to get them thinking about crafting their data in the Query or Results sections before they try to build their charts or pivots. You also want them thinking beyond simple chart and pivot design and into how others could use these sections better.

1 + 1 > 2 (or, “The whole is greater than the sum of the parts”)

Help your users to realize that by focusing on basic techniques above specific point solutions, they will be able to go beyond those specific solutions the next time they need to design something. These basic techniques allow your designers to go far beyond any specific solution. By adding them together, they can produce documents with far greater power and usefulness.

Deploy the Techniques

Create some kind of automated mechanism to “get the word out” to your users. You could create an internal technology newsletter in either hard copy or electronic form. Internal web-based news groups and named email distribution lists are also effective ways to deploy the information in this session.

A more personal way of sharing information, as well as getting people more involved, is to form an internal users group. You could use the material in these session notes as the basis for your first one or two meetings. As the group grows in size, each meeting could have a section devoted to sharing new tips and ideas.

Solicit Additional Tips

Make sure you set up a way to keep feeding your knowledge base with new tip ideas. Get creative with incentives to reward useful tips. Have various users submit topic ideas for your next users group meeting.

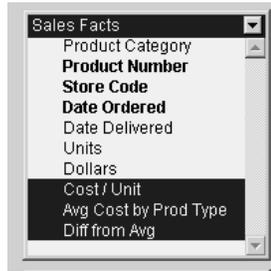
Build in Regular Feedback

Keep the loop going. Present any tips sent in to the rest of the user community through the deployment mechanisms you have created.

And now, on to the tips recipes...

Section Type: Query section

Tip Name: Creating Read-Only Computed Items

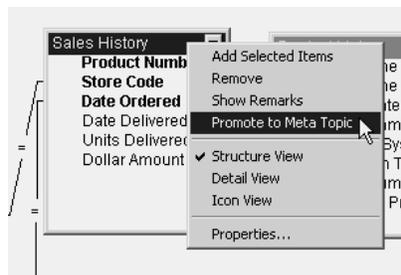


Purpose / Use: To provide users with standard computed items that they can use but not modify. This insures that all users have common computational definitions.

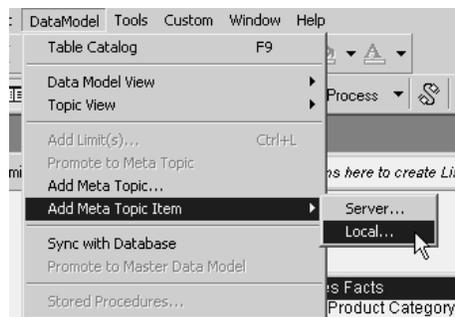
Sample File: Read-only computed items.bqy

Steps to Produce: To create read-only computations, you will build one or more computed items pre-defined in the query's data model, and then promote the query's data model to a read-only locked Master Data Model.

- 1) Build a regular data model as you normally would.
- 2) Right-click on each topic in the title bar and promote it to be a meta-topic. (Computed items in a data model can only be added to a meta-topic.)



- 3) Rename each meta-topic (if desired) by double-clicking on the meta-topic's title bar and changing the Topic Name property.
- 4) Select the meta-topic to which you want to add the computed items, then use the DataModel | Add Meta Topic Item menu selection to define the new computed item(s).



(Creating Read-Only Computed Items continued)

As shown in the previous screen shot, meta-topic computed items can be either server-based or locally computed. Server items are based on what can be computed via SQL, and are computed at the database server. Local items can use the full power of Brio's JavaScript interpreter, but require all items in the computation to be included on the Request Line. If a required field is not already present on the Request Line, adding the local computed item to the Request Line will automatically add all fields needed by the computation.

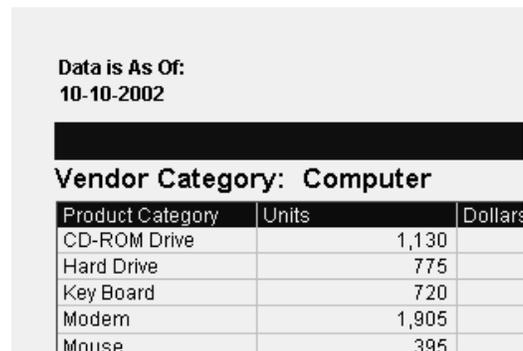
NOTE: To define a local meta-topic computed item, all columns referenced by the computed item's formula must be present in the same meta-topic as the computed item. You can drag any fields needed from other tables into the meta-topic to support multi-table computed items. You may want to then hide that field from the topic in which it normally resides.

- 5) Once all computed items are defined, use the DataModel | DataModel View | Meta menu selection to set the display to show only the meta-topics.
- 6) To make the items read-only, promote the query's data model to be a Master DataModel by using the DataModel | Promote to Master DataModel menu selection.

NOTE: To keep the Master DataModel locked for ODS users, deploy the BQY in "Query and Analyze" mode (not in "DataModel and Analyze" mode).

Section Type: Query section

Tip Name: Date-Stamping a Query



The screenshot shows a report header with the text "Data is As Of: 10-10-2002" and a filter "Vendor Category: Computer". Below this is a table with three columns: "Product Category", "Units", and "Dollars". The table contains the following data:

Product Category	Units	Dollars
CD-ROM Drive	1,130	
Hard Drive	775	
Key Board	720	
Modem	1,905	
Mouse	395	

Purpose / Use: To provide information in reports that allow users to know when the data was fetched from the database, rather than when the report is viewed or printed.

Sample File: Date stamping.bqy

Steps to Produce: Add a server-based Computed Item to the query that adds the system date and time to the columns added to the Results set.

- 1) Build a data model and query as you normally would.
- 2) Right-click on the Request Line of the query, and select "Add Computed Item" from the speed menu.
- 3) Name the Computed Item "As Of" in the Name field in the dialog box that appears.
- 4) Click on the Functions... button in the dialog box. Select "Date Functions" from the Function Categories column, and then select the appropriate date/time function for your database connection.
- 5) To use the "As Of" field, simply drag it into the desired Chart, Pivot or Report area.

Section Type: Query section

Tip Name: "What's Not" Query

Enter Store ID:

These reports show data for Store ID: 5

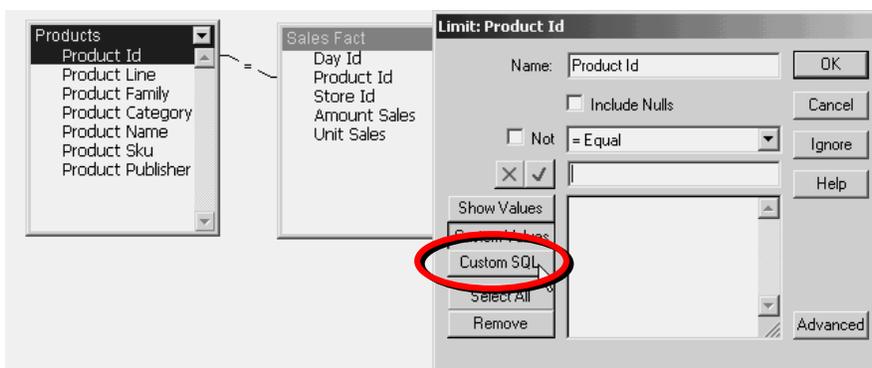
Products Sold with Sales :				Products Not Sold:	
Store Id	Product Id	Product Name	Amount Sales	Product Id	
5	5	Brilliant Deductions	6,022.14	1	The New Way T
	9	Thomas the Tank Engine: Complete Set	15,386.69	2	Five Brothers: A
	12	Java 1.1 Certification Study Guide	6,701.8	3	Autumn: Recipt
	21	The 48 Laws of Power	8,533.91	4	Motley Fool Inve
				6	The Rich Die R

Purpose / Use: To find out from the database what did *not* happen.

Sample File: Not sold.bqy

Steps to Produce: Add a limit based on a "not in" sub-query select.

- 1) Build a data model that includes one or more tables that would normally contain the absent information.
- 2) Add a limit on the primary key join field by either double-clicking on the field, or dragging it into the Limit Line.
- 3) In the Limit dialog box, click on the "Custom SQL" button.



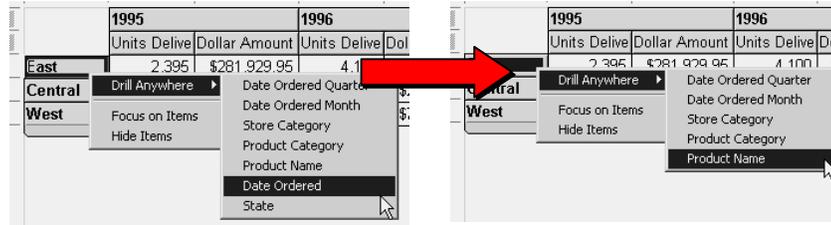
- 4) Enter the SQL for a "not in" query that tells the database to fetch data from the primary table where the linking field does *not* appear in the subordinate table. For example:

```
Products.Product_Id not in (select product_id from sales_fact)
```

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

Section Type: Query section

Tip Name: Controlling the Drill Path

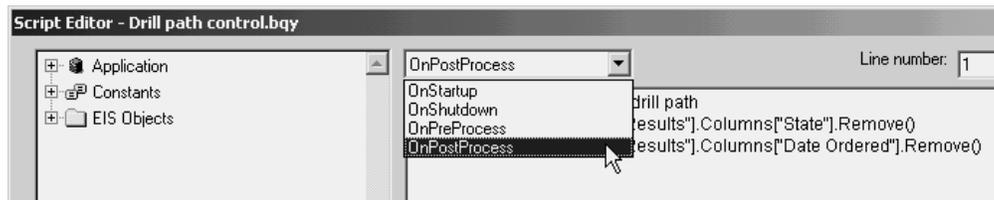


Purpose / Use: Sometimes you want to only show certain fields in a drill path, especially when some fields are only used to build computed items and you don't want them included in drilling or even visible to users. In addition, you may not want to use a Table section, since this would duplicate data.

Sample File: Drill path control.bqy

Steps to Produce: Add simple JavaScript code to the OnPostProcess and OnPreProcess document events to remove the undesired fields from the Results section after the query has processed and all computed items have been calculated. (Requires Brio Intelligence v6.5.0 or higher.)

- 1) Build a data model and query as you normally would.
- 2) Use the File | Document Scripts... menu selection to enter the script editor for the BQY document.
- 3) Select the OnPostProcess event from the pulldown menu at the top.



- 4) Enter code to remove the fields you want to hide from the Results section after the query finishes processing. For example:

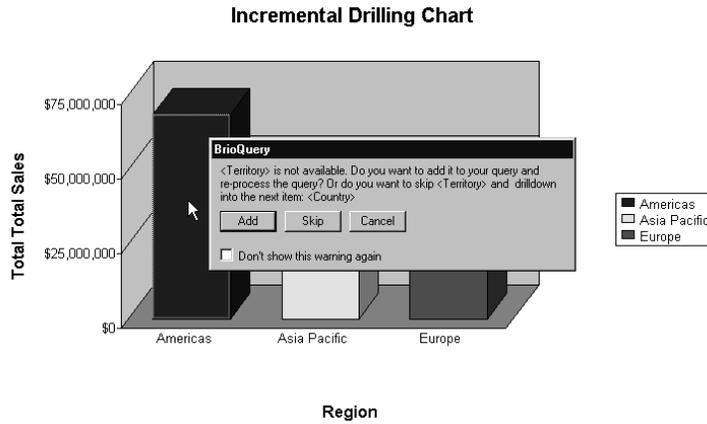
```
// Eliminate fields from the drill path
ActiveDocument.Sections["Results"].Columns["State"].Remove()
ActiveDocument.Sections["Results"].Columns["Date Ordered"].Remove()
```

- 5) Now select the OnPreProcess event and enter code to put the fields back into the Results section just before processing the query. That way, any dependent computed items will be properly refreshed. The corresponding code for the current example would be:

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

Section Type: Query section

Tip Name: Creating Incremental Drilling

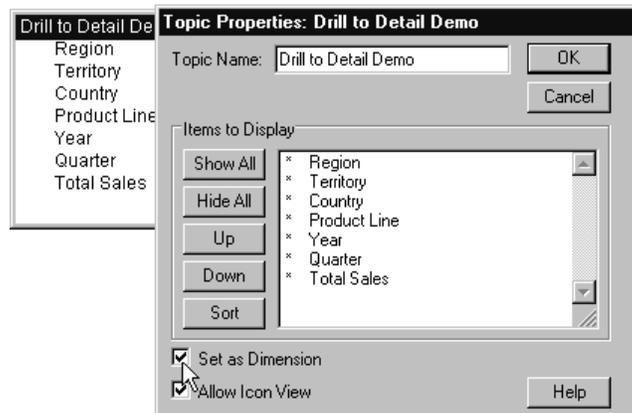


Purpose / Use: To allow users to drill to further detail without needing to fetch all information into the original document beforehand. Using incremental drilling provides a fast and efficient way to provide drilling while avoiding the creation monster BQY files.

Sample File: Incremental drilling.bqy

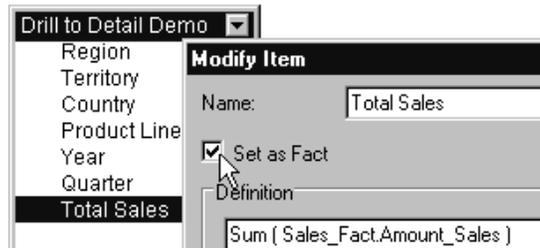
Steps to Produce: To create incremental drilling, you will use a combination of a pre-defined drill path plus the drill-to-detail feature. This combination will tell Brio to automatically drill into detail information that is not currently contained in the query.

- 1) Build a data model with a single Meta Topic. Once created, set the data model view to show only the Meta Topic (use DataModel | DataModel View | Meta in the menu system).
- 2) Double-click on the title bar of the meta topic to display the Topic Properties dialog. Use the "Up" and "Down" buttons to put the topic items in the order in which you want your user to drill. Then, turn on the "Set as Dimension" checkbox. This defines the order of items in the meta topic as the pre-defined drill path.



(Creating Incremental Drilling continued)

- 3) To remove a topic item from the drill path (such as items that are facts, not dimensions), click on the topic item. Right-click on the item and select “Properties” from the speed menu. Turn on the “Set as Fact” checkbox.



- 4) In the menu system, select DataModel | DataModel Options to display the Data Model Options dialog. On the General tab, turn on the “Drill to Detail” checkbox.



- 5) Create your top-level chart as usual. When a user double-clicks on a bar in the chart, Brio will ask if the user wants to add the missing dimension to the query, and then if the user will allow it to copy the query and chart definition to new sections to preserve the original. Answer “yes” to both questions. (You can avoid the dialog box questions in the future by checking the “Don’t show this warning again” checkbox in the dialogs the first time.)
- 6) Brio will copy the query and chart sections, but with a new dimension on the query’s Request line and a new limit based on the bar the user double-clicked on. Now, keep double-clicking on a target bar and Brio will continue to add new dimensions and limits, and then automatically re-process the query as needed to perform the drilling.

Section Type: Results or Table section

Tip Name: Creating Fiscal Year based information

	Store Category	State	City	Store Name	Date Ordered	Fiscal	Fiscal Year	Fiscal Quarter	Date Ordered Month	Fiscal Month Number	Units	Revenue
1	Computer	AZ	Tuscon	Computer Selection	04/14/1995	1/14/1996	1996	Q1	Apr	1	35	\$4
2	Computer	AZ	Tuscon	Computer Selection	06/02/1995	3/2/1996	1996	Q1	Jun	3	30	\$5
3	Computer	AZ	Tuscon	Computer Selection	06/19/1995	3/19/1996	1996	Q1	Jun	3	35	\$4
4	Computer	AZ	Tuscon	Computer Selection	07/02/1995	4/2/1996	1996	Q2	Jul	4	10	\$
5	Computer	AZ	Tuscon	Computer Selection	07/10/1995	4/10/1996	1996	Q2	Jul	4	25	\$4
6	Computer	AZ	Tuscon	Computer Selection	07/13/1995	4/13/1996	1996	Q2	Jul	4	10	\$1

Purpose / Use: Often, the only date-oriented information in your database is calendar year based. You can derive Fiscal Year based information with a combination of Date Groupings and Computed Items.

Sample File: Fiscal Year info.bqy

Steps to Produce: In this example, we will assume that your Fiscal Year starts on April 1.

- 1) In the Results section, select the date field to be transformed into Fiscal Year information. Right-click and select Add Date Groups from the speed menu. This action will add three columns. Select the Year and Quarter columns and delete them. This will leave you with a derived column that displays the Month name for the date.
- 2) Add a new Computed Item and call it "Fiscal". The formula you use will convert the original date into a date adjusted for the start of the fiscal year. You will want to add the number of months it would take to adjust the starting date of your fiscal year to be January 1 of the following calendar year. April 1 needs 9 months to be adjusted to January 1, so in our example we define the formula to be:

AddMonths (Date_Ordered, 9)

For fiscal years that have different starts, the following table shows how many months to add.

Fiscal Year Start	Months to Add
April 1	9
July 1	6
October 1	3

- 3) Select the Fiscal column created in the previous step. Right-click and select Add Date Groups from the speed menu. Select the Fiscal_Month column and delete it (since it really doesn't contain useful information). This action leaves you with Fiscal_Year, Fiscal_Quarter and the actual Date_Ordered_Month.

(Creating Fiscal Year based information continued)

- 4) This sequence of steps will produce a Date_Month column that displays the month names in short (3 letter) form. To display the full month name, right-click on the Date_Ordered_Month and select Number from the speed menu. In the number format dialog box that appears, select the Custom category and type in the following format formula:

```
"MMMM"
```

This display format (4 m's) will show the full month name without displaying the year or day.

- 5) To display monthly items in proper fiscal year order, you will also need to create a Fiscal_Month_Number column. To add a Fiscal_Month_Number column, create a new Computed Item with that name. Define the formula to be:

```
Decode (ToChar ( Date_Ordered, "mmm" ), 'Jan',10, 'Feb',11,  
'Mar',12, 'Apr',1, 'May',2, 'Jun',3, 'Jul',4, 'Aug',5,  
'Sep',6, 'Oct',7, 'Nov',8, 'Dec',9,0)
```

For fiscal years that start on a date other than April 1, adjust the numbers in the decode function accordingly.

Section Type: Results or Table section

Tip Name: Creating Range-based analysis

Fiscal Year	1996	1997
Order Size	Order Count	Order Count
\$ 0 to \$1000	130	114
\$1001 to \$2000	109	89
\$2001 to \$3000	85	83
\$3001 to \$4000	76	90
\$4001 to \$5000	46	53
Greater than \$5000	47	83

Purpose / Use: You may want to convert a measure with many discrete values into a derived measure that groups these items into a set of value ranges.

For fields with many discrete values, creating a Brio Grouping Column is often not appropriate. Grouping columns are unwieldy to deal with when a large list of values is present, and are defined in such a way that they can only handle a single contingency group for values not present at definition time. A better solution exists using a Computed Item with complex If-Then-Else logic defined.

Sample File: Results ranges.bqy

Steps to Produce: In this example, we want to define a column that displays in which member in a range of values the revenue size of each order falls. We then want to see how many orders fall into each size range.

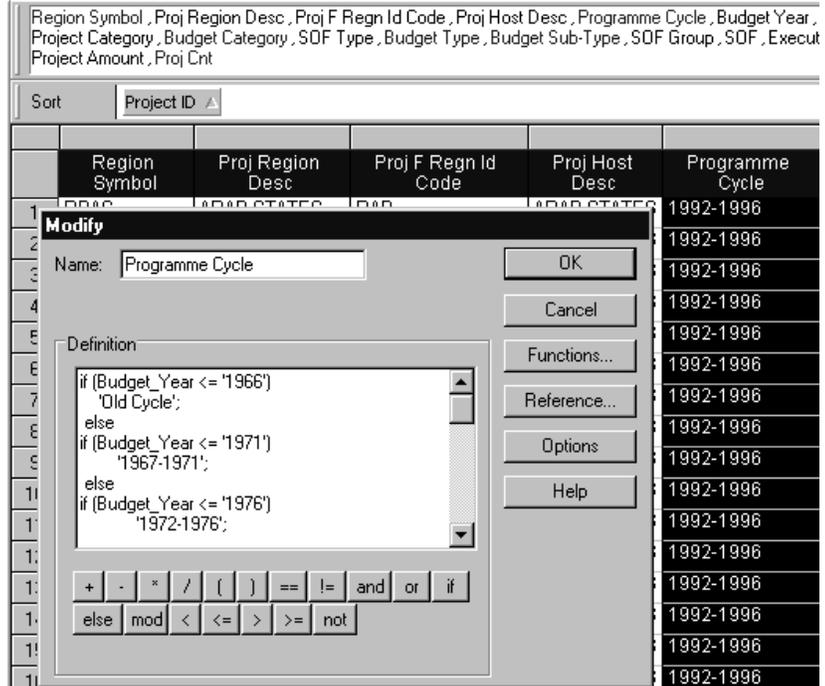
- 1) In the Results section, add a new Computed Item. Call it Order_Size, and define its formula to be:

```
if (Revenue <= 1000) {'$      0 to $1000'} else {
if (Revenue <= 2000) {'$1001 to $2000'} else {
if (Revenue <= 3000) {'$2001 to $3000'} else {
if (Revenue <= 4000) {'$3001 to $4000'} else {
if (Revenue <= 5000) {'$4001 to $5000'} else {
'Greater than $5000' } } } } }
```

- 2) Add another Computed Item. Call it Order_Count and define its formula to be 1.
- 3) Use the Order_Size field as a label and the Order_Count field as a measure in your Pivots, Charts and Reports.

Section Type: Results or Table section

Tip Name: Creating multi-contingency grouping / range columns



Purpose / Use: Sometimes, you don't necessarily have all the possible values to be grouped at design time. And in many cases you need more contingency options than the single contingency that the regular Grouping Column feature provides.

Sample File: *Not available to the public*

Steps to Produce: In this example, we want to bracket our range values. We want to plan for items that either fall off the bottom of the list (i.e., smaller than we want to analyze) or else fall off the top of the list (i.e., larger than we want to deal with separately).

We will be looking to group budget expenditures into multi-year Program Cycles. We aren't interested in separating out the Program Cycles for any items earlier than 1967, or later than 2009.

We also want to have ranges that are different sizes to handle historical changes in the way these values were analyzed. Prior to 1997, program cycles were 5 years long. From 1997 onward, the organization converted to 3-year program cycles.

(Multi-contingency Grouping Columns continued)

- 1) In the Results section, add a new Computed Item. Call it Program Cycle, and define its formula as follows:

```
if (Budget_Year <= '1966') { 'Old Cycle'
} else {
if (Budget_Year <= '1971') { '1967-1971'
} else {
if (Budget_Year <= '1976') { '1972-1976'
} else {
if (Budget_Year <= '1981') { '1977-1981'
} else {
if (Budget_Year <= '1986') { '1982-1986'
} else {
if (Budget_Year <= '1991') { '1987-1991'
} else {
if (Budget_Year <= '1996') { '1992-1996'
} else {
if (Budget_Year <= '2000') { '1997-2000'
} else {
if (Budget_Year <= '2003') { '2001-2003'
} else {
if (Budget_Year <= '2006') { '2004-2006'
} else {
if (Budget_Year <= '2009') { '2007-2009'
} else {
'Future Cycle' } } } } } } } } } }
```

Note that the first and last entries in this formula create the ceiling and floor contingencies. You can create more complex contingency scenarios by using multiple measures in the nested If-Then-Else logic.

- 2) Use this new column as a dimension label for your Pivots, Charts and Reports.

Section Type: Results or Table section

Tip Name: Analyzing “factless” data

	Americas	Asia Pacific	Europe
Store Type	Store Cnt	Store Cnt	Store Cnt
Franchisee	5	3	4
Online	1	2	2
Superstore	4	2	4

Purpose / Use: Some data sets do not have any obvious fact columns. For example, qualitative data sets are collections of ratings or item types. Brio Intelligence can still be used to analyze these data sets with a simple concept applied to the Results section.

Sample File: Factless data.bqy

Steps to Produce: To analyze “factless” data, you merely need to create a computed fact based on the dimensions present.

- 1) In the Results section, create a Computed Item named “Store Cnt”. Set the formula for this computed item to be:

1

Yes, create a computation with the simple formula of the number one.

- 2) Use this new “fact” in any chart, pivot or report to analyze the number of occurrences of each of the items in your query.

Section Type: Results or Table section

Tip Name: Count of Participants #1
 Displaying the number of States, not the number of rows in each State, in a Report

1997		
State	City	Units
CA	Burbank	690
	Glendale	160
	Oakland	50
	San Diego	1,525
	Westwood	115
FL	Palmetto	170
MN	Eagan	150
NY	New York	630
	Rochester	400
OR	Portland	270
Sales Participation for 1997 = 5 States		4,160

Purpose / Use: Dragging the State field into a Report design will display a list of the unique state names. However, changing the data function to “count” will display the number of rows within the group, not the *number* of unique states in the group.

Sample File: State count solution.bqy

Steps to Produce: In the Results section, sort in the order in which the report groups are defined plus the item to be counted uniquely. In this example, the report is grouped by Year, and the item to count uniquely is the State. Therefore, create a sort line with first Year, then State.

1) Add a Computed Item named “State Cnt” with the following formula:

```
if ( Prior ( State ) != State ) { 1 } else { 0 }
```

2) Now, drag the “State Cnt” field into the report design. The proper number of unique states within the report group will display.

If you want to further format the display, make sure the expression line is displayed and select the “State Cnt” item in your report. You can then edit the formula expression to add text and other data elements. In this example, the formula was changed to read as follows:

```
'Sales Participation for '+currBreak.Value("Year")+ ' = '+Tables("Results").Columns("State_Cnt").Sum(currBreak)+' States'
```

Section Type: Results or Table section

Tip Name: Count of Participants #2
Counting the number of stores in each state while also calculating the sum of all revenues.

	Store Cnt	Revenue
AZ	2	\$112,210.05
CA	10	\$1,217,486.45
FL	1	\$70,835.35
MN	2	\$159,106.90
NY	7	\$670,802.55
OR	1	\$115,263.90

Purpose / Use: Since you are also analyzing a total for each state, you can't simply create a query with Count(Distinct Store_Name).

Sample File: State count solution.bqy

Steps to Produce: The secret here is a creative combination of sorting the Results set and using the Next or Prior function in a computed item.

- 1) Select the Results section, and sort by State and Store_Name.
- 2) Create a computed item called Store_Cnt with the following definition:

```
if ( Prior ( Store_Name) != Store_Name) { 1 }
else { 0 }
```

The sorting insures that a new Store Name also means a new store to count.

- 3) Now, build your pivot, using the category field as labels and the dollar amount field and the new Store_Cnt computed item as the values.

Section Type: Results or Table section
 Tip Name: Qualified Sub-Totals #1
 Display the sub-total of all transactions over \$60K.

Anaheim	
Total Sales:	\$1,359,465
Transactions over 60k:	\$677,664
Fiscal Week	Amount Sales
1	6,752
2	12,831
3	30,902
4	10,211
8	11,020

Purpose / Use: You want to add up the transaction total within each group, but only for individual transactions larger than a certain threshold.

Sample File: Sales over 60k subtotal.bqy

Steps to Produce: This solution uses a computed item to separate out the qualifying transactions.

- 1) Build your query as normal and process it to fetch some initial data.
- 2) With the Results section selected, add a new Computed Item named "Over 60k" using the following formula:


```
if ( Amount_Sales >= 60000 ) { Amount_Sales } else { 0 }
```
- 3) In the event you want a limit that can be easily selected by end users, you can also create a computed item named "Threshold" and simply enter the threshold value desired. The "Over 60k" formula would then become:


```
if ( Amount_Sales >= Threshold ) { Amount_Sales } else { 0 }
```
- 4) Add this "Over 60k" computed item column to the appropriate group header in your report.
- 5) If using the Threshold column method, add the threshold value to your report by adding the Threshold field to the report group, right-clicking on it and changing the Data Function to either Max or Min.

Section Type: Table section

Tip Name: Qualified Sub-Totals #2
 Display all contributors that account for the top 40% of revenue within each group.

Store Type	Product Family	Amount Sales	Top 40% within Store Type
Franchisee	Computers and Internet	\$2,456,643.04	10.07%
Franchisee	Biographies and Memoirs	\$1,876,064.40	17.76%
Franchisee	Business and Investing	\$1,731,220.39	24.85%
Franchisee	Home and Garden	\$1,475,772.28	30.90%
Franchisee	History	\$1,448,965.78	36.84%
Online	Computers and Internet	\$12,734,704.95	17.73%
Online	Biographies and Memoirs	\$6,271,822.90	26.46%
Online	Business and Investing	\$5,752,155.51	34.46%
Superstore	Computers and Internet	\$2,766,286.46	7.55%
Superstore	Biographies and Memoirs	\$2,520,792.99	14.43%
Superstore	History	\$2,282,493.25	20.66%
Superstore	Action and Adventure	\$2,122,697.09	26.45%
Superstore	Arts and Music	\$2,118,898.86	32.24%
Superstore	Audiobooks	\$2,030,234.59	37.78%

Purpose / Use: You don't want a fixed number of contributors (which would be a TopN analysis). Instead, you want all the contributors (however many there are) that account for the top 40% of your total revenue for each group.

Sample File: Prod fam within store type up to 40%.bqy

Steps to Produce: This solution requires two Table sections. One table section to create the needed summary calculation, then another to convert the summary field to a regular number upon which a limit can be applied.

- 1) Build your query as normal and process it to fetch some initial data.
- 2) With the Results section selected, add a new Computed Item named "Pcnt of Store Type" using the following formula:

$$\text{Amount_Sales} / \text{Sum}(\text{Amount_Sales}, \text{Store_Type})$$

- 3) Now, insert a new Table section and drag all fields into the Table section design.
- 4) With the Table section selected, sort the Table on the category field used for the grouping and then on the percent calculation field. In this example, you would sort on "Store Type" then on "Pcnt of Store Type".
- 5) Add a new Computed Item to the Table section named "Top 40% within Store Type" with the following formula:

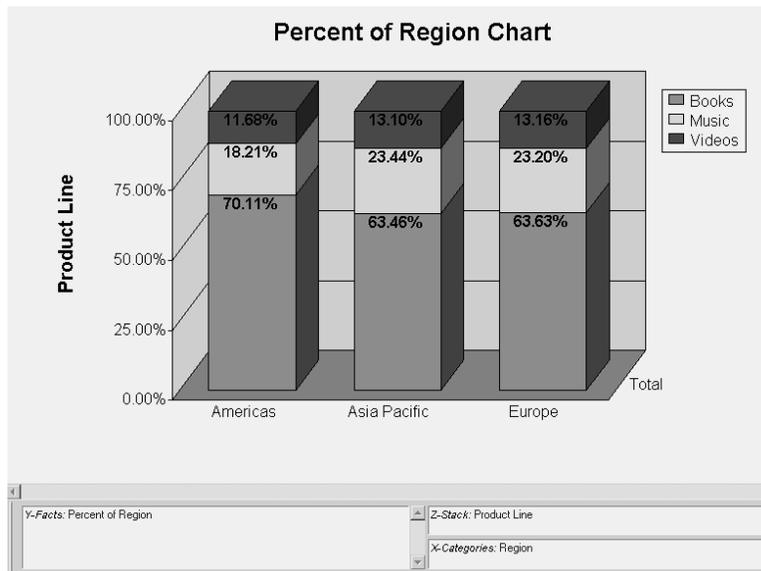
$$\text{Cume}(\text{Pcnt_of_Store_Type}, \text{Store_Type})$$

This calculation, in tandem with the sorting, provides a running sum of the total contribution so far of each store within the group.

- 6) With this Table section selected, insert another new Table section and drag all fields except "Pcnt of Store Type" from the first Table section into this new Table section.
- 7) Double-click on the "Top 40% within Store Type" column to create a limit. Set the limit to be less than or equal to 0.4 (the decimal equivalent of 40%).

Section Type: Results or Table section

Tip Name: Percent of Region



Purpose / Use: You want to display multiple values as percentages with a group, with each group adding up to 100%.

Sample File: 100% of region.bqy

Steps to Produce: This solution uses two computed items, one to calculate the group total, and a second that uses this group total to calculate the individual's percent contribution to the group total (rather than the overall total).

- 1) Build your query as normal and process it to fetch some initial data.
- 2) With the Results section selected, add a new Computed Item named "Region Total" using the following formula:

`Sum (Amount_Sold, Region)`

The second parameter in the Sum() function tells Brio Intelligence to only sum rows with a value in the Region column equal to the current row's value. In other words, it calculates the total for the current group.

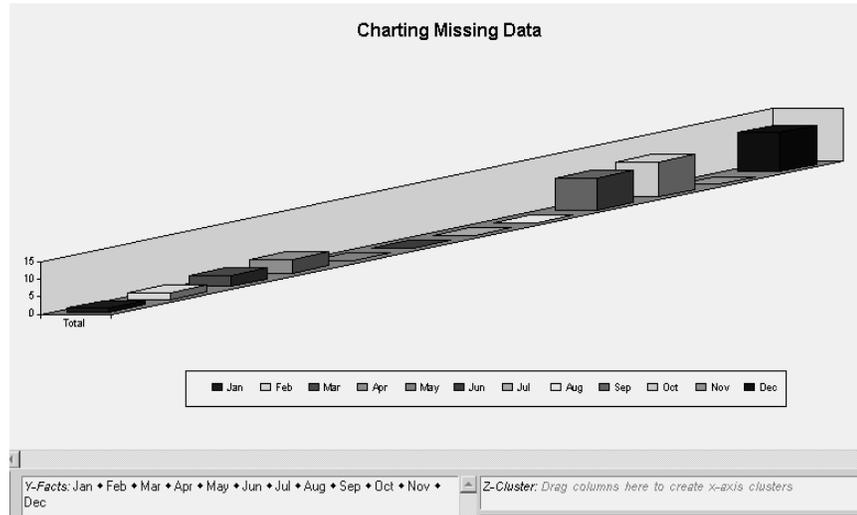
- 3) Add a second computed item named "Percent of Region" using this formula:

`Amount_Sold / Region_Total`

- 4) Use the "Percent of Region" column in your chart, pivot and report designs, and then format it as a percentage.

Section Type: Results or Table section

Tip Name: Charting Missing Data



Purpose / Use: You want to display a complete time series chart, even if data elements are missing for certain time slots.

Sample File: Charting missing data.bqy

Steps to Produce: This solution uses a series of computed items to insert a zero when no data exists for a time slot. By this technique, you take a single fact column and “dimensionalize” it into numerous computed fact columns.

- 1) Build your query as normal and process it to fetch some initial data.
- 2) With the Results section selected, add a new Computed Item for each time series item to test the presence of a value for that time slot. In the example, the time series is monthly, so we create a computed item for each of the 12 months. The formula for testing the January values based on the “Date” column would be:

```
if (ToChar ( Date, "mm" ) == "01" ) { Amount } else { 0 }
```

The Results section should look like this when all computed items have been defined:

	Date	Date Month	Amount	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1	01/02/2002	Jan	1	1	0	0	0	0	0	0	0	0	0	0	0
2	02/02/2002	Feb	2	0	2	0	0	0	0	0	0	0	0	0	0
3	03/02/2002	Mar	3	0	0	3	0	0	0	0	0	0	0	0	0
4	04/02/2002	Apr	4	0	0	0	4	0	0	0	0	0	0	0	0
5	09/02/2002	Sep	9	0	0	0	0	0	0	0	0	9	0	0	0
6	10/02/2002	Oct	10	0	0	0	0	0	0	0	0	0	10	0	0
7	12/02/2002	Dec	11	0	0	0	0	0	0	0	0	0	0	0	11

- 3) Once all time series computed items have been added, build your chart, pivot or report using all of the time series computed items.

Section Type: Table section

Tip Name: Hiding columns from a Pivot or Chart

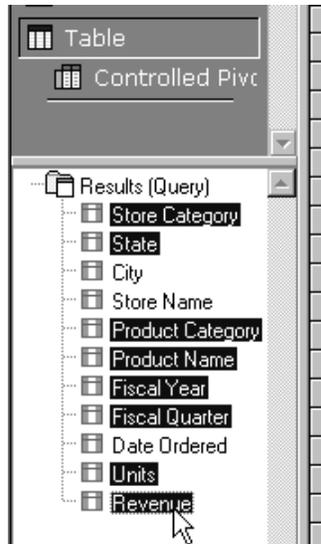
	1996			
	1st Half		2nd Half	
Store Category	Units	Revenue	Units	Rev
Computer				11
Discount				2
Electronics				9

Purpose / Use: In some applications, you need to fetch more columns than you want users to be able to drill into with Drill Anywhere. Hiding a column in the Results section doesn't hide it from the drill path of a pivot or chart.

Sample File: Hiding columns in a pivot.bqy

Steps to Produce: Create a Table section with the items you want users to drill into, then build your pivot or chart from the Table instead of the Results section. *(Works with all v6.x versions of Brio Intelligence.)*

- 1) Build your query as normal and process it to fetch some initial data.
- 2) With the Results section selected, insert a new Table section.
- 3) Drag into the Table design only those fields you want to use as either part of your initial charts and pivots, or as items available for drilling.



- 4) With the Table section selected, insert a new chart or pivot. This sequence ensures that your chart or pivot will be built based on the Table section, not the base Results section.

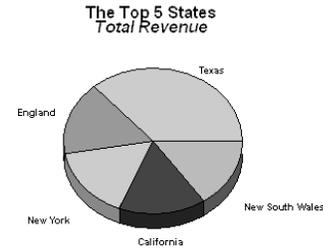
Section Type: Table section
 Tip Name: Top and Bottom N Analysis



**Automated
TopN Analysis**

Display the Top:

State	Revenue
Texas	\$32,713,875.41
England	\$15,053,892.94
New York	\$14,512,139.80
California	\$14,028,980.44
New South Wales	\$13,906,757.16



Purpose / Use: You want to create a TopN or BottomN analysis, with the user able to select what value of N to use on the fly.

Sample File: Top and Bottom N.bqy

Steps to Produce: Allowing users to select the value of N means creating a dynamically controlled limit. Unfortunately, you can't create a limit on an aggregated field such as a Rank or Sum. Using a Table section as the data staging area allows you to place a limit on the controlling aggregate value.

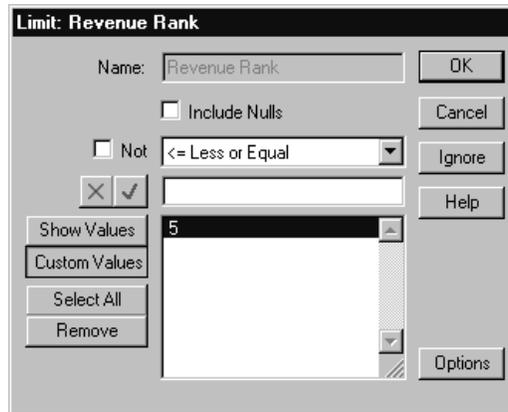
- 1) In the Results section, create a Computed Item to determine the ranking. In this example, we are looking at the revenue ranking by State.
- 2) For TopN analysis, create a Computed Item named "Revenue Rank" with a formula as follows:

$$\text{Rank (Revenue, State)}$$
- 3) For BottomN analysis, create a Computed Item named "Reverse Rank" with a formula as follows:

$$\text{RankAsc (Revenue, State)}$$
- 4) With the Results section selected, insert a new Table section.
- 5) Drag all regular fields into the Table design. Then add the appropriate ranking column to the Table (Revenue Rank for a TopN analysis, or Reverse Rank for a BottomN analysis).
- 6) In the example of a TopN analysis, double-click on the new Table section in the catalog and rename it to be called "TopN".

(Top and Bottom N Analysis continued)

- 7) In the TopN Table section, double-click on the ranking field to add a limit and set its initial value to be less than or equal to 5.



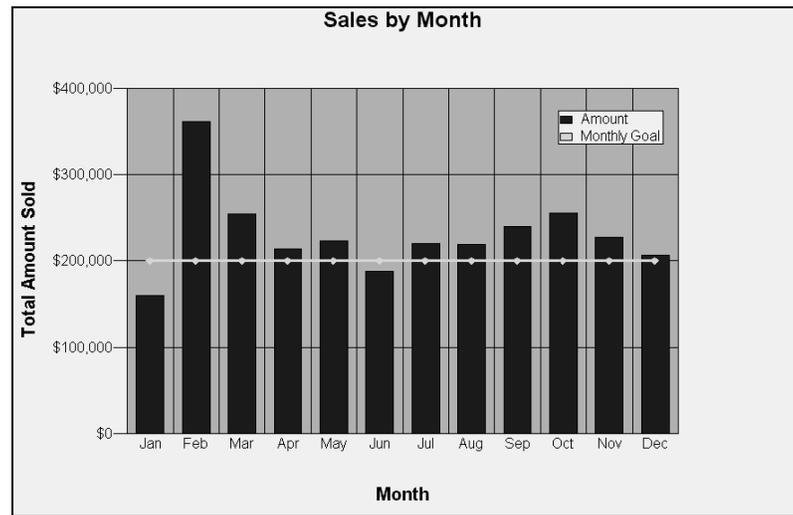
- 8) With the Table section selected, insert a new chart or pivot. This sequence ensures that your chart or pivot will be built based on the Table section, not the base Results section. Build your chart or pivot as usual.
- 9) Create an EIS section and embed the appropriate charts and pivots that were built based on the Table section. At a minimum, add a text box control with a script for the OnChange event similar to the following code:

```
newN = ActiveSection.Shapes["tbN"].Text

with (ActiveDocument.Sections["TopN"].Limits["1"]) {
    SelectedValues.RemoveAll()
    Ignore = false
    SelectedValues.Add(newN)
}
ActiveDocument.Sections["TopN Chart"].Title = "The Top " +
    newN + " States"
```

Section Type: Chart section

Tip Name: Baseline Comparison Chart (using a fixed target value)



Purpose / Use: You often want to compare actual values with a constant projected value or target. (The next tips show how to use a target that changes based on values in the database.)

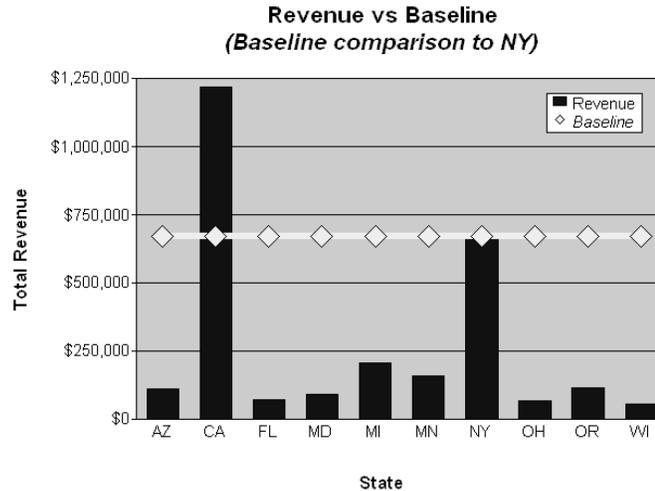
Sample File: Baseline and spotlighting.bqy

Steps to Produce:

- 1) Create your chart as usual.
- 2) Set the chart type to be Bar-Line (the last one in the toolbar chart type pulldown list).
- 3) Right-Click the chart and select Add Computed Item. Name the item "Target", "Baseline" or some other appropriate name. Enter the target value (say, 200000) as the definition of the Computed Item.
- 4) In the chart itself, select one of the values for the Target. Right-click, and select "Properties". On the "Patterns" tab, change the line width to 3 points, and change the marker size to 3 points.
- 5) Modify the chart settings.
 - On the General tab, remove the 3-D Objects option. (Most Bar-Line charts look better in 2-D.)
 - On the Values Axis tab, turn off the Show Values at Right option.
 - On the Bar Chart tab, the Shift setting should be Shift Points to Center.

Section Type: Chart section

Tip Name: Baseline Comparison Chart (using values from the database)



Purpose / Use: You often want to compare actual values with a target value that resides in the database. This target value may change based on a number of factors, so your comparison chart must key the baseline off data fetched from the database.

Sample File: Data-based baseline comparisons.bqy

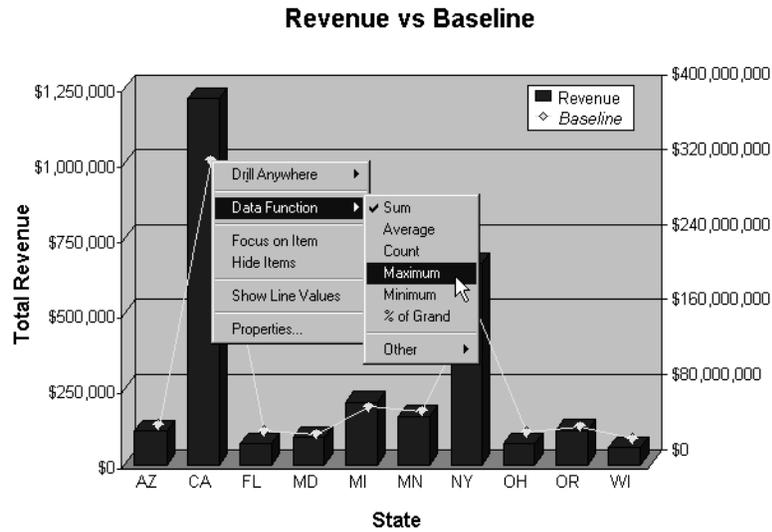
Steps to Produce: Brio Intelligence v6.x added support for creating summary computed items that key off a single group total. This new summarization ability can be used to create baseline comparison charts that work off a reference value contained in the database itself.

- 1) Before creating the chart, select the Results section. Create a new Computed Item. Name the item "Target", "Baseline" or some other appropriate name. In this example, we want to use the revenue figures for the state of New York as the baseline of comparison. Therefore, enter the target value's formula as the following:

Sum (Revenue, State, 'NY')

- 2) Create your chart as usual. Add the real value as the first Y-value and your new computed Target item as the second Y-value.
- 3) Set the Chart Type to be Bar-Line (the last one in the toolbar pulldown list).
- 4) In the chart itself, select one of the values for the Target. Right-click, and select Data Function from the speed menu. Change the data function from Sum to Maximum, Minimum or Average (see the figure below).

(Baseline Comparison Chart using values from the database continued)

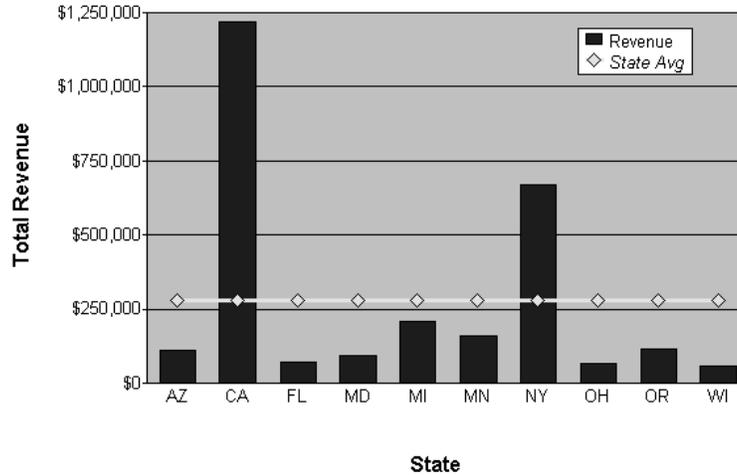


- 5) In the chart itself, select one of the values for the Target. Right-click, and select "Properties". On the "Patterns" tab, change the line width to 3 points, and change the marker size to 3 points.
- 6) Finally, modify the chart settings.
 - On the General tab, remove the 3-D Objects option. (Most Bar-Line charts look better in 2-D.)
 - On the Values Axis tab, turn off the Show Values at Right option. Also turn off the Auto checkbox in the Right Axis Scale section. Change the Min and Max settings for the right axis scale to match the ones for the left axis.
 - On the Bar Chart tab, the Shift setting should be Shift Points to Center.

Section Type: Chart section

Tip Name: Baseline Comparison Chart (using averages from the database)

Revenue vs State Avg



Purpose / Use: You often want to compare actual values with an average of values that reside in the database. Since this is a comparison against an average, your comparison chart must key the baseline off data fetched from the database.

Sample File: Data-based baseline comparisons.bqy

Steps to Produce: Brio Intelligence v6.x added support for creating computed items that key off the number of distinct groups rather than the number of rows. This new summarization ability can be used to create correct baseline averages for use in comparison charts.

- 1) Before creating the chart, select the Results section. Create a new Computed Item. Name the item "State Total" or some other appropriate name. Enter the computed item's formula as the following:

Sum (Revenue)

- 2) Next, we need to calculate the number of state groups (rather than the total number of rows). Create a second Computed Item called "State Cnt" with the formula:

CountDistinct (State)

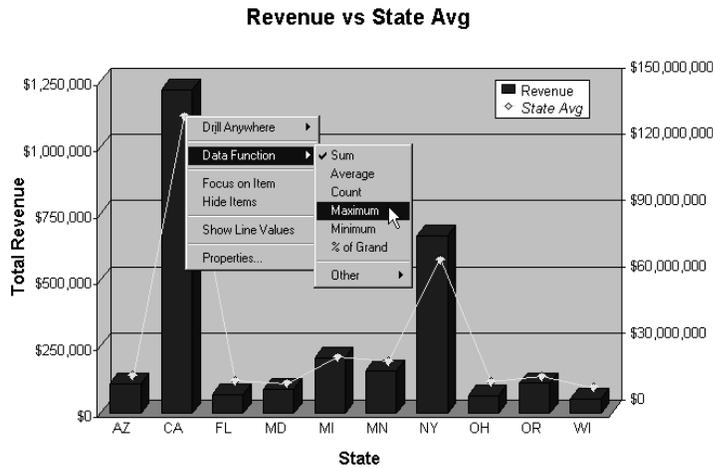
- 3) Now, create a third Computed Item that will hold the average revenue for an average state. Name it "State Avg" and define its formula as:

State_Total / State_Cnt

- 4) Create your chart as usual. Add the real value as the first Y-value and your new computed "State Avg" item as the second Y-value.

(Baseline Comparison Chart using averages from the database continued)

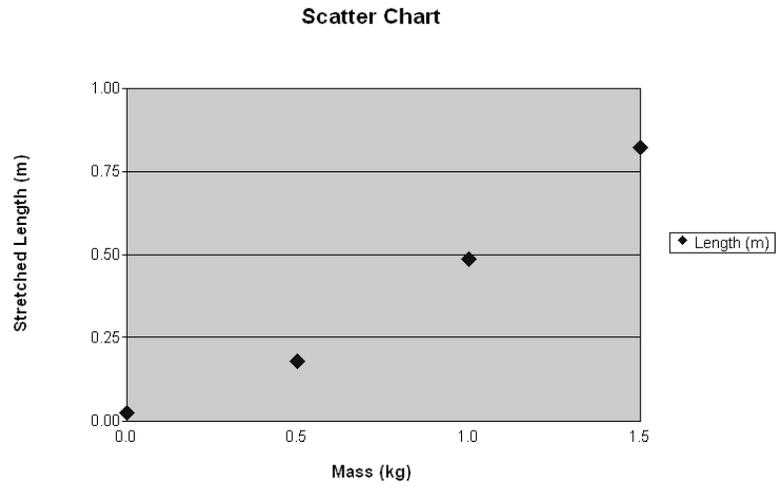
- 5) Set the Chart Type to be Bar-Line (the last one in the toolbar pulldown list).
- 6) In the chart itself, select one of the values for the Target. Right-click, and select Data Function from the speed menu. Change the data function from Sum to Maximum, Minimum or Average (see the figure below).



- 7) In the chart itself, select one of the values for the Target. Right-click, and select "Properties". On the "Patterns" tab, change the line width to 3 points, and change the marker size to 3 points.
- 8) Finally, modify the chart settings.
 - On the General tab, remove the 3-D Objects option. (Most Bar-Line charts look better in 2-D.)
 - On the Values Axis tab, turn off the Show Values at Right option. Also turn off the Auto checkbox in the Right Axis Scale section. Change the Min and Max settings for the right axis scale to match the ones for the left axis.
 - On the Bar Chart tab, the Shift setting should be Shift Points to Center.

Section Type: Chart section

Tip Name: Scatter Charts



Purpose / Use: Some data lends itself better to a chart type known as a scatter plot. Scatter charts are not a type available as a choice in the chart toolbar, but they can be easily created.

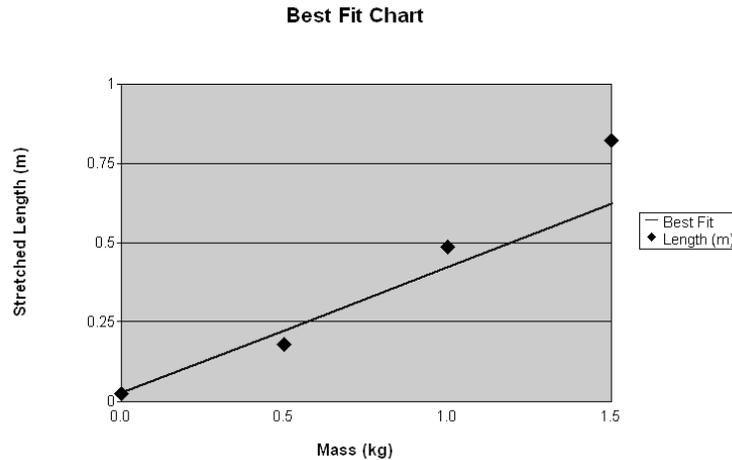
Sample File: Scatter plots.bqy

Steps to Produce: Create a scatter chart by building a line chart and then turning off the line.

- 1) Create your chart as usual, and select "Line" as the chart type.
- 2) Click on one of the data points, then right-click and select "Properties...".
- 3) On the Patterns tab, set the line pattern to "None".

Section Type: Chart section

Tip Name: Best-Fit Charts



Purpose / Use: One typical way to analyze data is to compare the actual data with an ideal set of values to see how the actual results compare with expected results. Best-fit charts are a way to display this. Best-fit charts are not a type available as a choice in the chart toolbar, but they can be easily created.

Sample File: Scatter plots.bqy

Steps to Produce: Create a scatter chart of the actual data, then add a computed item to plot the best fit values.

- 1) Create a line chart, using your actual data points.

Do not change any line properties till after you have added all the fact columns. Otherwise, the chart engine will reset all facts to the default settings each time a new fact is added.

- 2) You will use a linear regression formula to calculate the best-fit plot. A best-fit line will use the formula $y = mx + b$, where m =slope and b =y-intercept. To create a computed item named "Best Fit", you will need to compute the average slope of the data points and a y-intercept value.

The slope of a line is defined as $(y_2 - y_1) / (x_2 - x_1)$. In this case, the Length_m column holds the y-values, and the Mass_kg column holds the x-values. Since the sample values of y start where $x=0$, we can use the minimum column value as the y-intercept.

- 3) In the Results section, create a computed item named "Slope" using a formula similar to the following:

$$(\text{Length_m} - \text{Prior}(\text{Length_m})) / (\text{Mass_kg} - \text{Prior}(\text{Mass_kg}))$$

(Best Fit Charts continued)

- 4) In the Results section, create a computed item named "Avg Slope" using a formula similar to the following:

$$\text{Avg (Slope)}$$

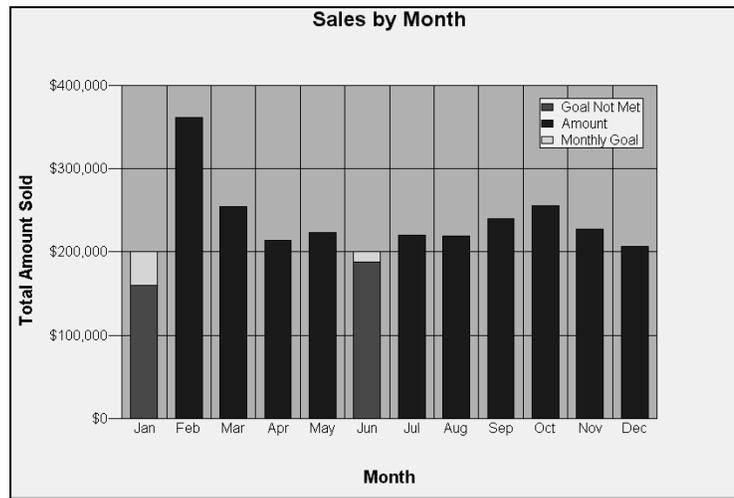
- 5) In the Results section, create a computed item named "Best Fit" using the following formula:

$$(\text{Mass_kg_} * \text{Avg_Slope}) + \text{ColMin}(\text{Length_m_})$$

- 6) Add the "Best Fit" computed item to your chart.
- 7) Right-click on one of the "Best Fit" data points and select "Properties...".
- 8) On the Patterns tab, set the Marker pattern to "None". Make sure the line color is one that will be easily visible.
- 9) Right-click on one of the actual data points and select "Properties...".
- 10) On the Patterns tab, set the line pattern to "None". Make sure the marker color is one that will be easily visible.

Section Type: Chart section

Tip Name: Spotighting in a Chart



Purpose / Use: The Spotighter is a very useful tool for visually tagging deviant values. However, the Spotighter is not available in the Chart section.

Sample File: Baseline and spotlighting.bqy

Steps to Produce: Here is how to create spotlighting by using layered computed items.

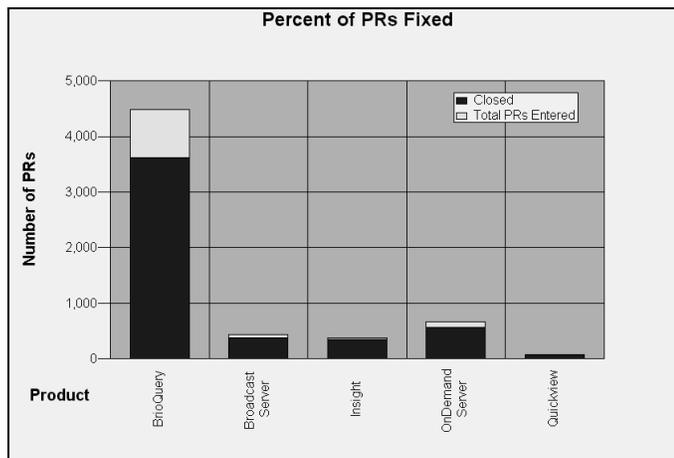
- 1) Create your chart as usual, with the actual value as the first Y value.
- 2) Right-click on the chart, and create a Computed Item. Name it something appropriate like "Monthly Goal". Set its definition to be your target value (say, 200000).
- 3) Create another Computed Item. Name it "Goal Not Met", and set its definition to be:


```
if (Monthly_Goal > Amount) {Amount} else {0}
```

 This formula creates a bar only when you have a value to spotlight.
- 4) Change the chart type to Vertical Stack Bar (3rd icon from the top in the toolbar's chart type pulldown).
- 5) Drag the Goal Not Met field to be the first item in the Y values.
- 6) Set the legend to be based on the Y values. Use the Color Pallette to reset the colors for each value series. One good setting is to use red for Goal Not Met (in front), blue for the actual value (in the middle), and yellow for Goal (in the back).
- 7) Modify the chart and turn off the 3-D Objects option.

Section Type: Chart section

Tip Name: Portion within a Category



Purpose / Use: Pie charts are good for displaying “portion of the whole” comparisons. Sometimes, you may want to display the portion of just a single item within each of several categories at the same time.

Sample File: Percent of category.bqy

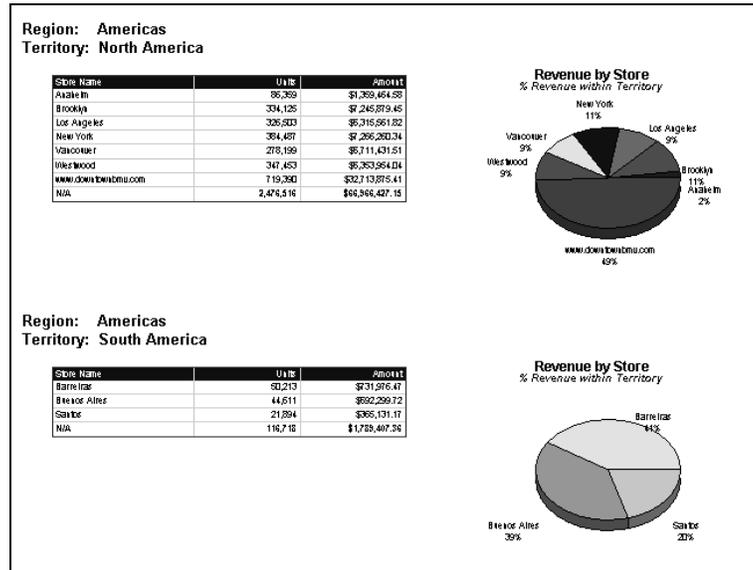
Steps to Produce: At Brio, we track Problem Reports and want to know how many are closed versus the total for each category. The secret here lies in using a Grouping Column in a unique way.

- 1) In the Results section, create a Grouping Column on the status item you want to use (i.e., the one that determines whether a PR is closed or not in our example). Place any status that is considered closed in a group called Closed, and place all other status descriptions in a group called Total PRs Entered.

In other words, instead of calling these other items Not Closed, we are intentionally “mislabeling” them in the interests of our end result, the portion chart. Therefore, it is important that the group label we “misuse” is set so that it falls later in the alphabet. That way, it gets charted last.

- 2) Insert a new chart and set its type to Vertical Stack Bar (3rd icon from the top in the toolbar’s list of chart types).
- 3) Drag the amount / count field into the Y values. Place the new Grouping Column in the Z labels area. Place an appropriate category label in as the X labels.
- 4) Set the legend to be based on the Z labels.
- 5) Modify the chart and turn off the 3-D Objects option.

Section Type: Chart section
 Tip Name: Chart Design for Report Embedding

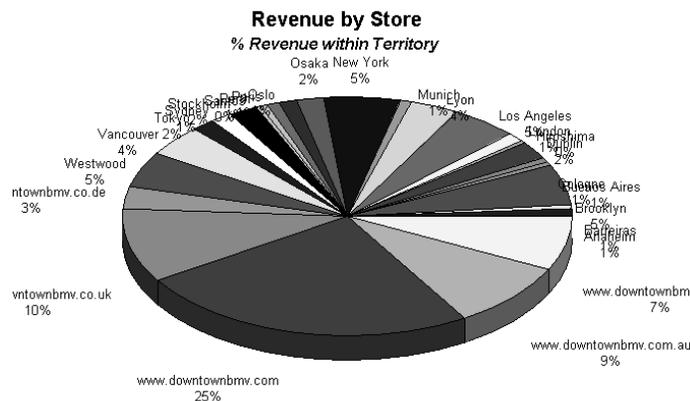


Purpose / Use: You want to design a chart differently if it will be used in a Report section.

Sample File: Chart for report embedding.bqy

Steps to Produce: Keep in mind what will happen to the chart once it is embedded in the report. The SmartReport logic will narrow down the contents of the chart to only display those items that apply to the current report group.

- 1) Build your chart with all the data elements your report will need. The chart will look very busy in its raw form. Don't worry. Embedding will clean it up.



- 2) Create a report with appropriate groupings defined. Drag your chart design into one of the group areas or into the body area. The chart will become context-sensitive automatically.

Section Type: Pivot section

Tip Name: Adding textual annotations to a Pivot

	1996			
	Units	Revenue	ASP	Reason
Q1	2,475	\$300,433.25	\$121.39	
Q2	2,805	\$299,895.05	\$106.91	
Q3	2,930	\$289,666.45	\$98.86	Too low for profit
Q4	3,300	\$334,764.95	\$101.44	

Purpose / Use: Often, you want your users to know *why* a particular item was highlighted in a Pivot.

Sample File: Pivot annotations.bqy

Steps to Produce: You add a simple Computed Item to the Pivot design to display a text string when a particular condition is met.

- 1) Build your pivot as usual.
- 2) Add a new Computed Item named "Reason" using the following formula:

```
if ( ASP <= 100 ) { 'Too low for profit' } else { '' }
```

You need the else clause to avoid seeing "N/A" displayed in the pivot where the comparison doesn't hold true. You can, of course, add nested if-then-else logic to provide multiple messages for different conditions.

Section Type: Pivot section

Tip Name: Adding symbolic annotations to a Pivot

	1996			
	Units	Revenue	ASP	Flag
Q1	2,475	\$300,433.25	\$121.39	
Q2	2,805	\$299,895.05	\$106.91	
Q3	2,930	\$289,666.45	\$98.86	←
Q4	3,300	\$334,764.95	\$101.44	

Purpose / Use: Sometimes, you want your users to focus on an item by tagging it with a marker or symbol.

Sample File: Pivot annotations.bqy

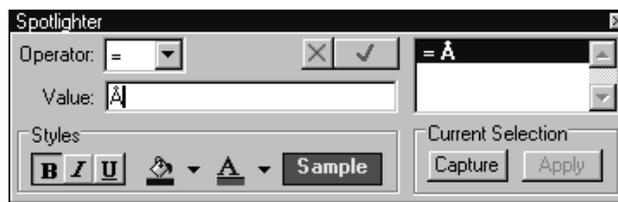
Steps to Produce: Use a combination of a simple Computed Item along with some creative use of fonts.

- 1) Build your pivot as usual.
- 2) Add a new Computed Item named "Flag" using the following formula:

```
if ( ASP <= 100 ) { 'A' } else { ' ' }
```

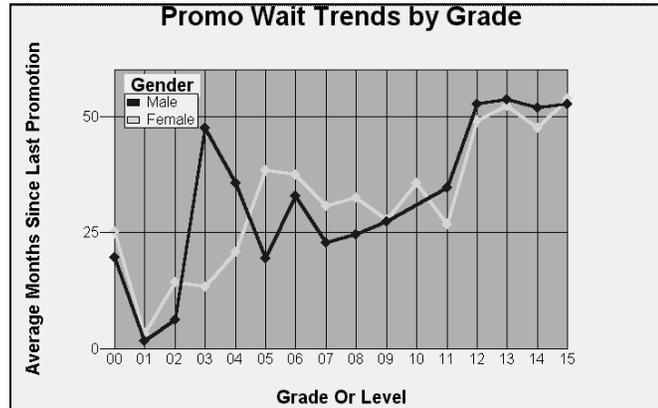
You need the else clause to avoid seeing "N/A" displayed in the pivot where the comparison doesn't hold true.

- 3) Now, invoke the Character Map windows application (Start | Programs | Accessories | Character Map). In the upper left, select an appropriate symbol font. In this example, Wingdings 3 was used.
- 4) Find the symbolic character you want to appear in your pivot. Click on it and hit the "Select" and "Copy" buttons to copy the symbol to your clipboard. Close the Character Map application.
- 5) Back in your pivot, right-click on the "Flag" column and select "Modify" from the speed menu. Highlight only the A in the formula and then press Ctrl-V to paste the selected symbol into your formula. Close the formula box.
- 6) With the "Flag" column still selected, change the font to match the one you used in the Character Map application. In this example, select Wingdings 3.
- 7) You can also use the Spotlighter on this strange character. With the "Flag" column still selected, simply invoke the Spotlighter (Format | Spotlighter). Set the desired display colors and use Ctrl-V to paste the selected symbol into the Value box.



Section Type: Pivot section (yes, the pivot section)

Tip Name: Demographic Analysis



Purpose / Use: To graphically display trends of average values in an organization over time or across different groups.

Sample File: Demographics.bqy

Steps to Produce: What appears as a simple chart may in fact require some imagination to set up. In this case, preliminary work includes building a Computed Item to transform the data into useable form, and then building the analysis as a Pivot first. We then convert the Pivot to a Chart.

- 1) In the Results section, create a Computed Item. Call it Months Since Last Promotion, and define its formula to be:

MONTHS_BETWEEN (Today, Last_Promotion_Date)

- 2) Insert a new Pivot. Add the category as the left-side label, and the analysis grouping as the top label. Place the new Computed Item (Months Since Last Promotion, in our example) in as the data element.

- 3) Right-click on the data column and select Data Function from the speed menu. Change the data function to be Average.

Gender	Male	Female
Grade Or Level	Avg Months Since Last Promotion	Avg Months Since Last Promotion
00	20	25
01	2	3
02	6	14
03	48	14
04	36	14
05	20	14
06	33	14
07	23	14
08	25	14
09	28	14
10	28	14
11	35	14
12	53	49

- 4) On the Brio main menu, select Insert | Chart This Pivot.

Section Type: Pivot section

Tip Name: Fact-based Delta Trends

SOF Type	Region	Host Country	Chg 1993	Chg 1994	Chg 1995	Chg 1996	Chg 1997	Chg 1998	Chg 1999	Chg 2000	Avg Annual Chg			
Non-Core	RBLAC	ARGENTIN	100.00%	-16.17%	186.17%	96.00%	-2.44%	-90.47%	-100.00%	0.00%	21.64%			
		BARBADOS	0.00%	100.00%	113.03%	-43.70%	-100.00%	0.00%	0.00%	0.00%	8.87%			
		BELIZE	100.00%	54.97%	77.71%	-11.68%	6.03%	-92.71%	-100.00%	0.00%	4.29%			
		BOLIVIA	1602.07%	160.03%	-36.09%	8.71%	53.47%	-78.81%	-100.00%	0.00%	201.20%			
		BRAZIL	8553.48%	13.88%	-23.89%	87.68%	47.05%	-46.11%	-81.32%	-100.00%	1058.85%			
		CENTRAL	0.00%	0.00%	0.00%	0.00%	100.00%	-100.00%	0.00%	0.00%	0.00%			
		CHILE	100.00%	95.71%	25.98%	26.21%					0.00%	10.41%		
		COLOMBIA	2642.35%	145.38%	327.79%	-27.78%					0.00%	375.00%		
		COSTA RIC	1775.22%	435.37%	29.30%	141.94%					0.00%	263.58%		
		CUBA	-54.42%	5404.39%	-65.26%	-63.55%					0.00%	633.35%		
		DOMINICAI	-57.29%	385.83%	310.08%	-22.52%					-100.00%	51.39%		
		ECUADOR	0.00%	100.00%	188.09%	-100.00%					0.00%	23.51%		
		EL SALVAD	84.63%	-2.38%	-73.37%	32.44%					0.00%	5.13%		
		GUATEMAL	0.00%	0.00%	100.00%	604.74%					-59.18%	83.97%		
		GUYANA	100.00%	424.39%	-17.90%	117.87%					0.00%	67.08%		
		HAITI	0.00%	0.00%	0.00%	0.00%					0.00%	-2.74%		
		HONDURAS	233.74%	34.84%	-10.52%	198.99%					-100.00%	57.71%		
		JAMAICA	0.00%	0.00%	0.00%	100.00%					0.00%	-10.10%		
		LATIN AME	-39.65%	452.01%	52.75%	29.64%					-52.55%	-100.00%	0.00%	42.78%
		MEXICO	0.00%	100.00%	1303.42%	-55.74%	171.08%	-84.01%	-100.00%	0.00%	0.00%	166.85%		
NICARAGU	0.00%	0.00%	100.00%	704.17%	93.29%	-97.44%	-100.00%	0.00%	0.00%	87.50%				
PANAMA	0.00%	100.00%	131.85%	750.76%	-9.53%	-68.26%	-31.75%	-100.00%	0.00%	96.63%				
PARAGUAY	0.00%	0.00%	0.00%	100.00%	-100.00%	0.00%	0.00%	0.00%	0.00%	0.00%				
PERU	308.60%	29.40%	82.24%	79.09%	27.67%	-81.78%	1.19%	-100.00%	0.00%	43.30%				
TRINIDAD	0.00%	100.00%	12763.19%	-71.36%	24.53%	-78.70%	0.00%	-100.00%	0.00%	1579.71%				

Purpose / Use: To analyze percent changes from one year to the next, and highlight those areas where the average year-to-year percent change is greater than a particular target.

Sample File: Not available to the public

Steps to Produce: This is another analysis that takes some preliminary set up before it can be created.

- 1) In the Result section, break out the dollar amounts by year (i.e., create dimensionalized facts). For example, add a new Computed Item for 1992 called Rsrc_1992. Define the formula as:

```
if (Budget_Year='1992') {Gross_Budget} else {0}
```

Repeat the creation of a dimensionalized fact for each year you want to analyze.

- 2) Insert a new Pivot. Add the appropriate categories to the left-side labels. Add the dimensionalized facts to the data area in the order they should be analyzed.
- 3) For each year-to-year comparison, add a Computed Item to the Pivot. For example, for the 1992-to-1993 delta, name the item Change 1993, and define its formula to be:

```
if ((Rsrc_1992 != null) && (Rsrc_1992 != 0)) {
    if (Rsrc_1993 != null) {
        (Rsrc_1993 - Rsrc_1992) / Rsrc_1992
    } else { -1 }
} else {
    if (Rsrc_1993 != null) { 1 }
    else { 0 }
}
```

(Fact-based Delta Trends continued)

The formula's logic is as follows. If dollar amounts exist for both years, perform a normal % change calculation (i.e., (Current-Previous)/Previous). If an amount exists for the prior year but not the following year, tag that as a delta of -100%. If no amount exists for the prior year but one exists for the following year, tag that as a delta of +100%.

- 4) Add a final Computed Item to calculate the average delta across the entire range of years. In our example, name it Avg_ Annual_Chg, and define its formula as:

$$(\text{Chg}_{1993} + \text{Chg}_{1994} + \text{Chg}_{1995} + \text{Chg}_{1996} + \text{Chg}_{1997} + \text{Chg}_{1998} + \text{Chg}_{1999} + \text{Chg}_{2000}) / 8$$

- 5) Select the Avg_ Annual_Chg column, right-click on it and change the numeric display to be 0.00%.
- 6) With the Avg_ Annual_Chg column still selected, open the Spotlighter and define the appropriate highlight definition. For example, say we wanted to highlight any average change greater than 10%, you would define the Spotlighter as shown below:

g 1996	Chg 1997	Chg 1998	Chg 1999	Chg 2000	Avg Annual Chg
					21.64%
					8.67%
					4.29%
					201.20%
87.68%	47.05%	-46.11%	-61.32%	-100.00%	1058.85%

Remember that Spotlighter values are defined in terms of the "raw" unformatted numbers. Therefore, you need to enter the target value as 0.1 (the raw equivalent of 10%).

Section Type: Pivot section

Tip Name: Dimension-based Delta Trends

		1999										
		Q1			Q2			Q3			Q4	
		Unit Sales	Inc/Dec	Pct Inc/Dec	Unit Sales	Inc/Dec	Pct Inc/Dec	Unit Sales	Inc/Dec	Pct Inc/Dec	Unit Sales	Inc/Dec
Books	Americas	266,618			276,523	9,905	4%	298,008	21,485	8%	494,923	196,915
	Asia Pacific	128,853			141,344	12,491	10%	158,560	17,216	12%	359,972	201,412
	Europe	144,800			149,023	4,223	3%	235,426	86,403	58%	373,889	138,463
	Total	540,271			566,890	26,619	5%	691,994	125,104	22%	1,228,784	536,790
Music	Americas	125,676			128,669	2,993	2%	132,958	4,289	3%	226,451	93,493
	Asia Pacific	71,572			74,150	2,578	4%	84,775	10,625	14%	204,779	120,004
	Europe	73,850			87,794	13,944	19%	129,323	41,529	47%	211,307	81,984
	Total	271,098			290,613	19,515	7%	347,056	56,443	19%	642,537	295,481
Videos	Americas	52,301			60,820	8,519	16%	61,809	989	2%	111,265	49,456
	Asia Pacific	31,041			37,256	6,215	20%	37,590	334	1%	83,149	45,559
	Europe	32,679			35,755	3,076	9%	57,818	22,063	62%	91,386	33,568
	Total	116,021			133,831	17,810	15%	157,217	23,386	17%	285,800	128,583
Total		927,390			991,334	63,944	7%	1,196,267	204,933	21%	2,157,121	960,854

Purpose / Use: To analyze percent changes from one quarter to the next by keying off the changes in the dimension labels.

Sample File: Dimension-based deltas.bqy

Steps to Produce: This is an analysis that takes advantage of the unique properties of the Next() and Prior() functions when they are used in a Pivot section.

- 1) Create your pivot as usual; with the dimension(s) you want to track the changes over placed in the top labels.
- 2) Create a computed item in the Pivot that tracks the actual change amounts from one top dimension value to the next. In this example, we are tracking the changes from one quarter to the next, so create a computed item named "Inc/Dec" with a formula of:

```
if (Quarter=="Q1") {}
else { Unit_Sales-Prior(Unit_Sales) }
```

- 3) Create a second computed item in the pivot to calculate the amount of change as a percentage of the previous quarter's value. Give the computed item a name of "Pct Inc/Dec" and use a formula of:

```
if (Quarter=="Q1") {}
else { Inc_Dec / Prior(Unit_Sales) }
```

- 4) Set the display of the Pct Inc/Dec column by right-clicking on it, selecting the Number option and setting the number format to #,##0.0%.

Section Type: Pivot section
 Tip Name: Reporting the Break % of the Total

AZ

Product Category	Amount	Pcnt of State	Pcnt of Grand
CD-ROM Drive	\$36,054	32.1%	1.3%
Hard Drive	\$14,250	12.7%	0.5%
Key Board	\$7,485	6.7%	0.3%
Modem	\$44,884	40.0%	1.6%
Mouse	\$1,836	1.6%	0.1%
Speakers	\$7,701	6.9%	0.3%
	\$112,210	100.0%	4.1%

CA

Product Category	Amount	Pcnt of State	Pcnt of Grand
CD-ROM Drive	\$282,527	23.2%	10.2%
Hard Drive	\$443,534	36.4%	16.0%
Key Board	\$87,711	7.2%	3.2%
Key Pad	\$9,601	0.8%	0.3%
Modem	\$344,793	28.3%	12.4%
Mouse	\$34,510	2.8%	1.2%
Speakers	\$14,810	1.2%	0.5%
	\$1,217,486	100.0%	44.0%

Purpose / Use: To create a report that properly displays the percent of the grand total for each grouping.

Sample File: Break percent of total.bqy

Steps to Produce: The Report section by itself cannot actually produce this report with the correct numbers. You need to embed a special pivot into the report to replace the default table object.

- 1) In the Results section, create a Computed Item to calculate the grand total of the revenue. Name this item "Total Amt" and use the following formula:

$$\text{Sum (Amount)}$$

- 2) Next, create a second Computed Item named "Pcnt of Grand" using the formula:

$$\text{Amount / Total_Amt}$$

- 3) Now, insert a pivot section.
 - Place one or more dimensions into the side labels
 - Place "Amount" into the Facts twice. Right-click on the second "Amount2" column and change the data function to be "% of Column".
 - Add the "Pcnt of Grand" column to the Facts.
 - Click on the Side Label column and then click on the toolbar's Sigma button.

(Reporting the Break % of the Total continued)

Your pivot should now look something like the figure shown below.

	Amount	Amount2	Pcnt of Grai
CD-ROM Drive	\$698,668	25%	25.22%
Hard Drive	\$851,758	31%	30.75%
Key Board	\$187,772	7%	6.78%
Key Pad	\$15,268	1%	0.55%
Modem	\$879,569	32%	31.75%
Mouse	\$71,962	3%	2.60%
Speakers	\$65,164	2%	2.35%
Total	\$2,770,161	100%	100.00%

- 4) Next, format the pivot to look more like a Report section's table object.
 - Double-click on the "Amount2" column name and change it to "Pcnt of State".
 - Right-click on the "Pcnt of State" column itself and set the Number format to "#,##0.0%". Set the Number format for "Pcnt of Grand" to be the same.
 - Right-click anywhere in the Pivot, select "Corner Labels" from the speed menu and set it to "Side".
 - Click on the Side Labels column, set the border type to "Horz and Vert" (use the menu selection Format | Borders | Horz and Vert), the fill color to "Transparent", the text color to black and the font style to regular.
 - Click on the cell that says "Total" and set its text color to "Transparent". Then Alt-Click on the Total cell and set the row's font style to bold.
 - Click on all top column labels, set the fill color to blue, the text color to white, and the border type to "Horz and Vert". Next center-align and middle-align (Format | Justify | Middle) the text. Adjust the height of the column labels so that all the text displays properly.

Your pivot should now look something like the figure below.

Product Category	Amount	Pcnt of State	Pcnt of Grand
CD-ROM Drive	\$698,668	25.2%	25.2%
Hard Drive	\$851,758	30.7%	30.7%
Key Board	\$187,772	6.8%	6.8%
Key Pad	\$15,268	0.6%	0.6%
Modem	\$879,569	31.8%	31.8%
Mouse	\$71,962	2.6%	2.6%
Speakers	\$65,164	2.4%	2.4%
	\$2,770,161	100.0%	100.0%

- 5) Insert a Report section and add "State" to Report Group 1. Delete the table object in the body and then drag the pivot into the body instead. The SmartReport feature will automatically adjust the "Pcnt of Grand" column correctly.

Section Type: Pivot section

Tip Name: Profit and Loss Reporting

		1998												
		Total	Total Equity Derivs	Total Converts	Us Listed Shares	Nasdaq	Euro Shares	Uk Shares	Japan Shares	Asian Shares	Latin America Shares	Canada Shares	New Markets	Global Sec Scvs
Revenue	Commissions	521.8	25.4	12.7	76.1	177.5	147.0	14.7	18.3	14.6	7.3	7.6	9.2	11.5
	Cust Bal/Stock Loan	8,586.2	417.2	208.6	1,251.6	2,920.4	2,419.7	242.0	300.4	240.3	120.2	125.2	151.4	189.3
	ECM Block Trades	9,447.7	459.1	229.5	1,377.2	3,213.4	2,662.5	266.3	330.5	264.4	132.2	137.7	166.6	208.3
	Other Income	12.3	0.6	0.3	1.8	4.2	3.5	0.3	0.4	0.3	0.2	0.2	0.2	0.3
	Trading Income	44.7	2.2	1.1	6.5	15.2	12.6	1.3	1.6	1.3	0.6	0.7	0.8	1.0
	Underwriting Income	310.5	15.1	7.5	45.3	105.6	87.5	8.7	10.9	8.7	4.3	4.5	5.5	6.8
	Total	18,923.2	919.5	459.7	2,758.4	6,436.2	5,332.9	533.3	662.0	529.6	264.8	275.8	333.8	417.2
Variable Expense	Brokerage and Clearance	34.2	1.7	0.8	5.0	11.6	9.6	1.0	1.2	1.0	0.5	0.5	0.6	0.8
	Error Expense	192.1	9.3	4.7	28.0	65.3	54.1	5.4	6.7	5.4	2.7	2.8	3.4	4.2
	PCS Compensation	252.1	12.2	6.1	36.7	85.7	71.0	7.1	8.8	7.1	3.5	3.7	4.4	5.6
	Soft Dollar Charges	51.7	2.5	1.3	7.5	17.6	14.6	1.5	1.8	1.4	0.7	0.8	0.9	1.1
	Total	530.1	25.8	12.9	77.3	180.3	149.4	14.9	18.5	14.8	7.4	7.7	9.3	11.7
Departmental Expense	Other Net Departmental	263.5	12.8	6.4	38.4	89.6	74.2	7.4	9.2	7.4	3.7	3.8	4.6	5.8
	Sales Net Departmental	288.0	14.0	7.0	42.0	97.9	81.2	8.1	10.1	8.1	4.0	4.2	5.1	6.3
	Trading Net Departmental	304.1	14.8	7.4	44.3	103.4	85.7	8.6	10.6	8.5	4.3	4.4	5.4	6.7
	Total	855.5	41.6	20.8	124.7	291.0	241.1	24.1	29.9	23.9	12.0	12.5	15.1	18.9
Other Expense	Enterprise	312.6	15.2	7.6	45.6	106.3	88.1	8.8	10.9	8.7	4.4	4.6	5.5	6.9
	Investment Research	808.7	39.3	19.6	117.9	275.1	227.9	22.8	28.3	22.6	11.3	11.8	14.3	17.8
	Other Allocated Expense	55.2	2.7	1.3	8.0	18.8	15.6	1.6	1.9	1.5	0.8	0.8	1.0	1.2
	Total	1,176.5	57.2	28.6	171.5	400.2	331.6	33.2	41.2	32.9	16.5	17.1	20.8	25.9
Total	16,361.1	795.0	397.5	2,384.9	5,564.8	4,610.8	461.1	572.4	457.9	229.0	238.5	288.6	360.7	

Purpose / Use: To create a display that shows both debits and credits as positive numbers, while displaying the correct totals at the same time.

Sample File: *Not Available to the Public*

Steps to Produce: The solution is simple once you remember that display formatting is a separate issue from the actual internal numeric representation.

- 1) Build your pivot as you normally would.
- 2) Select each of the numeric facts, right-click and select "Number..." from the speed menu.
- 3) Select the "Custom" category and create a new custom format as:

#,###.0;#,###.0

In other words, the negative value format is set identical to the positive value format.

- 4) Click "OK" to apply the new format.
- 5) The Spotlights can be used to add color highlighting so that, if desired, conditional formatting can still identify negative numbers.

Section Type: Report section
 Tip Name: Maintaining relative position

Books

Product Family	Units	Dollars
Arts and Music	325,669	\$8,122,192
Audiobooks	285,644	\$5,545,267
Biographies and Memoirs	414,443	\$10,668,680
Business and Investing	260,056	\$9,461,796
Childrens Books	268,846	\$3,330,005
Computers and Internet	439,271	\$17,957,634
Entertainment	284,109	\$7,377,248
Food and Wine	234,792	\$5,135,048
Health Mind and Body	303,340	\$4,102,254
History	366,017	\$8,763,217
Home and Garden	379,468	\$8,498,261
Sub-Totals:	3,561,655	\$88,961,602

Music

Product Family	Units	Dollars
Alternative	236,928	\$3,077,855
Blues	137,448	\$1,899,121
Classical	309,740	\$6,574,707
Country	289,056	\$4,444,103
Dance and DJ	195,562	\$2,739,085
International	290,106	\$3,702,468
Jazz	331,910	\$5,032,575
Sub-Totals:	1,790,750	\$27,469,914

Videos

Purpose / Use: To keep objects in the same spatial relationship, even when the size of the report group changes due to changes in the number of items displayed in each group.

In this example, we want to add a horizontal line and some custom text to the summary area of a report section's table object. The table object will grow and shrink from group to group, but we want the line and text to always be at the bottom of the table.

Sample File: Spring objects.bqy

Steps to Produce: Use the "Spring Objects" formatting option to maintain the relative position between two objects.

- 1) Build your report as usual.
- 2) Add a text object and a line object to the table in the first group. Position the objects into the desired location relative to the first table object.
- 3) Click on the text object, then shift-click on the table object to select both items. (Make sure you selected them from the same group.)
- 4) Use the main menu and select Format | Spring Objects.
- 5) Click on the line object, then shift-click on the table object to select both items. (Make sure you selected them from the same group.)
- 6) Use the main menu and select Format | Spring Objects.
- 7) Repeat the process for any other objects you want to hold in position. (Spring Objects only works with 2 objects at a time.)

Section Type: Report section
 Tip Name: Custom group summaries

Books		
Product Family	Units	Dollars
Arts and Music	325,669	\$8,122,192
Audiobooks	285,644	\$5,545,267
Biographies and Memoirs	414,443	\$10,668,880
Business and Investing	260,056	\$9,461,796
Childrens Books	268,846	\$3,330,005
Computers and Internet	439,271	\$17,957,634
Entertainment	284,109	\$7,377,248
Food and Wine	234,792	\$5,135,048
Health Mind and Body	303,340	\$4,102,254
History	366,017	\$8,763,217
Home and Garden	379,468	\$8,498,261
Sub-Totals for all Books:	3,561,655	\$88,961,602
Number of different Books items:	7,880	

Music		
Product Family	Units	Dollars
Alternative	236,928	\$3,077,855
Blues	137,448	\$1,899,121
Classical	309,740	\$6,574,707
Country	289,056	\$4,444,103
Dance and DJ	195,562	\$2,739,085
International	290,106	\$3,702,468
Jazz	331,910	\$5,032,575
Sub-Totals for all Music:	1,790,750	\$27,469,914
Number of different Music items:	4,416	

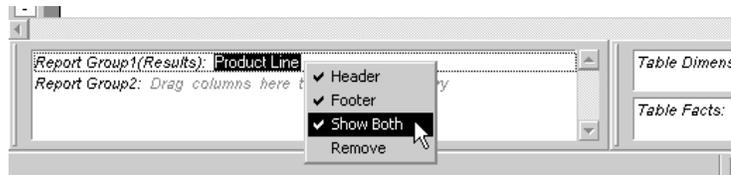
Videos

Purpose / Use: Sometimes you want group summary information in a format different from what the report section's table object provides.

Sample File: Spring objects.bqy

Steps to Produce: Create a group footer, and then craft whatever summaries you want by simply dragging data elements into the group footer.

- 1) Build your report as usual.
- 2) For each fact column, click on the column, then right-click and click on the "Show Column Total" menu item to turn off that option. When finished, your table object should have no totals showing for any columns.
- 3) In the group portion of the Outliner, right-click on the group for which you want the custom summary to appear. Select "Show Both" from the header/footer options menu.



- 4) Drag any desired data elements into the group footer. Any numeric field will automatically be totaled.
- 5) You can add text fields as summary data as well. Drag a text field into the group footer, then right-click on the field. Select Data Function | Count from the speed menu that appears. The field in the group footer will now display the number of times that item appears within the group.
- 6) Add appropriate text objects to the group footer to serve as labels.

Section Type: Report section

Tip Name: Multi-Line Column Reports (using table objects)

Americas					
Store Name	Street Address	City	State/Prov	Postal Code	Country
Manager	Phone Number	Fax Number			
Barreiras	2853 Bailey Rd	Barreiras	Bahia	40938	Brazil
Vilhena	262-555-5124	262-555-5121			
Buenos Aires	5203 Catanzaro Way	Buenos Aires	Buenos Aires	40837	Argentina
Maia	605-555-8203	605-555-8201			
Vancouver	1501 Ramsey Circle	Vancouver	British Columbia	81392	Canada
Green	801-555-4324	801-555-4321			
Santos	1250 Coggins Drive	Santos	São Paulo	40258	Brazil
Redentor	509-555-1596	801-555-4321			
Brooklyn	1077 Wharf Drive	Brooklyn	New York	92094	USA
White	477-555-7967	477-555-7961			
Westwood	1028 Century Blvd	Los Angeles	California	98079	USA
Wilson	797-555-3417	797-555-3411			
New York	673 Fifth Avenue	Manhattan	New York	10023	USA
Wah	212-555-1702	151-555-1701			
Los Angeles	5179 Flower Street	Los Angeles	California	670909	USA
Lee	977-555-2724	977-555-2721			
Anaheim	3173 Buena Vista Ave	Anaheim	California	90493	USA

Purpose / Use: To display fields on multiple lines per record in a column-oriented report.

Sample File: Multi-line wrapping columns.bqy

Steps to Produce: Use multiple table objects, one table object per line desired.

- 1) Create groupings so that the body area will hold only one record per group.
- 2) Turn off the group headers for all groups except the top-level group. Right-click on each sub-group in the group area of the Outliner and select "Header" to turn it off.
- 3) In the table definition area of the Outliner, drag all the first-line fields into the Table Dimensions. Format the field widths as desired.
- 4) Select any column in the table object, right-click and choose "Show Column Titles" to turn off the table-based column titles.
- 5) With the table still selected, choose Format | Border and Background from the main menu, and set the Border style to "None". Select Format | Grid Lines, and set the Vertical Grid Lines to "None".
- 6) From the main menu, select Report | Insert Table, and draw out a second table in the report's body area.
- 7) Select the new table object and drag all the second-line fields into the Table Dimensions area of the Outliner. Format the field widths as desired.
- 8) Select any column in the second table object, right-click and choose "Show Column Titles" to turn off the table-based column titles. Choose Format | Border and Background from the main menu, and set the Border style to "None". Select Format | Grid Lines, and set the Vertical Grid Lines to "None".
- 9) Add a text object to the top-level header area to serve as column labels. Format as needed to align the labels with the columns in the tables.

Section Type: Report section

Tip Name: Multi-Line Column Reports (using field objects)

Americas					
Store Name Manager	Street Address	City Phone Number	State/Prov Fax Number	Postal Code	Country
Barreiras Vilhena	2853 Bailey Rd	Barreiras 262-555-5124	Bahia 262-555-5121	40938	Brazil
Buenos Aires Mala	5203 Catanzaro Way	Buenos Aires 605-555-8203	Buenos Aires 605-555-8201	40837	Argentina
Vancouver Green	1501 Ramsey Circle	Vancouver 801-555-4324	British Columbia 801-555-4321	81392	Canada
Santos Redentor	1250 Coggins Drive	Santos 509-555-1596	São Paulo 801-555-4321	40258	Brazil
Brooklyn White	1077 Wharf Drive	Brooklyn 477-555-7967	New York 477-555-7961	92094	USA
Westwood Wilson	1028 Century Blvd	Los Angeles 797-555-3417	California 797-555-3411	98079	USA
New York Wah	673 Fifth Avenue	Manhattan 212-555-1702	New York 151-555-1701	10023	USA
Los Angeles Lee	5179 Flower Street	Los Angeles 977-555-2724	California 977-555-2721	670909	USA
Anaheim Patt	3173 Buena Vista Ave	Anaheim 949-555-2447	California 949-555-2444	90493	USA

Purpose / Use: To display fields on multiple lines per record in a column-oriented report, when using multiple table objects will not produce the desired effect.

In this example, the user wants the Street Address field to wrap to 2 lines. All other "second line" fields should appear on the same line as the second line of the Street Address. Because this is the only multi-line field, and because it appears in the middle of the list of fields, multiple table objects won't create the desired effect.

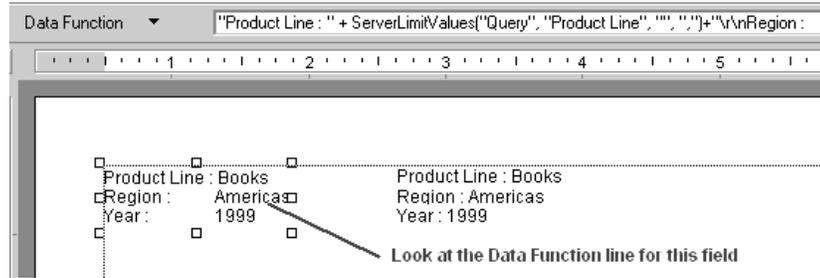
Sample File: Multi-line wrapping columns.bqy

Steps to Produce: Delete the default table object and drag individual field objects into the report body, formatting them as needed.

- 1) Create report groupings so that the body area will hold only one record per group.
- 2) Turn off the group headers for all groups except the top-level group. To do this, right-click on each sub-group in the group area of the Outliner and select "Header" to turn it off.
- 3) Select the blank default table object and delete it.
- 4) Drag the desired fields directly into the body area of the report (instead of into the Outliner). This will create a free-form report design.
- 5) Format and align the fields as desired.
- 6) To set any field to automatically word-wrap, right-click on the field and select "Properties". Select the "Alignment" tab in the dialog box that appears. Set Horizontal alignment to Left, Vertical alignment to Top, and turn on the Wrap text checkbox.

Section Type: Report section

Tip Name: Multi-Line fields (using JavaScript codes for new line)



Purpose / Use: To create a single text object that wraps at a pre-determined point in the text, not simply when the width of the text object has been exceeded. In this example, we want the various query limits of the report displayed.

Sample File: Multi-line fields.bqy

Steps to Produce: Use the special JavaScript codes for carriage return and new line as part of the text object’s formula.

- 1) Create a header by selecting Report | Headers and Footers | Page Header.
- 2) Drag one or more field objects directly into the report header area (instead of into the Table Outliner). This will create text objects with the correct JavaScript code to display the selected query limit. For our example, click on the plus sign next to “Fields” in the catalogue, and drag Query Limit into the page header. Do this once each for Product Line, Region and Year. Each object will include the name of the field, and its current limit setting.
- 3) Click on the Product Line object in the page header. Add a colon to the text in the front of the formula, so that it now reads “Product Line: “.
- 4) At the end of the formula, add a + sign, and the text for the JavaScript codes for carriage return and new line, which are “\r\n”. The formula for the Product Line object should now look like this:

```
"Product Line: " + ServerLimitValues("Query", "Product Line", "", "")+"\r\n"
```

- 5) Select the Region object, and use the Ctrl-C key combination to copy its formula to the Windows clipboard.
- 6) Select the Product Line object, go to the end of the formula text, and then hit Ctrl-V to paste in the formula for Region. Edit the formula so it now reads:

```
"Product Line: " + ServerLimitValues("Query", "Product Line", "", "")+"\r\nRegion: " + ServerLimitValues("Query", "Region", "", "")
```

(Multi-Line fields using JavaScript codes for new line continued)

- 7) As before, add a +"\r\n" to the end of the formula.
- 8) Select the Year field object, and use the same copy and paste logic to add it to the Product Line object's formula, so it now reads:

```
"Product Line: " + ServerLimitValues("Query", "Product  
Line", "", ",")+"\r\nRegion: " + ServerLimitValues("Query",  
"Region", "", ",")+"\r\nYear: " +  
ServerLimitValues("Query", "Year", "", ",")
```

- 9) Clean up the report design by deleting the extra Region and Year fields.

Section Type: Report section

Tip Name: Multi-Line fields (using word wrap formatting)

Int Sales	Included Categories:
42,650.99411	Architecture, Drama, Photography,
97,581.49309	Theater
740,232.4872	

Look at the Alignment Properties for this field

Purpose / Use: To provide a comma-delimited list of values that automatically word-wrap.

Sample File: Multi-line fields.bqy

Steps to Produce: Drag a field object onto the report design, then set its formatting to word-wrap as needed.

- 1) Design your report as usual.
- 2) Drag a field object directly onto the report design surface, instead of into the Table Outliner. If a text field is used, Brio will automatically display all unique values in a comma-delimited string.
- 3) Right-click on the field and select "Properties". Select the "Alignment" tab in the dialog box that appears. Set Horizontal alignment to Left, Vertical alignment to Top, and turn on the Wrap text checkbox.

Section Type: Report section

Tip Name: Full-Row Spotlighting

1996			
Product Category	Units	Dollars	ASP
CD-ROM Drive	2,210	\$275,358	\$124.60
Hard Drive	2,085	\$314,995	\$151.08
Key Board	1,315	\$75,726	\$57.59
Key Pad	465	\$14,638	\$31.48
Modem	3,450	\$457,725	\$132.67
Mouse	1,500	\$28,855	\$19.24
Speakers	485	\$57,463	\$118.48
	11,510	\$1,224,760	\$106.41

Purpose / Use: To highlight an entire row of data when a particular field item is out of scope.

Sample File: Full row spotlighting.bqy

Steps to Produce: Build a field-object based report that is formatted to look like a table report, add a second copy of the spotlighting field and set it as a full-row background. (Requires Brio Intelligence v6.1.0 or higher.)

1. Build your query as normal and process it to fetch a representative data sample.
2. Insert a new Report section, select the default table object and delete it. For ease of design, you may want to display the report's section boundaries in the report by selecting the Report | Section Boundaries menu selection.
3. Add any grouping fields you want to use into the Report Groups area of the Outliner.
4. In your Results section, add a computed item named "One" using a formula of 1. Add this field as the lowest report group (so far), and then delete the display of this field from the report group's header.
5. Select a field to be the "group-controlling field". This is usually the left-most field in your report "table". In this example, the controlling field is the "Product Category". Add this field into the Report Groups area of the Outliner as the last group field. Delete the display of this group-controlling field from the report group header.
6. Drag each "table" field into the group-controlling field's report group header and format as desired. Your report should look like this so far:

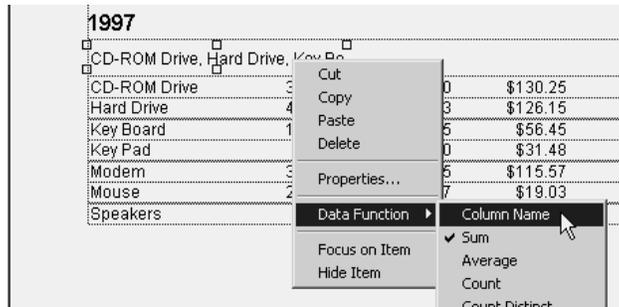
1997				Report Group1 Header: (Results) Fiscal Year
				Report Group2 Header: (Results) One
CD-ROM Drive	3,250	\$423,310	\$130.25	Report Group3 Header: (Results) Product Category
Hard Drive	4,255	\$536,763	\$126.15	Report Group3 Header: (Results) Product Category
Key Board	1,985	\$112,045	\$56.45	Report Group3 Header: (Results) Product Category
Key Pad	20	\$630	\$31.48	Report Group3 Header: (Results) Product Category
Modem	3,650	\$421,845	\$115.57	Report Group3 Header: (Results) Product Category
Mouse	2,265	\$43,107	\$19.03	Report Group3 Header: (Results) Product Category
Speakers	65	\$7,701	\$118.48	Report Group3 Header: (Results) Product Category

Report Group1(Results): Fiscal Year
 Report Group2(Results): One
 Report Group3(Results): Product Category
 Report Group4: Drag columns here to create a category

Table Dimensions: To see table dimension columns, click on a your report
Table Facts: To see table fact columns, click on a table in you

(Full-Row Spotighting continued)

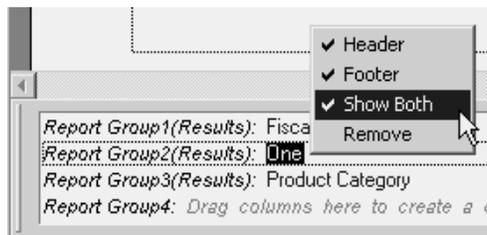
- In the group header associated with the "One" computed item group, drag in each of the "table" fields, positioning them over the corresponding data fields in the group header below. Right-click to change the data function to be "Column Name" and format to mimic a table header.



Now your report should look like this:

1997			
Product Category	Units	Dollars	ASP
CD-ROM Drive	3,250	\$423,310	\$130.25
Hard Drive	4,255	\$536,763	\$126.15
Key Board	1,985	\$112,045	\$56.45
Key Pad	20	\$630	\$31.48
Modem	3,650	\$421,845	\$115.57
Mouse	2,265	\$43,107	\$19.03
Speakers	65	\$7,701	\$118.48

- To add summary information at the bottom of the "table", right-click on the "One" group in the Outliner and turn on the group footer.



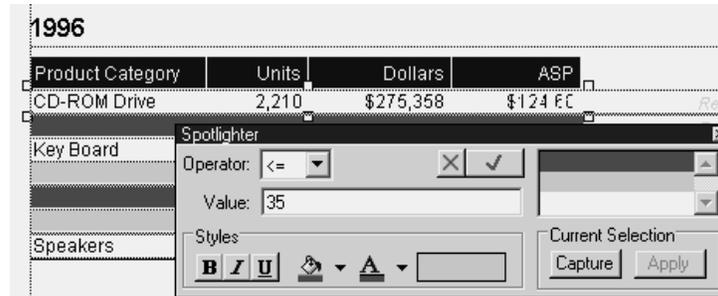
- Drag items for the summary into the "One" group's footer.
- Add the spotlighting field to the lowest level group header a second time. In this example, this is the "ASP" field.
- Set the format of the spotlighting field to be transparent text with a transparent background. Expand the field to cover the entire row.

1996			
Product Category	Units	Dollars	ASP
CD-ROM Drive	2,210	\$275,358	\$124.60
Hard Drive	2,200	\$314,000	\$142.73

- With the new spotlighting field still selected, invoke the Spotlighter via the Format | Spotlighter... menu selection.

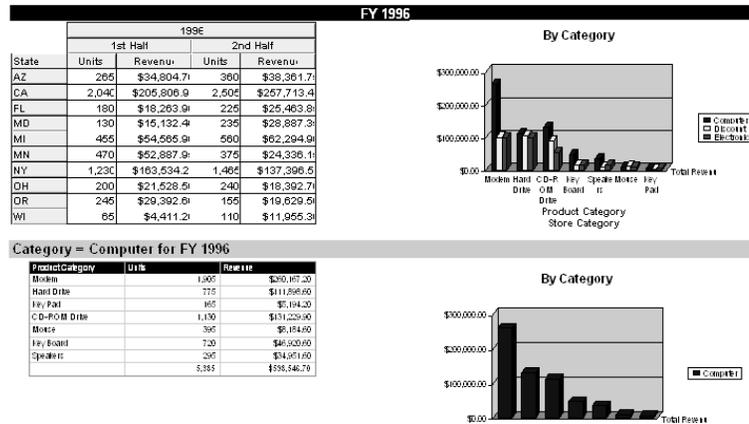
(Full-Row Spotighting continued)

13. Use the Spotighter to set the conditional formatting such that every condition is defined to use the same text and background colors. For example:



14. Send the new spotighting field to the background via the Format | Layer | Send to Back menu selection.
15. Finally, turn off the display of report section boundaries with the Report | Section Boundaries menu selection.

Section Type: Report section
 Tip Name: Context-Sensitive Charts / Pivots



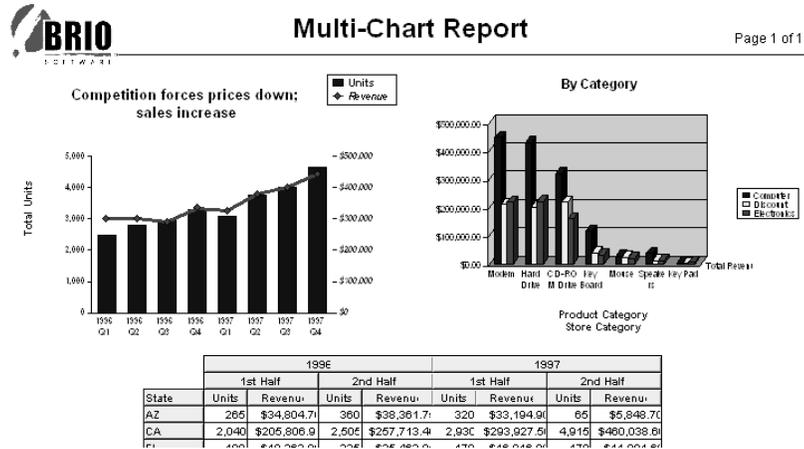
Purpose / Use: To create a report with charts or pivots specific to the data within a report group.

Sample File: Smartreports.bqj

Steps to Produce: Create a report with the appropriate groups, and then embed the desired charts or pivots into either the body area or into the desired group's header.

- 1) Create the charts and pivots to be used.
- 2) Create the report as usual, with one or more groups defined.
- 3) Drag a chart or pivot into the body area of the report to make it display only those values that pertain to the lowest level group defined.
- 4) Drag a chart or pivot into a report group header to make it display a summary of those values that pertain to selected group.
- 5) Use the main menu, and select Report | Headers and Footers | Report Header to create a report-wide header area. Drag a chart or pivot into the report header to make it display a global summary of all values in the report.

Section Type: Report section
 Tip Name: Displaying multiple charts / pivots on 1 page



Purpose / Use: Sometimes, you want to be able to see multiple charts or pivots on a single page.

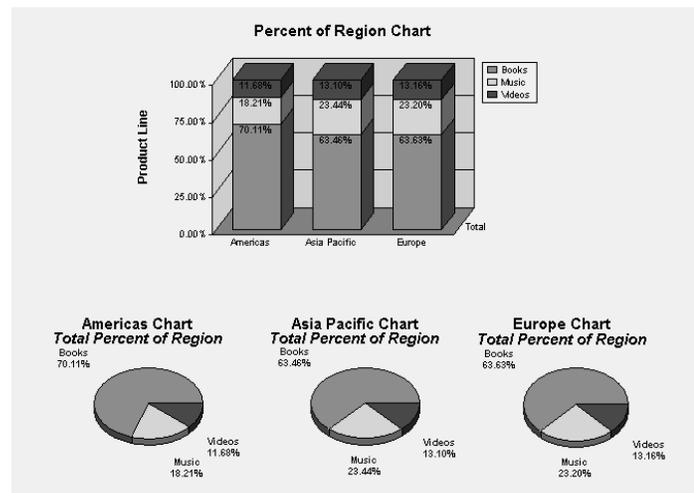
Sample File: Smartreports.bqy

Steps to Produce: Create a report section with no groups, and embed the desired charts and pivots in the body area.

- 1) Add a new Report section. Do not define any groups. (Groups will create context-sensitive charts and pivots.)
- 2) Select the blank default table object and delete it.
- 3) Drag all desired charts and pivots into the report's body area.
- 4) Format page headers, page footers and graphical elements as desired.

Section Type: Report section

Tip Name: Per-Portion Charts on 1 page



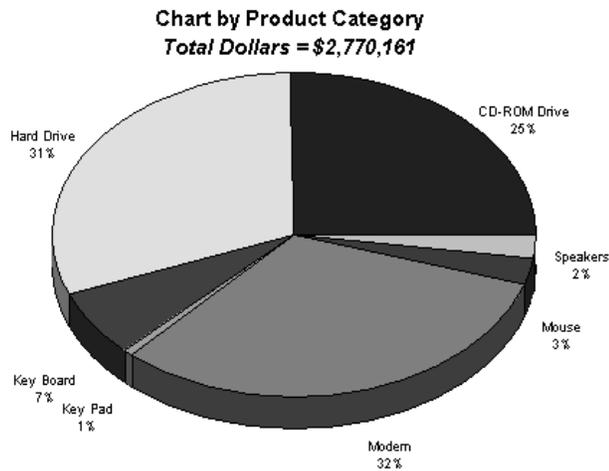
Purpose / Use: Display multiple charts that show the same data for different groups on a single page.

Sample File: 100% of region.bqy

Steps to Produce: Create a report section with no groups, and embed the desired charts and pivots in the body area.

- 1) Create your query and process it to fetch representative data.
- 2) For each chart group, create a new Table section from the Results, and then define a local limit to display only 1 group's data within each table section.
- 3) For each group Table section, build a chart to display the group data. Make sure every group chart is designed in exactly the same manner.
- 4) Add a new Report section. Do not define any groups. (Groups will create context-sensitive charts.)
- 5) Select the blank default table object and delete it.
- 6) Drag all desired charts into the report's body area.
- 7) Format page headers, page footers and graphical elements as desired.

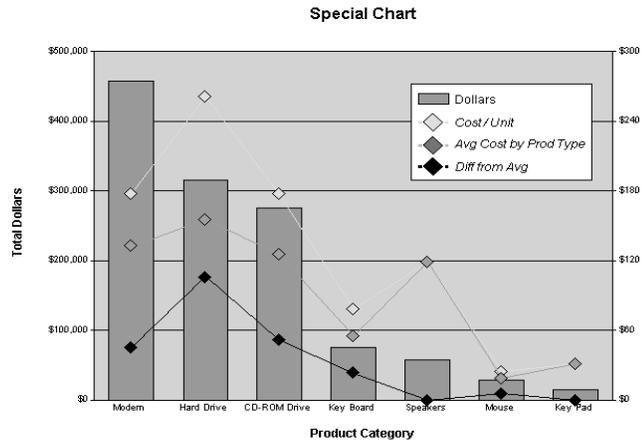
Section Type: Report section
Tip Name: Chart and Summary Data on 1 page



Purpose / Use: To display a chart and its summary details on a single page.
Sample File: Pie chart with totals.bqy
Steps to Produce: Create a report section with no groups, and embed the desired chart and summary data in the body area.

- 1) Build your chart as usual.
- 2) If you want to display the summary as if it were a part of the chart's sub-title, define a sub-title with a value of "Total Dollars = ". The extra spaces help center the text properly for its eventual display.
- 3) Insert a new Report section.
- 4) Select the blank default table object and delete it.
- 5) Drag the chart into the report's body area.
- 6) Drag the field object that represents the data to be summarized into the report's body area. Align the field so that it appears to be part of the chart's sub-title.

Section Type: Report section
 Tip Name: Multi-Line Bar Chart



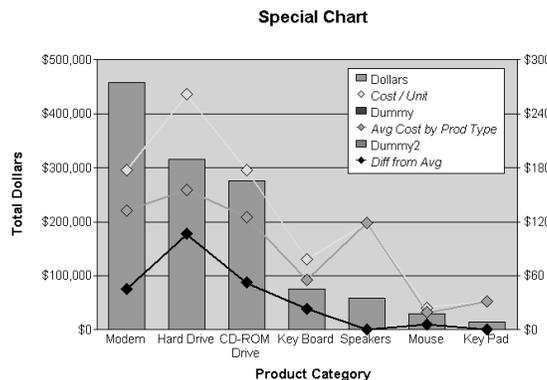
Purpose / Use: To display a Bar-Line Chart that has 1 bar and multiple lines.

Sample File: Multi-line bar chart.bqy

Steps to Produce: Create a report section with no groups, embed a specially formatted bar-line chart and design a custom chart in the report section.

The challenge stems from the fact that bar-line charts work with pairs of values, one being a bar and one being a line. The trick is getting the chart and its legend to display only the first value as a bar and all others as lines.

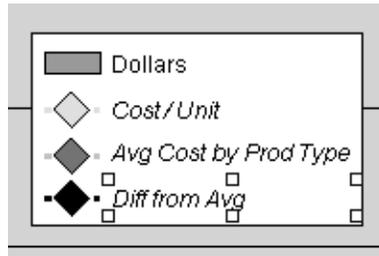
- 1) In the Results section, create a computed item named "Dummy" with a formula of 0.
- 2) Add a new chart and change its chart type to be a bar-line chart by selecting the last item in the chart type pulldown.
- 3) Drag the field to be displayed as a bar into the Outliner as the first item in the Y-Facts area.
- 4) Drag the field to be displayed as the first line into the Outliner as the next item in the Y-Facts area.
- 5) Create subsequent bar-line groupings by dragging the Dummy field as the bar (with a value of zero) and the next line-oriented field as its bar-line group partner. Your chart should now look like this:



(Multi-Line Bar Chart continued)

Note that the chart displays the data exactly as desired, but the legend displays the dummy fields as well. We'll use the Report section to override the way the legend is displayed.

- 6) Insert a new Report section. Do not define any groups. (Groups will create context-sensitive charts.)
- 7) Select the blank default table object and delete it.
- 8) Drag the chart into the report's body area.
- 9) Drag a box object on top of the chart's legend to hide it (or you can simply hide the legend by going into the chart section, right-clicking on the legend and selecting "Hide Legend" from the speed menu).
- 10) Use other report section objects to build a custom legend display from scratch. You may want to use embedded BMPs to display the proper legend "widgets" (such as diamonds, etc.)



Section Type: Report section (with an embedded pivot)

Tip Name: Calendar-style Reports

January 1999		Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
UNITS							35	15
AMOUNT							\$8,211.80	\$1,919.70
	3	4	5	6	7	8	9	
UNITS	55	N/A	N/A	25	85	35	25	
AMOUNT	\$8,133.90	N/A	N/A	\$3,737.00	\$9,065.80	\$4,479.30	\$787.00	
	10	11	12	13	14	15	16	
UNITS	N/A	5	N/A	85	10	55	30	
AMOUNT	N/A	\$612.40	N/A	\$14,325.20	\$1,309.80	\$6,926.40	\$3,839.40	
	17	18	19	20	21	22	23	
UNITS	85	N/A	10	30	40	20	10	
AMOUNT	\$7,365.80	N/A	\$1,309.80	\$2,699.40	\$4,826.70	\$2,519.50	\$1,469.80	
	24	25	26	27	28	29	30	
UNITS	30	35	110	35	N/A	10	10	
AMOUNT	\$654.40	\$6,211.80	\$17,930.30	\$3,419.30	N/A	\$1,309.80	\$1,469.80	
	31							
UNITS	45							
AMOUNT	\$2,166.60							

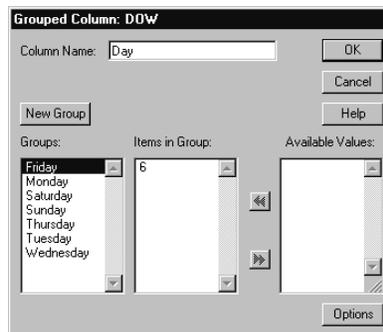
February 1999		Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
UNITS			1	2	3	4	5	6
AMOUNT			N/A	\$5,524.00	\$5,144.30	\$2,662.20	N/A	N/A
	7	8	9	10	11	12	13	

Purpose / Use: To build a report that displays daily information in a calendar-style format.

Sample File: Calendar.bqy

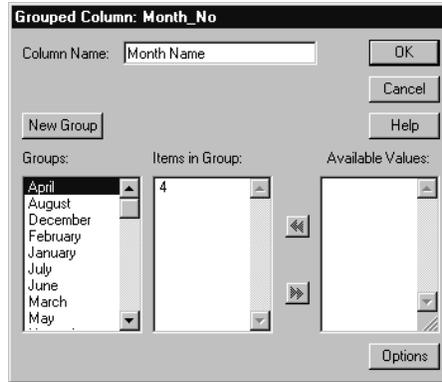
Steps to Produce: This tip requires some imaginative formatting of a pivot, which will then get embedded in the report. The key is to have a week number column that you use as a side label and put the days of the week in a top label. The day number is the fact for the pivot. The resulting pivot won't make sense until you drop it into a report section grouped by Year and Month.

- 1) You start with two tables. The first (Periods2000.xls in this example) has date, year number, month number, day number, ordinal day of the week (Sunday = 1 through Saturday = 7), and week number. The week number starts with a value of 1 for the first day of the year and then increments by 1 every Sunday. The second table (SALES.XLS in this example) has date, units, and dollar amount.
- 2) If a column for the name of the day doesn't already exist in the Periods2000 results, create a Grouping Column named "Day" to convert the ordinal day of the week into a day name.

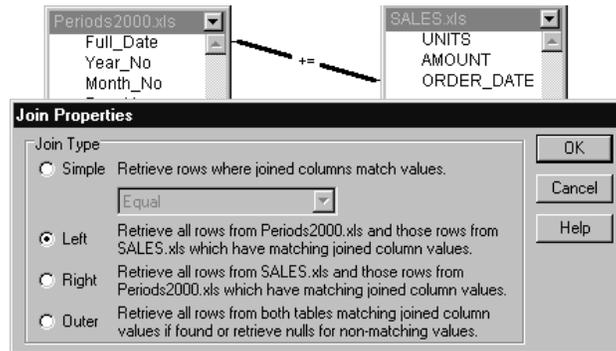


(Calendar-style Reports continued)

- 3) Create another Grouping Column called "Month Name" to create names for each month based on the "Month_No" column.



- 4) Now, create a local join query that links the Periods2000 data with the data to be analyzed. Include all of the Periods2000 columns in the Request line of the query. Double-click on the join line that links the Full Date in Periods2000 to the Order Date in the Sales data. Set the join Properties to display all rows in the Periods2000 data set and those Sales rows that have dates in Periods2000 (a left join in this example).



- 5) Once the local join query is processed, insert a new pivot section based on the Results of the local join query.
 - Place Week_No in the Side Labels
 - Put DOW and Day in the Top Labels
 - Place Day_No, Units and Amount in the Facts
 - Right-click somewhere in the pivot, select the Data Labels option from the speed menu and set it to "Down Side".
 - Right-click on the Day_No row and change the data function to be "Minimum".

Because of the left join, the pivot will show "N/A" on any date with no sales data. You should now see something similar to the figure shown below.

(Calendar-style Reports continued)

		1	2	3	4	5	6	7
		Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
1	Day_No						1	1
	UNITS						35	15
	AMOUNT						\$6,211.80	\$1,919.70
2	Day_No	2	3	4	5	6	7	8
	UNITS	85	70	N/A	25	95	130	125
	AMOUNT	\$12,348.30	\$4,573.60	N/A	\$3,737.00	\$10,370.60	\$15,774.90	\$9,995.00
3	Day_No	9	10	11	12	13	14	15
	UNITS	65	5	N/A	105	70	70	30
	AMOUNT	\$10,361.75	\$612.40	N/A	\$14,834.80	\$10,588.00	\$8,113.60	\$3,839.40
4	Day_No	16	17	18	19	20	21	22
	UNITS	100	55	75	55	40	50	135
	AMOUNT	\$9,570.50	\$6,884.30	\$8,661.00	\$4,536.40	\$4,826.70	\$4,824.00	\$10,829.80
5	Day_No	23	24	25	26	27	28	29
	UNITS	65	35	155	60	35	60	10
	AMOUNT	\$4,131.20	\$6,211.80	\$22,514.40	\$6,718.80	\$4,566.80	\$3,733.80	\$1,469.80
6	Day_No	30	1	1	2	3	4	5
	UNITS	90	25	75	110	35	N/A	N/A
	AMOUNT	\$5,370.40	\$3,062.00	\$6,603.50	\$12,157.80	\$5,471.80	N/A	N/A
7	Day_No	6	7	8	9	10	11	12

6) Next, format the pivot for calendar-style display in the report section.

- Select the Week_No label column, set both its text and fill colors to “Transparent”, its border type to “None”, and then make the column as thin as possible. This is to hide it from the final display.
- Select the DOW label row, set both its text and fill colors to “Transparent”, its border type to “None”, and then make the column as short as possible. This is to hide it from the final display also.
- Now select the Day label row. Set its text color to white and its fill color to blue. Set the font to be bold, the border style to “Horz and Vert”, and then center align the text.
- Select the three fact labels, set their fill colors to “Transparent”, their border style to “None” and right-align the text.
- Select just the Day_No fact label and set its text color to “Transparent”.
- Select the Day_No row and set its font style to bold.
- Select the Units and Amount fact rows, set the border style to “Vertical” and left-align the text.

You should now see a pivot that looks something like the next figure shown below.

(Calendar-style Reports continued)

	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
UNITS						35	15
AMOUNT						\$6,211.80	\$1,919.70
	2	3	4	5	6	7	8
UNITS	85	70	N/A	25	95	130	125
AMOUNT	\$12,348.30	\$4,573.60	N/A	\$3,737.00	\$10,370.60	\$15,774.90	\$9,995.00
	9	10	11	12	13	14	15
UNITS	65	5	N/A	105	70	70	30
AMOUNT	\$10,361.75	\$612.40	N/A	\$14,834.80	\$10,588.00	\$8,113.60	\$3,839.40
	16	17	18	19	20	21	22
UNITS	100	55	75	55	40	50	135
AMOUNT	\$9,570.50	\$6,884.30	\$8,661.00	\$4,536.40	\$4,826.70	\$4,824.00	\$10,829.80
	23	24	25	26	27	28	29
UNITS	65	35	155	60	35	60	10
AMOUNT	\$4,131.20	\$6,211.80	\$22,514.40	\$6,718.80	\$4,566.80	\$3,733.80	\$1,469.80
	30	1	1	2	3	4	5
UNITS	90	25	75	110	35	N/A	N/A
AMOUNT	\$5,370.40	\$3,062.00	\$6,603.50	\$12,157.80	\$5,471.80	N/A	N/A
	6	7	8	9	10	11	12
UNITS	40	35	165	N/A	150	100	35

- 7) Finally, create a Report section to hold the final calendar.
- Place the Year_No field in Report Group 1, and both Month_No and Month_Name into Report Group 2.
 - Delete the Month_No field that appears in the group 2 header, leaving only the Month Name. Move the Month Name to the top of the Group 2 header.
 - Select the Year_No item in the Group 1 header, copy it to the clipboard and then paste it into the Group 2 header. Move it to be underneath the Month Name, right-click on it. Select "Properties" and change its Number format to be "###0". Delete Year_No from the Group 1 header.
 - Select the Table object in the Report's body and delete it.
 - Select the Group 2 header, right-click and select "Keep Together" from the speed menu. Then drag the formatted pivot into the Group 2 header area. Adjust the pivot's location within the Group 2 header as needed but *do not* change the vertical size of the Group 2 header after inserting the pivot.
 - You will notice that each month's calendar is missing the bottom line. To add a floating bottom line, drag a Horizontal Line object into the *body* area of the report. Size the line to be the same width as the pivot. Use the cursor keys to place the line at the extreme bottom of the pivot. (You may want to Zoom the report to check for exact placement of the line.)
 - Click on the horizontal line, then shift-click on the pivot in the same body section. With both the line and the pivot selected, go to the main menu and use the Format | Spring Objects menu item. This will set the line so that it will automatically float with the pivot as the pivot changes height for each month.