Actuate One™

One Design
One Server
One User Experience

Using BIRT Studio - iServer Edition

Information in this document is subject to change without notice. Examples provided are fictitious. No part of this document may be reproduced or transmitted in any form, or by any means, electronic or mechanical, for any purpose, in whole or in part, without the express written permission of Actuate Corporation.

© 1995 - 2011 by Actuate Corporation. All rights reserved. Printed in the United States of America.

Contains information proprietary to:
Actuate Corporation, 2207 Bridgepointe Parkway, San Mateo, CA 94404

www.actuate.com
www.birt-exchange.com

The software described in this manual is provided by Actuate Corporation under an Actuate License agreement. The software may be used only in accordance with the terms of the agreement. Actuate software products are protected by U.S. and International patents and patents pending. For a current list of patents, please see http://www.actuate.com/patents.

Actuate Corporation trademarks and registered trademarks include:
Actuate, ActuateOne, the Actuate logo, Archived Data Analytics, BIRT, Collaborative Reporting Architecture, e.Analysis, e.Report, e.Reporting, e.Spreadsheet, Encyclopedia, Interactive Viewing, OnPerformance, Performancesoft, Performancesoft Track, Performancesoft Views, Report Encyclopedia, Reportlet, The people behind BIRT, X2BIRT, and XML reports.

Actuate products may contain third-party products or technologies. Third-party trademarks or registered trademarks of their respective owners, companies, or organizations include:

Adobe Systems Incorporated: Flash Player. Apache Software Foundation (www.apache.org): Axis, Axis2, Batik, Batik SVG library, Commons Command Line Interface (CLI), Commons Codec, Derby, Shindig, Struts, Tomcat, Xerces, Xerces2 Java Parser, and Xerces-C++ XML Parser. Bits Per Second, Ltd. and Graphics Server Technologies, L.P.: Graphics Server. Bruno Lowagie and Paulo Soares: iText, licensed under the Mozilla Public License (MPL). Castor (www.castor.org), ExoLab Project (www.exolab.org), and Intalio, Inc. (www.intalio.org): Castor. Codejock Software: Xtreme Toolkit Pro. DataDirect Technologies Corporation: DataDirect JDBC, DataDirect ODBC. Eclipse Foundation, Inc. (www.eclipse.org): Babel, Data Tools Platform (DTP) ODA, Eclipse SDK, Graphics Editor Framework (GEF), Eclipse Modeling Framework (EMF), and Eclipse Web Tools Platform (WTP), licensed under the Eclipse Public License (EPL). Jason Hsueth and Kenton Varda (code.google.com): Protocole Buffer. ImageMagick Studio LLC.: ImageMagick. InfoSoft Global (P) Ltd.: FusionCharts, FusionMaps, FusionWidgets, PowerCharts. Mark Adler and Jean-loup Gailly (www.zlib.net): zLib. Matt Ingenthron, Eric D. Lambert, and Dustin Sallings (code.google.com): Spymemcached, licensed under the MIT OSI License. International Components for Unicode (ICU): ICU library. KL Group, Inc.: XRT Graph, licensed under XRT for Motif Binary License Agreement. LEAD Technologies, Inc.: LEADTOOLS. Microsoft Corporation (Microsoft Developer Network): CompoundDocument Library. Mozilla: Mozilla XML Parser, licensed under the Mozilla Public License (MPL). MySQL Americas, Inc.: MySQL Connector. Netscape Communications Corporation, Inc.: Rhino, licensed under the Netscape Public License (NPL). Oracle Corporation: Berkeley DB. PostgreSQL Global Development Group: pgAdmin, PostgreSQL, PostgreSQL JDBC driver. Rogue Wave Software, Inc.: Rogue Wave Library SourcePro Core, tools.h++. Sam Stephenson (prototype.conio.net): prototype.js, licensed under the MIT license. Sencha Inc.: Ext JS. Sun Microsystems, Inc.: JAXB, JDK, Jstl. ThimbleWare, Inc.: JMemcached, licensed under the Apache Public License (APL). World Wide Web Consortium (W3C)(MIT, ERCIM, Keio): Flute, JTidy, Simple API for CSS. XFree86 Project, Inc.: (www.xfree86.org): xvfb. Yuri Kanivets (code.google.com): Android Wheel gadget, licensed under the Apache Public License (APL). ZXing authors (code.google.com): ZXing, licensed under the Apache Public License (APL).

All other brand or product names are trademarks or registered trademarks of their respective owners, companies, or organizations.

Document No. 110812-2-761302 August 11, 2011

Contents

Chapter 9

About Using BIRT Studio - iServer Edition

Using BIRT Studio - iServer Edition provides information about using and configuring the BIRT Studio report design tool. This manual explains how to design a report and how to configure and customize BIRT Studio.

- *About Using BIRT Studio - iServer Edition.* This chapter provides an overview of this guide.
- *Part 1. Designing reports using BIRT Studio.* This part describes the tasks that users complete to design reports.
- *Chapter 1. Getting started.* This chapter introduces BIRT Studio and describes how to create, save, and run a report. The chapter also describes how to enable and disable default hyperlinks in a report design, and create bookmarks to report elements such as tables, charts, cross tabs, and summary tables.
- *Chapter 2. Editing and formatting report content.* This chapter describes the formatting options in BIRT Studio, including applying a theme, changing font properties, changing the display of data, formatting data based on conditions, and modifying the page layout. The chapter also describes how to merge several columns into a single column, and work with data in merged columns.
- *Chapter 3. Inserting calculated data.* This chapter describes how to write expressions to calculate data.
- *Chapter 4. Organizing data in a report.* This chapter describes how to sort, group, and aggregate data.
- *Chapter 5. Filtering data.* This chapter describes how to create filters to display only the information that your report needs. The chapter also explains how to prompt for filter values at run time by creating regular and dynamic filter parameters.
- *Chapter 6. Presenting data in a cross tab.* This chapter describes how to create a cross tab, obtain data for a cross tab, and how to format and analyze data in a cross tab.

- *Chapter 7. Working with summary tables.* This chapter describes how to create and modify summary tables, as well as how to change table types.
- *Chapter 8. Presenting data in a chart.* This chapter describes the types of charts you can create, and discusses the procedures for displaying data in a chart. This chapter also describes how to create and format Flash charts.
- *Chapter 9. Functions and operators.* This chapter is a reference to all the supported functions and operators you can use to create calculated data. The chapter describes the functions you can use to create aggregate calculations, and the operators you can use in expressions, conditional formatting, as well as in filter conditions.
- *Chapter 10. Using data from multiple information objects.* This chapter explains how to join information objects to access data from multiple data sources.
- *Part 2. Customizing and Integrating BIRT Studio.* This part is a guide to customizing and integrating BIRT Studio.
- *Chapter 11. Introduction to BIRT Studio customization.* This chapter introduces BIRT Studio, provides an overview of the types of customizations, and lists the Actuate software required to perform the customization tasks.
- *Chapter 12. Creating and publishing report templates.* This chapter provides guidelines for designing effective report templates for use with BIRT Studio. The chapter also describes the procedures for publishing the templates and their associated resources.
- *Chapter 13. Providing data.* This chapter describes the different ways to provide BIRT Studio users with data for their reports. The chapter provides guidelines for designing information objects and developing custom data sources.
- *Chapter 14. Configuring BIRT Studio.* This chapter discusses how to modify the appearance of BIRT Studio pages, how to control the functionality available to users, how to customize BIRT Studio for different user groups, and how to configure the BIRT Studio application.
- *Chapter 15. Actuate BIRT Studio URIs.* This chapter discusses how to access BIRT Studio using a URI and how to use the BIRT Studio servlet.

Part One

Designing reports using BIRT Studio

Chapter

1

Getting started

This chapter contains the following topics:

- About BIRT Studio
- Starting BIRT Studio
- Creating a report
- Saving a report
- Running and viewing a report
- Working with default hyperlinks
- Working with bookmarks

About BIRT Studio

BIRT Studio is a web-based report design tool for users who want to create professional-looking reports quickly and easily without detailed understanding of database architecture or report design techniques. You use predefined data sources and templates that provide the data and basic layout for your reports. Then, using BIRT Studio's intuitive graphical interface, you can add a wide variety of charts; rearrange and reformat content; sort, group, and filter data. As you edit and format a report, a preview feature provides an instant view of your changes with sample data.

BIRT Studio is a highly customizable application. A system or application administrator creates and integrates custom data sources and templates that you use to create reports. An administrator can also customize the user interface and functionality in BIRT Studio to tailor the tool to your organization's report design requirements.

This guide provides instructions for using the features included in the default installation of BIRT Studio. The functionality available to you can differ depending on how your system administrator configures and customizes BIRT Studio.

Starting BIRT Studio

The configuration of your organization's application system determines how you access BIRT Studio. Your system administrator should provide a URL and, in most cases, login information. As BIRT Studio is an option in Actuate BIRT iServer or Actuate Information Console, you access BIRT Studio by logging in to either iServer or Information Console. The URL you use to access iServer Express or Information Console and the login credentials you supply are items you need to get from your system administrator.

After you log in, on the Documents page that appears, choose the BIRT Studio link, as shown in Figure 1-1.

BIRT Studio opens in your web browser, as shown in Figure 1-2.

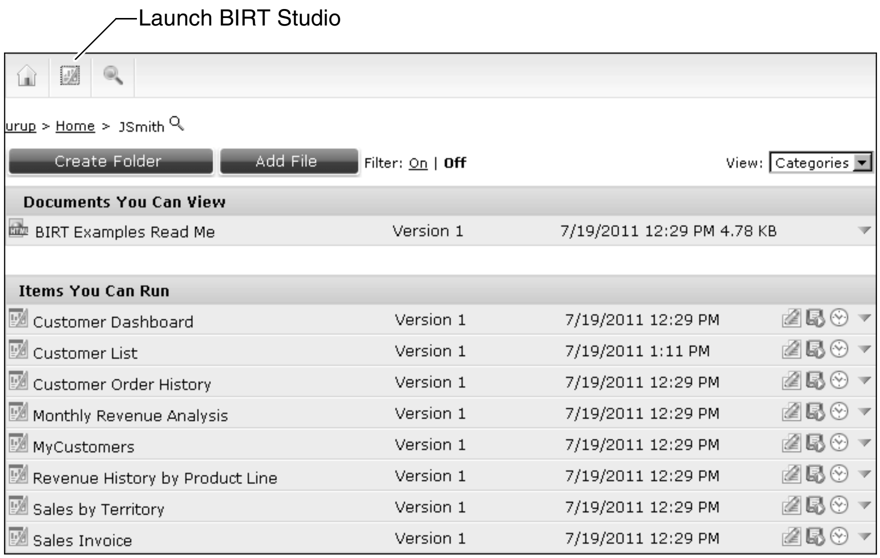


Figure 1-1 Launching BIRT Studio from an application

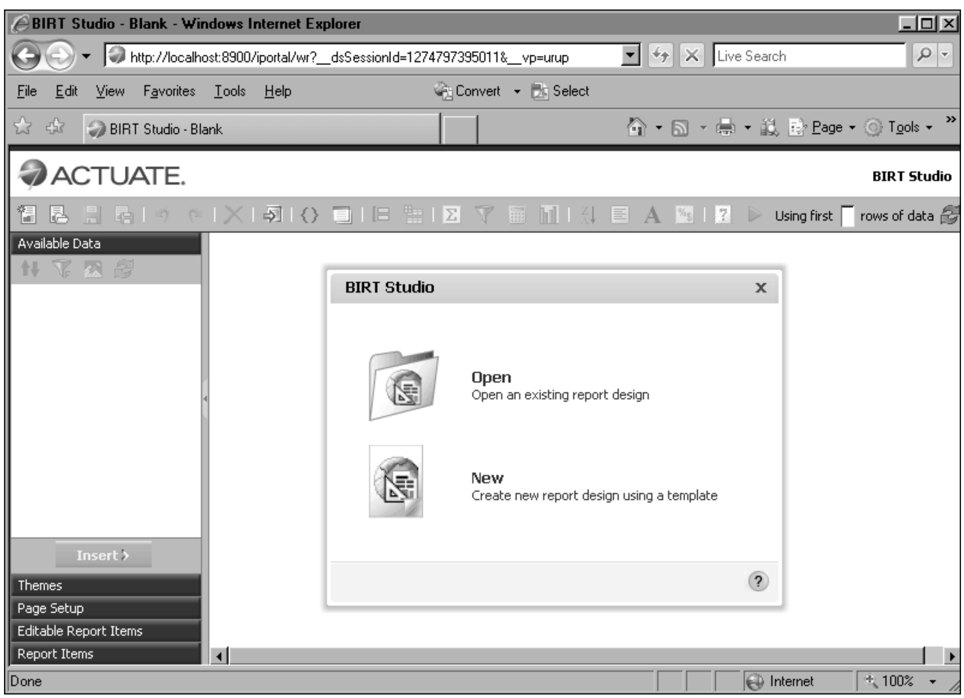


Figure 1-2 Opening an existing report or creating a new one in BIRT Studio

Creating a report

After you start BIRT Studio, you can either create a new report or open an existing report. When you choose to create a report, BIRT Studio displays a series of dialog boxes to guide you through the following required steps:

- Select a template on which to base your report.
- Select a data source that provides the data for your report.

After you select a template and data source, you perform the following tasks:

- Select and insert the data in the report.
- Edit and format the report.
- Save, run, and view the report.

Selecting a template

Every report you create in BIRT Studio is based on a template. A template determines basic report structure and typically contains the following elements:

- Report items, such as a report title
- A report table to display data
- Page footer items, such as a date and time stamp or a page number

Figure 1-3 shows two templates provided in the default BIRT Studio installation.

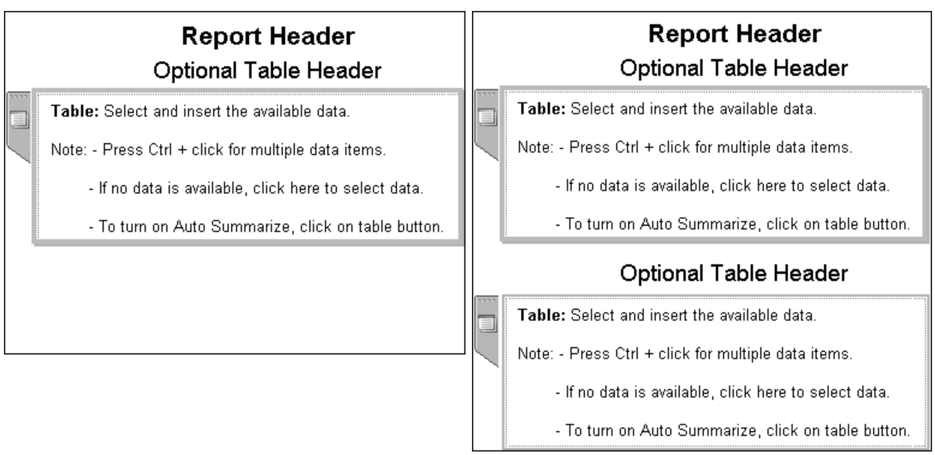


Figure 1-3 Two templates included in the default installation of BIRT Studio

Figure 1-4 identifies the items a template typically includes. All templates contain at least one empty table as a placeholder for data. Notice that the table contains instructions for inserting data. Typically, a BIRT Studio template provides tips or information about using items.

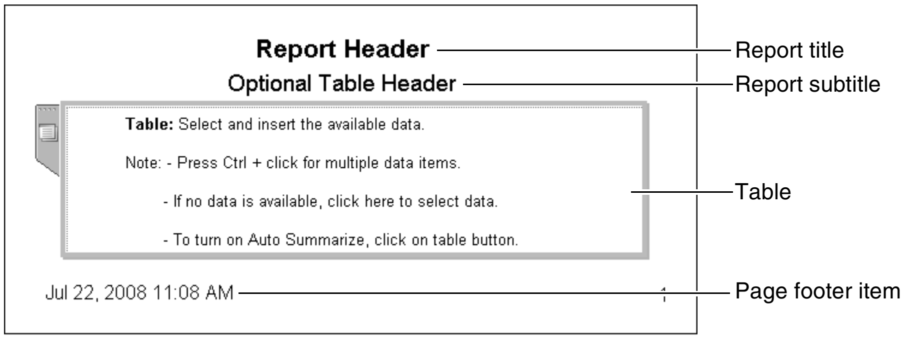


Figure 1-4 Items in a typical template

Your BIRT Studio installation may provide custom templates designed for the types of reports you need to create. BIRT Studio organizes templates in categories, a convenient way to group similar templates. Templates can be organized by report type or by department, for example. Templates may be stored in a single category, depending on the requirements at your site and the configuration your system administrator implements. Your access privileges determine which categories and templates you can view and use.

How to select a template for your report

1 When you create a new report, BIRT Studio displays Report Template. In Report Template, select a template category from the drop-down list.

Report Template displays thumbnail images of the templates in the selected category. Figure 1-5 shows sample templates in the Standard category included in the default installation of BIRT Studio.

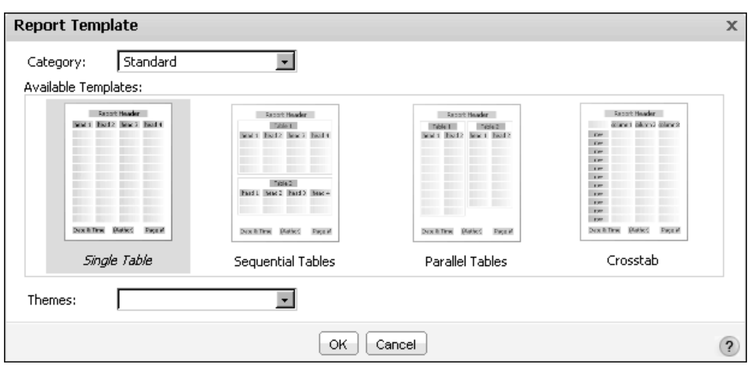


Figure 1-5 Selecting a report template

2 To view a description of the template, move your cursor over the thumbnail image of the template.

3 Select the template to use, then choose OK.

Selecting a data source

After you select a template, you select a data source from a list of available sources. Depending on your system configuration, the data source you use can be one of the following:

- A data set included in the selected template
- An information object
- A data object

These data sources contain all the information to connect to a data repository, such as a database, and retrieve a specific set of data.

About data sources

Data sets present in information objects, and data objects classify data fields retrieved from a data source, in dimensions, measures, and attributes. These categories are specified by the information object developer, or the data object developer at the time of designing the information object, or data object, and cannot be modified using BIRT Studio. It is useful to understand these categories to use them appropriately in a report design.

Dimensions are data fields that group other data fields. Each dimension field can contain multiple attributes. These attributes are typically associated with a dimension. Table 1-1 lists the possible attributes for each dimension field.

Table 1-1 Dimensions with associated attribute fields

Dimension	Attribute fields
Product Line	Product name, product code
Region	Country, state, city

Measures are data fields that can be aggregated, or computed such as revenue, profit, price, and so on.

Data sets included in templates, information objects, as well as data objects provide easy access to the data fields that contain the data for a report. For example, an information object or a data set named CustomerData can contain customer information fields, such as customer name, address, phone number, and so on. You choose this data source to create a report that lists customers and their contact information.

You use an information object to create a report table, a summary table or a chart. You use a data object to create a report table, a chart, or a summary table. If the data object contains one or more cubes, you can use it to create a cross tab. BIRT Studio provides a convenient Table Builder wizard that enables you select data fields from information objects or data objects and insert them into a report design.

A data object is a collection of data sets and cubes that can be used to create BIRT reports. A data object contains data sets and cube items, specified by the data object developer. A data object must contain at least one data set but does not need to contain a cube.

A data object provides access to predesigned data sources, data sets, and sometimes cubes. You use Actuate BIRT Designer Professional to create data objects. Report developers create data objects to streamline the report creation process. Data objects provide the following benefits:

- Simplified data access and retrieval. The predesigned data sources, data sets, and cubes in a data object enable report developers to select the data to use in a report without requiring knowledge of the underlying data source, how to connect to it, and how to extract data from it.
- Reusability across multiple reports. If a suite of reports require the same data, designing the data sources, data sets, and cubes once in a shared data object eliminates the need to design the same elements repeatedly for each report.
- Dynamic updates to data items. Changes to data items in a data object propagate to reports that use the data object.

A report accesses data from a data object using either a data object design (.datadesign) file or a data object store (.data). The data design file retrieves data, on demand, each time the report is run. A data object store contains cached, or materialized, data, and provides more efficient access to data. If getting real-time data is more important than report generation speed, use the data object design file. If data in the underlying data source does not change constantly, or if a data object store is generated regularly, use the data object store.

You can limit the data available to a report design by applying filters to a data set in a template, an information object, or a data object. Both information objects and data objects often contain very large amounts of data. Using filters to limit the data available to a report design is useful when you need to create a report that displays specific data quickly. For example, if the data set displays sales totals for the years 2003, 2004, and 2005, and you need to view the sales totals for the most recent of those years, you can specify a filter condition such that BIRT Studio retrieves and displays data only for the year 2005 in the report design.

Using a data source

The following section describes how to select a data source. The procedure for selecting the data source for your report depends on the template you select. If you select a template that does not include a data set, BIRT Studio prompts you to select an information object or a data object stored in the iServer Encyclopedia volume. If you choose a cross-tab template, you must select a data object that contains a cube. For more information about creating a cross tab, see Chapter 6, "Presenting data in a cross tab."

How to select a template that includes one or more data sets

1 In Report Template, select a template that includes one or more data sets. Choose OK. Data appears, as shown in Figure 1-6.

Data displays the data sets included in the template.

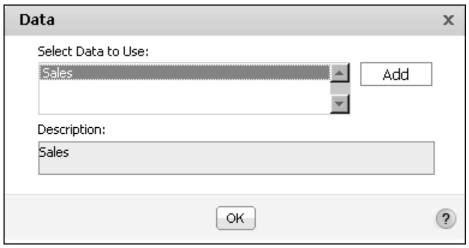


Figure 1-6 Selecting a data set

2 Select a data set, then choose OK.

Table Builder appears listing the data fields in the selected data set. For more information about using Table Builder, see "How to select and insert data fields in a report design using Table Builder," later in this chapter.

How to select an information object

1 In Report Template, select a template that does not contain a data set. Choose OK. Data Source appears.

2 In Data Source, select Information Objects as shown in Figure 1-7, then choose OK.

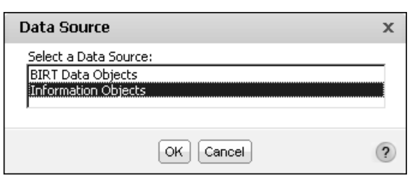


Figure 1-7 Selecting an information object data source

3 Select Information Object appears. Navigate to the folder that contains the information object to use. An information object has a .iob file-name extension. Select the information object, as shown in Figure 1-8, then choose OK.

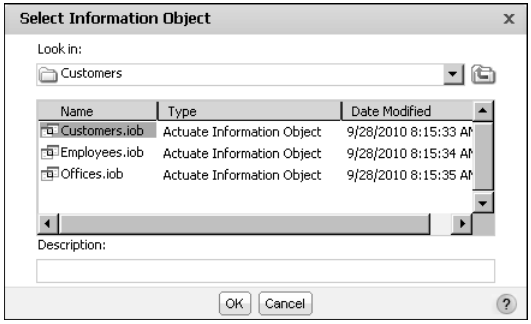


Figure 1-8 Selecting an information object

Table Builder appears listing the data fields in the selected data set. For more information about using Table Builder, see "How to select and insert data fields in a report design using Table Builder," later in this chapter.

How to select a data object

1 In Report Template, select a template that contains no data set. Choose OK.

2 In Data Source, shown in Figure 1-9, select BIRT Data Objects, then choose OK.

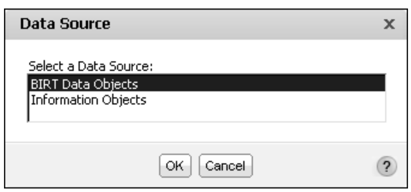


Figure 1-9 Selecting a data object

Select Data appears as shown in Figure 1-10.

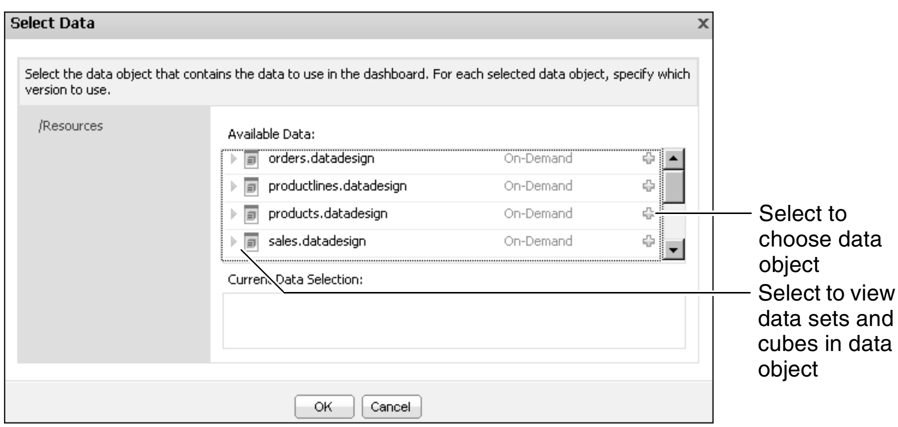


Figure 1-10 Selecting a data object

3 In Available Data, select a data object design (.datadesign) file to retrieve data on-demand, or a data object store (.data) file to use cached data.

- To view the data sets and cubes in each data object, select the arrow next to the data object file name, as shown in Figure 1-10.

  Available Data displays a list of data sets and cubes, in the selected data object, as shown in Figure 1-11.

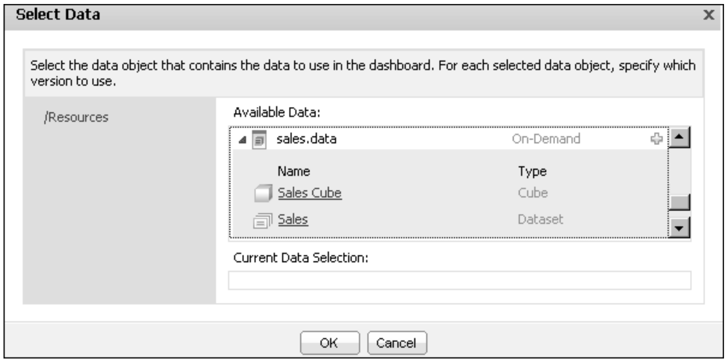


Figure 1-11 Data set and cube in the selected data object

- Select the cube to view the data fields it contains. Figure 1-12 shows the data fields in Sales Cube. Choose OK.

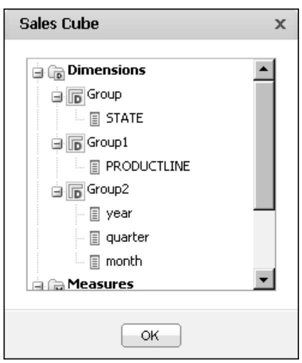


Figure 1-12 Data fields in the cube

- Select the data set to view the data fields it contains. Figure 1-13 displays the data fields in Sales.

  Choose OK.

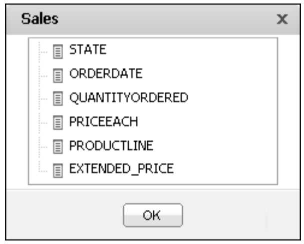


Figure 1-13 Data fields in the Sales data set

4 On Select Data, in Available Data choose '+' to select the data object design file, or data object store file as shown in Figure 1-14.

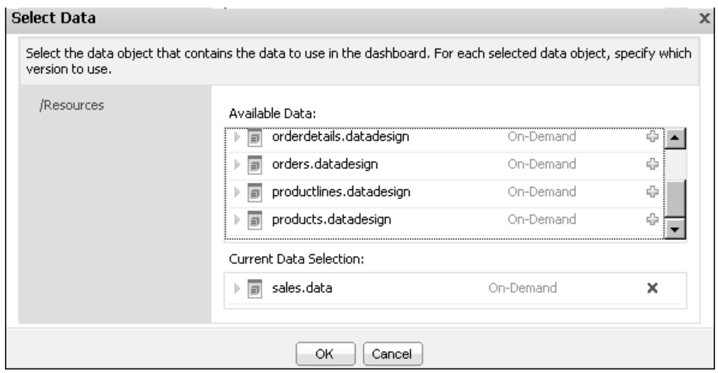


Figure 1-14 Current Data Selection displaying the chosen data object

The selected file appears in Current Data selection, as shown in Figure 1-14. To change your selection, in Current Data Selection, select Delete. Then, select a different data object in Available Data. Choose OK.

Select Data Set appears, displaying the data sets in the selected data object, as shown in Figure 1-15.

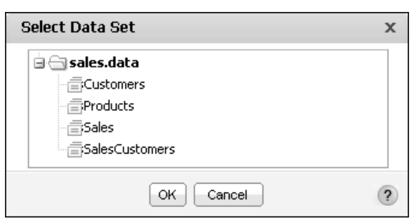


Figure 1-15 Selecting a data set from the data object

Select a data set to use, then choose OK.

Table Builder appears listing the data fields in the selected data set. For more information about using Table Builder, see "How to select and insert data fields in a report design using Table Builder," later in this chapter.

Selecting and inserting data in a report

Once you have chosen a data source, the next step is to select the data fields, and lay them out in the report design. You can do one of the following:

- Use Table Builder to select and insert the data fields in the report design. You can also limit the data retrieved from the data set using Table Builder.
- To preview the data as you select and insert data fields in the report design, choose Cancel on Table Builder, without making any selections. All the data fields from the selected data set appear in Available Data in BIRT Studio. Follow the procedure described in "How to manually select and insert data fields in a report table," later in this chapter.

How to select and insert data fields in a report design using Table Builder

On Table Builder, shown in Figure 1-16, complete the following steps:

1 In Data, in Use Data From, select an information object, or data object, if necessary.

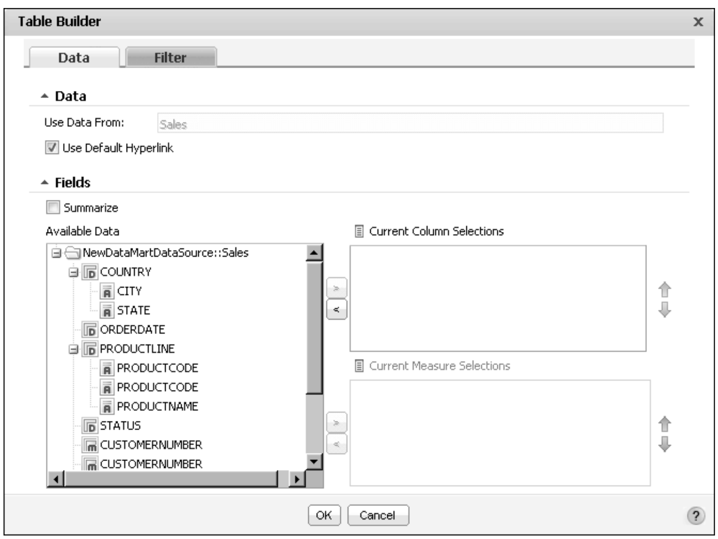


Figure 1-16 Table Builder displaying data fields in the selected data object

2 To use default hyperlinks present in the selected data set, select Use Default Hyperlink, if necessary. For more information about using default hyperlinks, see "Working with default hyperlinks," later in this chapter.

3 In Fields, do one of the following:

- To create a summary table, select Summarize. For more information about creating a summary table see Chapter 7, "Working with summary tables."
- To create a detail report, deselect Summarize.

4 In Available Data, press Ctrl, and select each data field to use from the list of available data fields. Then select the right arrow. The selected data fields appear in Current Column Selections, as shown in Figure 1-17.

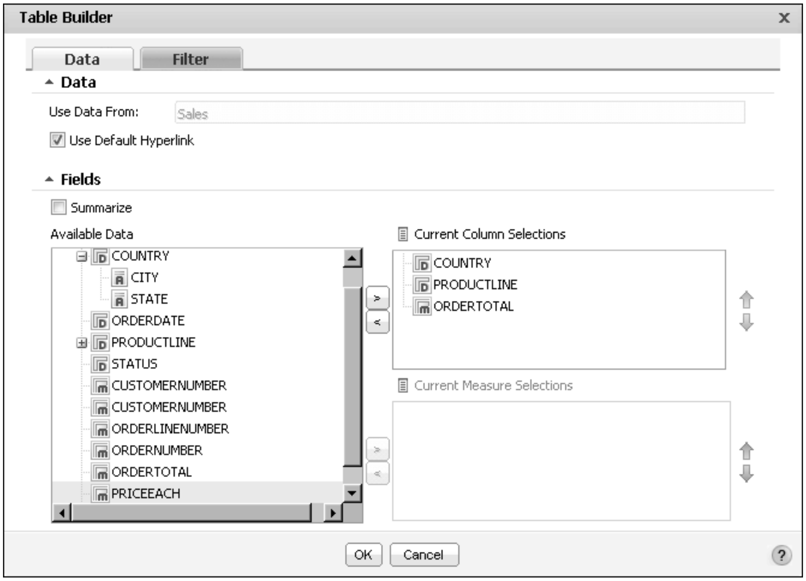


Figure 1-17 Current Column Selections displaying the selected data fields

Use the up and down arrows to rearrange the order in which the data fields are displayed in the report design. You can now specify a filter condition if you want to limit the data retrieved from the data source and displayed in the report.

How to limit the data displayed in a report

You can use Table Builder—Filter to specify a filter at the data-set level. The following example sets a filter condition such that only data for the country USA is retrieved from the data set.

1 In Table Builder, select Filter.

2 In Filter By, select the column to evaluate.

3 In Condition, select a comparison operator from the drop-down list.

Based on your selection, either one or two fields appear in Value.

4 In Value, do one of the following:

- Type a value.
- Choose Select Values, then select a value from the list of values.

5 Choose Add Condition. The condition appears in Filters, as shown in Figure 1-18. Choose Validate. After a message appears confirming validity of the syntax of the filter condition, choose OK.

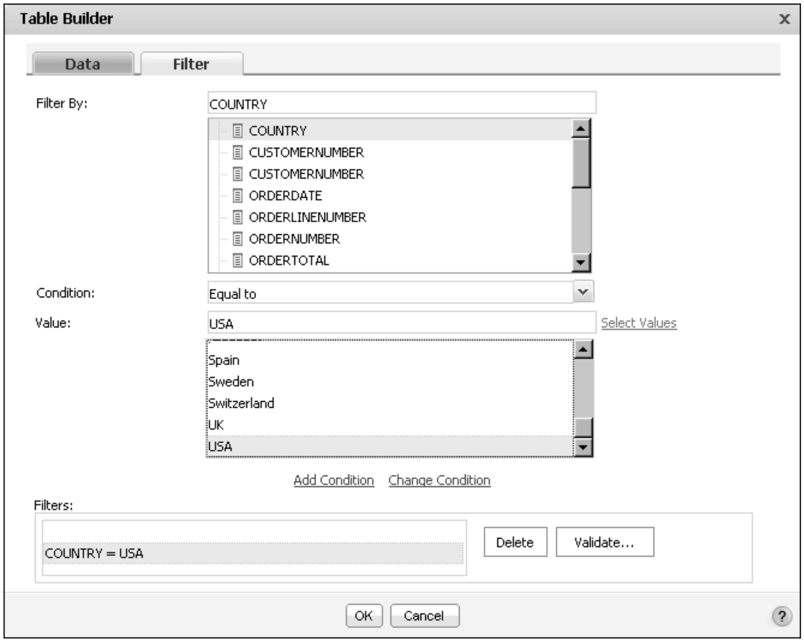


Figure 1-18 Specifying a data-set filter condition

The selected data fields appear in the report design, as shown in Figure 1-19.

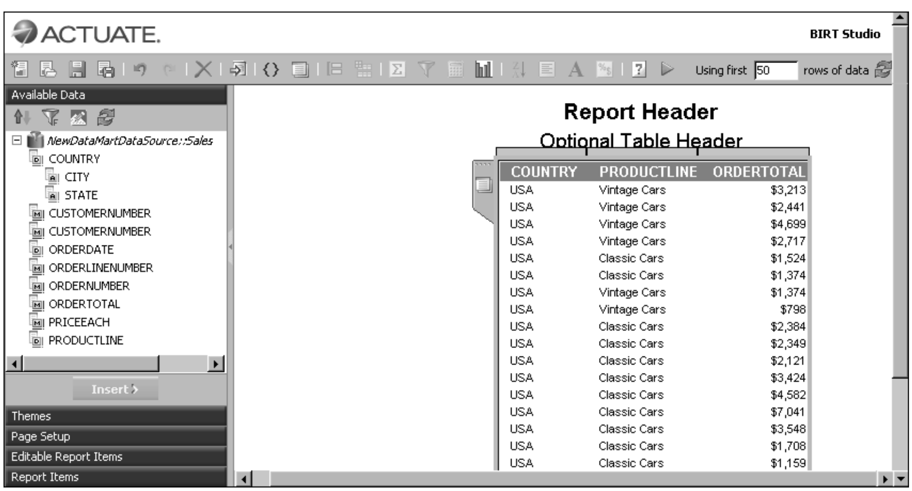


Figure 1-19 BIRT Studio displaying the report design

You are now ready to format the report design. For more information on how to format a report, see "Editing and formatting a report," later in this chapter.

How to manually select and insert data fields in a report table

In BIRT Studio, press Ctrl, and in Available Data, select each data field to insert in the table, then choose Insert. Alternatively, drag the data fields and drop them in the table. The order in which you select the data fields is important because they appear in the report in the order of selection. Figure 1-20 shows an example of three data fields, Country, Product Line, and Order Total, inserted in the report table.

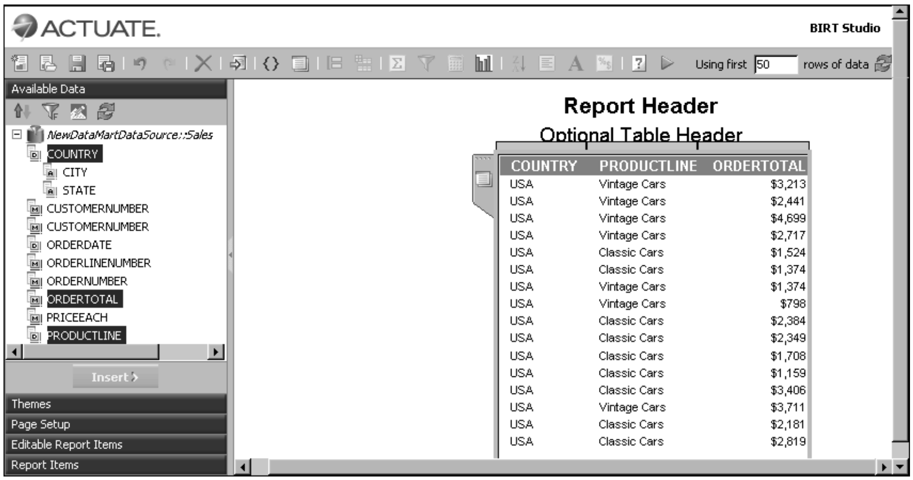


Figure 1-20 BIRT Studio displaying the selected data

For each data field you insert, a column appears in the table, displaying up to 50 values. Notice that the data field names, COUNTRY, PRODUCTLINE, and ORDERTOTAL appear at the top of each column. These items are called column headers, and BIRT Studio creates them when you insert data fields in a table.

How to use a new data source after previewing data in a report design

After previewing the data displayed in the report design, if you want to select a different data source, do one of the following:

- Choose New to select a different data set included in a template, or information object data source. Then, create the report design over again.

- Choose Manage Data to select a different data object data source. Manage Data appears. Complete the following steps:
  1. In Available Data, select a new data object to use. The data object you selected appears in Current Data Selection, as shown in Figure 1-21. Choose OK.

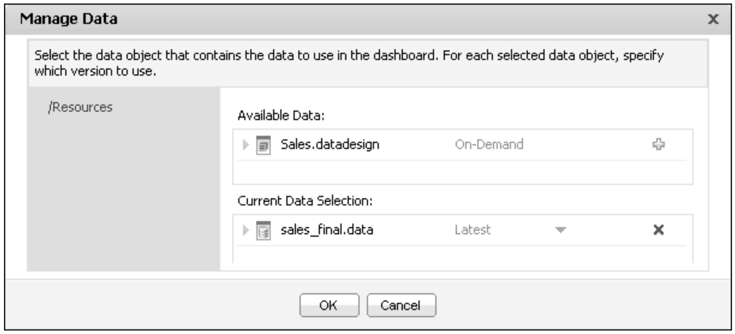


Figure 1-21 Selecting a new data object

2 In BIRT Studio, choose Modify in the left navigation pane, as shown in Figure 1-22.

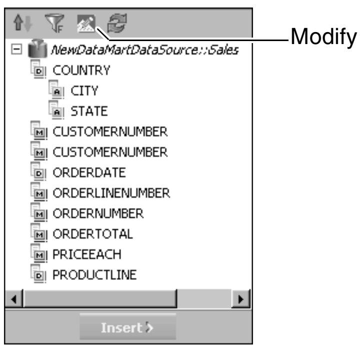


Figure 1-22 Choose Modify

Select Data Set appears displaying the data set currently in use, and the data sets in the new data object you selected in step 1.

3 Select a new data set from the list of available data sets. Then choose OK.

Table Builder appears displaying the data fields in the data set you selected.

4 To add data to the report design, repeat "How to select and insert data fields in a report design using Table Builder," earlier in this chapter.

Editing and formatting a report

So far, you have learned how to create a simple listing report. The next step is to enhance the report design. For example, the report shown in Figure 1-19 needs a real title, and the names used in the column headers need editing. If you do not like the fonts and colors in the template, you can change them. Editing and formatting tasks are described later in this document.

As you edit and format a report, the report design provides a preview of your changes. The report design, however, does not show all rows of data. By default, the report design shows the first 50 rows. To see more rows of data, change the number in Using first 50 rows of data, then press Enter. You can set a number up to 200.

Saving a report

You save a report to a folder in an iServer Encyclopedia volume. Your access privileges determine the folders in which you can save a report.

The first time you save a report, BIRT Studio prompts you to select the folder and provide a name and description for the report. The next time you save the report, BIRT Studio saves the report to the same folder and you can choose whether to replace the original report or create a new version.

How to save a new report

1 Choose Save. The first time you save a report, Save Report Design appears, as shown in Figure 1-23. This dialog box typically displays your home folder.

2 If necessary, navigate through the folder structure to select a different folder.

3 In File Name, type a name for the report. The file name can contain alphanumeric characters and spaces, but it cannot contain punctuation.

4 Optionally, in Description, type a brief description of the report. A description is helpful if you are sharing this report with other users.

   Save as type shows that the report is saved as a BIRT Report Design file with a .rptdesign file-name extension. Choose OK.

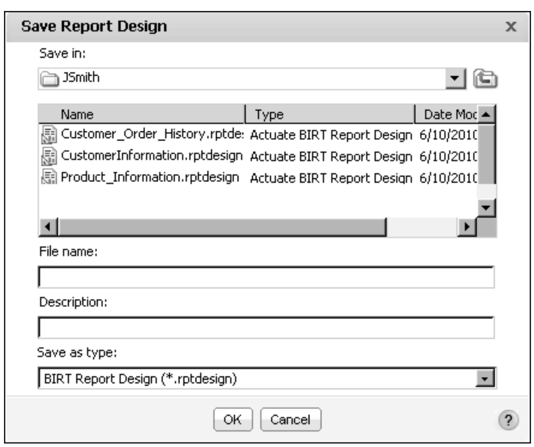


Figure 1-23 Save Report Design

How to save a report and change its name or location

Sometimes you need to create a new version of a report while retaining the original report or you want to save the report in a different folder. To accomplish either task, choose Save As. In Save Report Design, select a folder, and type a file name in the same way as when using Save.

Running and viewing a report

As you design a report, the data you see In BIRT Studio is just a preview. To view all the data, save the report, then run the report. You run a report by choosing Save and View on the toolbar. The generated report appears in Actuate BIRT Viewer in your web browser. Figure 1-24 shows the viewer displaying the finished version of the report design shown in Figure 1-20.

In the viewer, you can page through the report, print the report, export the report to another format, and even modify the content, organization, and formatting of the report. For more information about all the report operations you can perform in the viewer, see *Working with Actuate BIRT Viewers.*

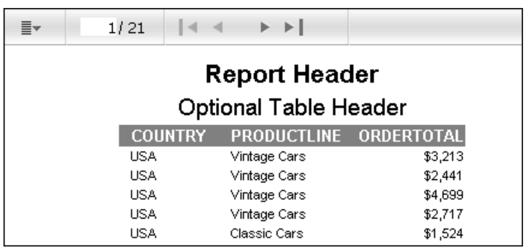


Figure 1-24 Actuate BIRT Viewer displaying a report

Working with default hyperlinks

In BIRT Studio, when you use a data object data source that contains data fields with links to additional information such as sub-reports, charts, and so on, a report developer can either allow the user to use the existing hyperlinks in the report, or prevent the user from accessing the linked data. This feature is useful when you need to share the linked content with a certain group of users, but not with others.

How to enable default hyperlinks in a BIRT report design

To activate hyperlinks for all the inserted columns in a report design that contain linked content, complete the following steps:

1 Create a new report design, and select the data object design, or data object store file to use. The data fields appear in Available Data, in Table Builder.

2 Right-click the table handle, then choose Enable Default Hyperlink from the menu that appears, as shown in Figure 1-25.

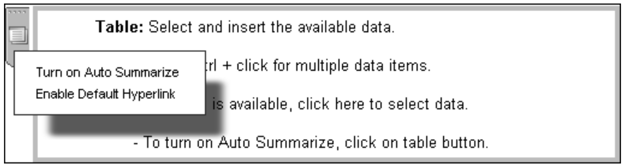


Figure 1-25 Selecting Enable Default Hyperlink from the table menu

3 Select one or more data fields containing hyperlinks, and choose Insert. The data fields appear in the BIRT report design, displaying any predefined hyperlinks as shown in Figure 1-26. Choose each link to view the referenced content.

PRODUCTLINE	PRODUCTVENDOR	QUANTITYINSTOCK
Motorcycles	Min Lin Diecast	7933
Classic Cars	Classic Metal Creations	7305
Motorcycles	Highway 66 Mini Classics	6625
Motorcycles	Red Start Diecast	5582
Classic Cars	Motor City Art Classics	3252
Classic Cars	Second Gear Diecast	6791
Classic Cars	Autoart Studio Design	68
Classic Cars	Second Gear Diecast	3619

Figure 1-26 Report design displaying default data field hyperlinks

How to enable default hyperlinks for some columns in a BIRT report design

You can also enable default hyperlinks for some data fields containing linked content, and disable the option for other data fields containing hyperlinks.

1 Create a new report design, and select the data object design, or data object store file to use. The data fields appear in Available Data.

2 Right-click the table handle, and choose Enable Default Hyperlink from the menu that appears.

3 Select the data fields for which you want to enable the user to use associated hyperlinks, then choose Insert.

The data fields appear in the BIRT report design displaying any predefined hyperlinks. Choose each link to view the referenced content.

4 Right-click the table handle, and choose Disable Default Hyperlink from the menu that appears, as shown in Figure 1-27.

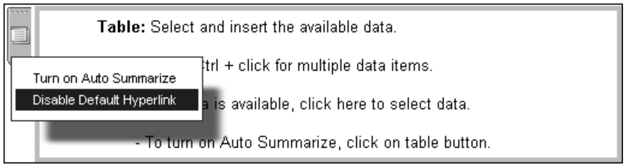


Figure 1-27 Selecting Disable Default Hyperlink from the table menu

5 Select the data fields for which you want to disable associated hyperlinks, then choose Insert.

The data fields appear in the BIRT report design as regular data columns without displaying hyperlinks, as shown in Figure 1-28.

PRODUCTLINE	PRODUCTVENDOR	QUANTITYINSTOCK
Motorcycles	Min Lin Diecast	7933
Classic Cars	Classic Metal Creations	7305
Motorcycles	Highway 66 Mini Classics	6625
Motorcycles	Red Start Diecast	5582
Classic Cars	Motor City Art Classics	3252
Classic Cars	Second Gear Diecast	6791
Classic Cars	Autoart Studio Design	68
Classic Cars	Second Gear Diecast	3619

Figure 1-28 Report design displaying disabled data field hyperlinks

Working with bookmarks

A bookmark is a marker that defines a place in a report. You can use BIRT Studio to create a bookmark to a report table, a regular chart or Flash chart in a report table or section, a summary table, or a cross tab.

When you define a bookmark, you specify a name that is used by BIRT Dashboards to identify the bookmarked content. To create a bookmark that is usable in BIRT Dashboards, the bookmark name must begin with a letter ([A-Za-z]) and may be followed by any number of letters, digits ([0-9]), hyphens ("-"), underscores ("_"), colons (":"), and periods ("."). Do not enclose the name in single or double quotation marks, or use spaces in the name. The letters in the name are case-sensitive.

How to create a bookmark

1 Select the target report element for which you want to define a bookmark, such as the report table, or chart element, and choose Bookmark from the context menu.

2 Bookmark appears, as shown in Figure 1-29. In Enter Bookmark, type a name for the bookmark. Choose OK.

Figure 1-29 Creating a bookmark

To edit a defined bookmark, select the element containing a bookmark, and choose Bookmark from the context menu. In Bookmark, edit the name of the element in Enter Bookmark.

To delete a defined bookmark, in Bookmark, delete the name of the element in Enter Bookmark.

The report elements for which you define bookmarks in BIRT Studio can be displayed in BIRT Dashboards using a reportlet gadget. For more information, see *Building BIRT Dashboards.*

Chapter

2

Editing and formatting report content

This chapter contains the following topics:

- Editing and formatting options
- Removing items from a report
- Editing labels
- Applying a theme
- Selecting an item for formatting
- Changing font properties
- Changing the alignment of text
- Applying a number format
- Applying a date-and-time format
- Applying a string format
- Formatting data based on conditions
- Changing the order of columns
- Merging data into one column
- Adding a new column header row
- Changing the page layout to fit data

Editing and formatting options

The template you choose determines the basic layout of your report, and the items available in the report design. Report items can include common elements, such as a report title, page numbers, report-creation date, and a table in which to display report data. You select which report items to include in a report design.

A typical template contains some editable report items, and some non-editable items. Examples of editable items include report titles or tables in which you insert data. Examples of non-editable items include standard copyright or confidentiality statements.

A typical template also defines the style of a report—the color schemes, fonts, spacing and alignment, page size, and other presentation aspects. A template also can provide multiple sets of styles called themes, which you use to change the appearance of a report with one click. A template serves as a basis for reports and supports the developer changing specific report elements such as page size, column width, or font size.

This chapter describes the editing and formatting tasks you can perform using the default installation of BIRT Studio. Some editing or formatting functionality may not be available to you if your system administrator configured BIRT Studio to omit that functionality.

Removing items from a report

When you create a new report from a template, all items in the template appear by default in the report. If you do not want an item, such as a report-generation date or a label, to appear in the report, deselect the item.

How to select the items to remove or display

1 In the side menu, select Report Items. A list of every item in the template appears. Figure 2-1 shows an example of such a list.

Figure 2-1 Report items included in a template

2 Select the items to display in the report. Then, choose Apply. If unsure which item in the list corresponds to which item in the report design, select one item. Then, determine which item appears in the design. Repeat this task for each item in the list.

Editing labels

Labels contain static text. In a typical template, some labels are editable, and others are not. Examples of labels you can edit include report titles or author names. Examples of non-editable labels include standard copyright or confidentiality statements.

When you insert data fields in a table, BIRT Studio displays the values in rows and columns and creates labels, or column headers that show the names of the data fields. These names are appropriate in a database, but usually are not suitable for a report. You typically edit the text in these column headers.

How to edit a label

1 Double-click the label. The label becomes an editable field that contains a blinking cursor.

2 Delete the existing text, and type the text to display.

3 Press Enter. The edited text appears in the label.

Applying a theme

A theme in a template is similar to a theme in Microsoft Windows. A Windows theme helps you personalize, with one click, your computer display to show a particular background and custom color and font settings for various items, such as title bars, icons, and buttons. Similarly, a template theme can determine the colors used in the report, such as the background color for column headers or data rows. A theme can also determine the font used for specific report items, the spacing between rows, the styles of borders, the alignment of text, and more.

Themes are optional. A template can, but does not have to, include themes. A template includes multiple themes if the template designer decides to provide users with different style options to apply to a report. On the other hand, a template does not include themes if, for example, a particular type of report must adhere to a specific layout and style.

How to apply a theme

1 In the side menu, choose Themes. A list of available themes appears, if the template includes themes. Figure 2-2 shows an example of a list of themes.

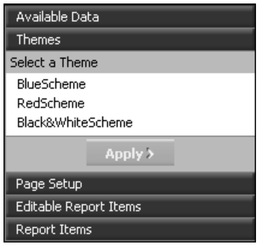


Figure 2-2 Themes included in a template

2 Select a theme, and choose Apply. The report design reflects the styles defined in the theme.

Selecting an item for formatting

You can format only items that the template sets as editable. Typically, these items are titles and tables. In a table, you can format the column headers and the data in the columns. To format an item, first, select the item, then choose the formatting action from the toolbar or from a context menu that appears when you right-click the selected item. The toolbar provides access to the most common actions. The context menu provides access to all the available actions for the selected item.

Selecting a label or a column header is straightforward. You simply click the item. To select data for formatting, select the entire column by clicking the column area or the column handle situated above the column header. A box appears around the selected item. Figure 2-3 shows an example of a selected column header.

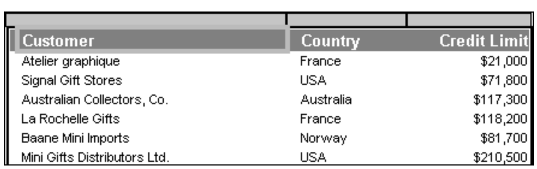


Figure 2-3 A selected column header in a table

Figure 2-4 shows an example of a selected column.

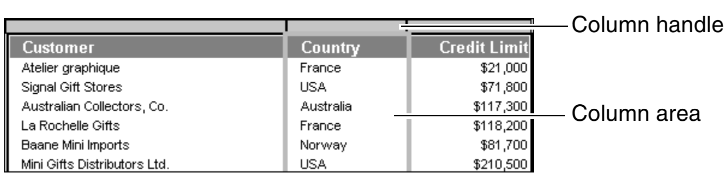


Figure 2-4 A selected column in a table

You cannot select a single row of data in a table, nor can you select a single data value. If you want to highlight a particular row or value, use conditional formatting. This topic appears later in this chapter.

Changing font properties

You can change the font properties of editable labels, column headers, and data in a column. The properties you can set include the font, font size and color, and the background color, and you can make the text bold, italic, or underlined.

How to change font properties

1 Select the label, column header, or column. A box appears around the selected element.

2 Choose Font. Font, shown in Figure 2-5, displays the current font property values. A value of Auto means the property uses a default value, which is specified by a theme, the template, or the software.

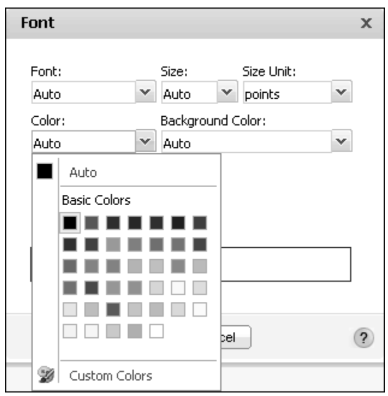


Figure 2-5 Setting font properties

3 Use the drop-down lists to specify the font, size, and size unit for the selected element.

4 In Color, and Background Color, do one of the following:

- Select a color in Basic Colors.
- Select Custom Colors, and use the spectrum bar to specify a color group. Then, select a shade in the gradient square, and choose Pick Color, as shown in Figure 2-6. The selected options appear in the Preview field.

The RGB value of the selected color appears in Color. Choose OK.

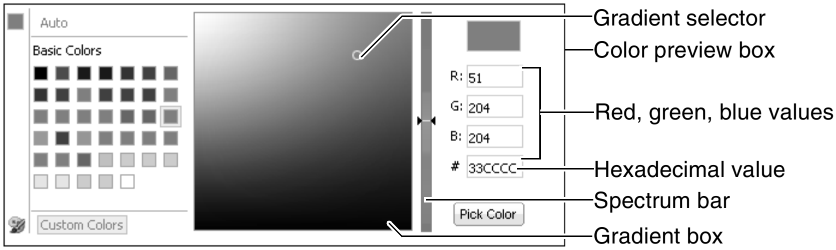


Figure 2-6 Specifying a custom color using color picker

Changing the alignment of text

You can change text alignment in editable labels, column headers, and columns.

How to change the alignment of text

1. Select the label, column header, or column. A box appears around the selected element.
2. Choose one of the alignment options, Align Left, Align Center, or Align Right, as shown in Figure 2-7.

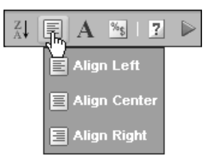


Figure 2-7 Selecting alignment options

Applying a number format

The appearance of numeric data depends on the following factors:

- How the data is stored
- The locale to which the system is set
- The format, if any, set by the template or theme

BIRT Studio provides common formats you can use to change how numbers appear. You can, for example, display numbers with or without decimal values, in scientific notation, as a percentage, or with a currency symbol.

Even when a column displays numbers, the data in the column can be of string type. Postal codes, for example, are frequently stored as string data. Numeric formats do not apply to numbers of string type. When you select a column for formatting, the title of the formatting dialog box tells you what type of data the column contains.

How to apply a number format

1 Select a column that contains numeric data.

2 Choose Format Data.

3 On Number Column Format, select a format from the drop-down list, as shown in Figure 2-8. You can select a standard format or define a custom format. The standard and custom formats are described in the next sections.

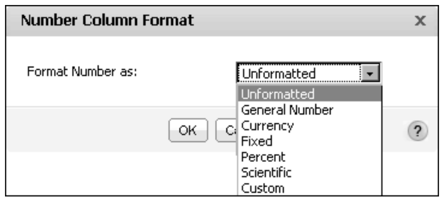


Figure 2-8 Formats for numeric data

Selecting a standard number format

Table 2-1 describes the standard number formats that BIRT Studio supports. The examples in the table reflect the English (United States) locale. If you work in a different locale, the data is formatted differently. For example, a number that appears as 1352.45 in the English (United States) locale appears as 1352,45 in the French (France) locale. When you select a number format, you can set additional formatting options, such as the number of decimal places, the inclusion of a thousands separator, a currency symbol, and so on.

Table 2-1 Standard number formats

Format	Example of data display
General Number	6066.45 or 6066.5 or 6067, depending on the original value. This format displays up to two decimal places. Whole numbers and numbers with one or two decimal places appear in their original format.
Currency	\$6,067.45 or ¥6067 or 6067€, depending on the symbol, symbol position, decimal place, and thousands separator values you set.
Fixed	6067 or 6067.5 or 6,067.45, depending on the decimal place and thousands separator values you set.

(continues)

Table 2-1 Standard number formats (continued)

Format	Example of data display
Percent	45% or 45.8% or %45, depending on the symbol position and decimal place values that you set. This format multiplies the original value by 100 and adds the percent (%) symbol.
Scientific	2E04 or 2.67E04, depending on the decimal place value you set. The number after the E represents the exponent of 10. For example, 2.67E04 means 2.67 multiplied by 10 raised to the fourth power.

Defining a custom number format

You can define a custom number format using special symbols to construct a format pattern. A format pattern shows places held by the currency symbol and the thousands and decimal separators. The thousands separator and the decimal separator are locale-dependent. In the US (English) locale, for example, a comma separates thousands places, and a period separates decimal places. In most European locales, a space separates thousands places, and a comma separates decimal places. Table 2-2 lists example results of custom number format patterns.

Table 2-2 Results of custom number formats

Format pattern	Data in original format	Result of formatting
0000.00	12.5 124.5 1240.553	0012.50 0124.50 1240.55
#.000	100 100.25 100.2567	100.000 100.250 100.257
$#,###	2000.00 20000.00	$2,000 $20,000
ID #	15	ID 15

Applying a date-and-time format

As with numeric data, the appearance of date-and-time data depends on how the data is stored, your system's locale, and the date-and-time format, if any, that is defined in the template or theme. BIRT Studio provides common formats you can use to change how dates and times appear. You can, for example, display dates as January 19, 2008 11:00:00 AM PST or Jan 19, 2008 or 1/19/08.

How to apply a date-and-time format

1 Select a column that contains date-and-time data.

2 Choose Format Data.

3 On Date/Time Column Format, select a format from the drop-down list. You can select a standard format or define a custom format. The standard and custom formats are described in the next sections.

Selecting a standard date-and-time format

The formats that appear in the drop-down list differ, depending on the locale in which you are working. Figure 2-9 shows the sample formats that appear in the English (United States) locale.

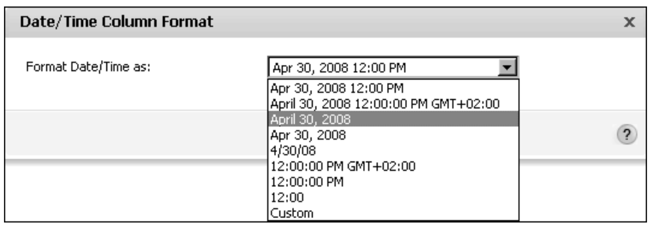


Figure 2-9 Date-and-time formats displayed in English (United States) locale

Figure 2-10 show the sample formats that appear in the Japanese locale.

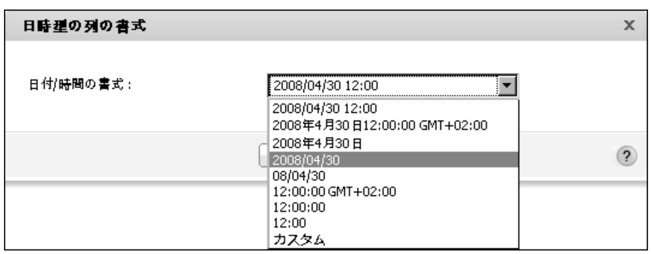


Figure 2-10 Date-and-time formats displayed in Japanese locale

Defining a custom date-and-time format

You can define a custom date-and-time format using special symbols to construct a format pattern. Use custom date formatting only for reports that are viewed in a single locale. Custom formats always display dates in the format you set. This format can be inappropriate for other locales. For example, if you use the format MM-dd-yy, the date January 10, 2006, always appears as 01-10-06, regardless of the locale in which the report is viewed. For locales in which dates customarily appear in day-month-year format, the date 01-10-06 is interpreted as October 1, 2006.

Table 2-3 lists the specials symbols used to define custom date-and-time formats.

Table 2-3 Symbols for defining custom date-and-time formats

Symbol	Description	Example
yy	Short year	08
yyyy	Long year	2008
MM	Month as a number	07
MMM	Short month name	Jul
MMMM	Full month name	July
d	Day in month	10
W	Week in month	2
w	Week in year	28
DD	Day in year	192
E	Short day of week	Thu
EEEE	Long day of week	Thursday
H	Hour in day (0 - 23)	0
k	Hour in day (1 - 24)	24
K	Hour in AM/PM (0 - 11)	0
h	Hour in AM/PM (1 - 12)	12
mm	Minutes	30
ss	Seconds	55
a	AM or PM	8:00:00AM

Table 2-4 shows examples of custom formats and their effects on a date that is stored as April 15, 2006 12:15:30 PM.

Table 2-4 Examples of custom date-and-time formats

Format pattern	Result of formatting
MM-dd-yy	04-15-06
E, M/d/yyyy	Fri, 4/15/2006
MMM d	Apr 15
MMMM	April
yyyy	2006
W	3 (the week in the month)
w	15 (the week in the year)

Table 2-4 Examples of custom date-and-time formats

Format pattern	Result of formatting
DD	105 (the day in the year)
h:mm:ss	12:15:30

Applying a string format

A column that displays string data displays the data as it is stored in the data source. You can format string data to modify text that contains inconsistent capitalization or to include characters, such as a space or a punctuation mark, at a specific place in the string. For example, you can display telephone numbers in one of the following formats:

```
(415) 555-2121
415.555.2121
415-555-2121
```

How to apply a string format

1 Select a column that contains string data.

2 Choose Format Data.

3 On String Column Format, select a format from the drop-down list. You can select a standard format or define a custom format. Standard and custom formats are described in the following sections.

Selecting a standard string format

Table 2-5 describes the string formats you can choose and provides examples of how the formatted data appears.

Table 2-5 Standard string formats

Format	Description
Lowercase	The string displays in all lowercase, for example: john smith
Uppercase	The string displays in all uppercase, for example: JOHN SMITH

Defining a custom string format

You can define a custom string format using special symbols to construct a format pattern. Table 2-6 describes these symbols.

Table 2-6 Symbols for defining custom string formats

Symbol	Description
@	Character placeholder. Each @ character displays a character in the string. If the string has fewer characters than the number of @ symbols that appear in the format pattern, spaces appear. Placeholders are filled from right to left, unless you specify an exclamation point (!) at the beginning of the format pattern.
&	Same as @, except if the string has fewer characters, spaces do not appear.
!	Specifies that placeholders are to be filled from left to right.
>	Converts string characters to uppercase.
<	Converts string characters to lowercase.

Table 2-7 shows example results of custom string format patterns on string data.

Table 2-7 Examples of custom string formats

Format pattern	Data in original format	Results of formatting
(@@@) @@@-@@@@	6175551007 5551007	(617) 555-1007 ( ) 555-1007
(&&&) &&&-&&&&	6175551007 5551007	(617) 555-1007 () 555-1007
!(@@@) @@@-@@@@	6175551007 5551007	(617) 555-1007 (555) 100-7
!(&&&) &&&-&&&&	6175551007 5551007	(617) 555-1007 (555) 100-7
!(@@@) @@@-@@@@ + ext 9	5551007	(555) 100-7 + ext 9
!(&&&) &&&-&&&& + ext 9	5551007	(555) 100-7 + ext 9
>&&&-&&&&&-&&	D1234567xy	D12-34567-XY
<&&&-&&&&&-&&	D1234567xy	d12-34567-xy

Formatting data based on conditions

When you format data in a selected column, the format applies to all the values. Often, it is useful to change the format of data when a certain condition is true. For example, you can display sales numbers in red if the value is below target, and in black if the value is above target.

You also can change the format of data in a column according to the values in another column. For example, in a report that shows customer names and the number of days each customer's invoice is past due, you can highlight in blue any customer name that has an invoice past-due value that is between 60 and 90 days. Then, you can highlight in red and bold any customer name that has an invoice past-due value that is greater than 90 days.

To apply conditional formatting, create a rule defining when and how to change the appearance of data. Only to data in columns supports conditional formatting.

How to set conditional formats

1 Select the column containing the data you want to format. If the column is not selected, you do not see the correct context menu.

2 Choose Format➛Conditional Formatting from the context menu.

3 On Conditional Formatting, create a rule to specify:

   - The format to apply. Choose Format to select formatting options.
   - The condition that must be true to apply the format, for example, Profit Greater than or Equal to 2000. For more information about specifying a condition, see the next section.

   Figure 2-11 shows an example of a rule specified for a selected column, Profit.

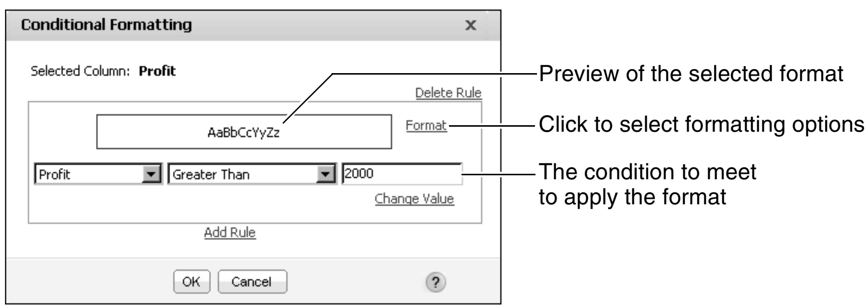


Figure 2-11 Conditional Formatting displaying a rule

4 Choose OK. Figure 2-12 shows the conditional formatting applied to the report. In the Profit column, numbers greater than 2000 are displayed in bold. If the column contains aggregate data, aggregate values do not reflect the applied conditional formatting properties.

Specifying a condition

The condition part of a conditional formatting rule is an If expression that must evaluate to true. For example:

```
If the order total is less than 1000
If the customer credit limit is between 10000 and 20000
If the sales office is Tokyo
If the order date is 7/21/2008
```

Sales Office:	San Francisco		
Product Line	Product Name	Total	Profit
Classic Cars			
	1948 Porsche 356-A Roadster	$3,215.52	$628.32
	1956 Porsche 356A Coupe	$6,534.41	$1,914.31
	1957 Corvette Convertible	$4,151.52	$1,983.69
	1961 Chevrolet Impala	$1,766.40	$1,022.81
	1968 Dodge Charger	$5,298.72	$1,691.04
	1968 Ford Mustang	$7,273.14	**$3,268.86**
	1969 Dodge Charger	$3,005.56	$1,302.39
	1969 Dodge Super Bee	$2,967.30	$760.05
	1970 Plymouth Hemi Cuda	$1,742.88	$976.80
	1971 Alpine Renault 1600s	$1,405.35	$363.69
	1976 Ford Gran Torino	$5,864.88	**$2,778.30**
	1982 Lamborghini Diablo	$687.20	$362.40
Classic Cars		*$43,912.88*	*$17,052.66*
Motorcycles			
	1957 Vespa GS150	$2,238.30	$755.55
	1974 Ducati 350 Mk3 Desmo	$2,755.50	$1,071.60
	1982 Ducati 900 Monster	$1,948.22	$582.32
	1982 Ducati 996 R	$1,274.46	$477.84
Motorcycles		*$8,216.48*	*$2,887.31*
Trucks and Buses			
	1940 Ford Pickup Truck	$4,973.50	**$2,115.33**
	1958 Setra Bus	$2,554.44	$918.54
	1996 Peterbilt 379 Stake Bed with Outrigger	$1,352.86	$579.83
Trucks and Buses		*$8,880.80*	*$3,613.70*
Vintage Cars			
	1939 Cadillac Limousine	$2,163.50	$1,006.50
Vintage Cars		*$2,163.50*	*$1,006.50*
San Francisco		*$63,173.66*	*$24,560.17*
		$63,173.66	*$24,560.17*

Figure 2-12 A report design displaying conditional formatting

The Conditional Formatting dialog box helps you construct the If expression by breaking it down to its logical parts. In Figure 2-11, the expression consists of three parts. In Figure 2-13, the expression has four parts.

In the first field, select a column. This column contains the value that determines when conditional formatting takes effect. The selected column does not have to be the same as the column you selected for formatting in the report design. For example, if Product Name is the column selected for formatting, you can select Profit in this field to indicate that for a certain profit amount, conditional formatting applies to the product name.

In the second field, select the comparison test, or operator, to apply to the selected column. You can select Equal to, Less than, Less than or Equal to, and so on. If you select Is Null, Is Not Null, Is True, or Is False, the If expression does not require additional information. For more information about the operators, see Chapter 9, “Functions and operators.”

If the selected operator requires a comparison to one or more values, one or more additional fields appear. For example, if you select Less than or Equal to, a third field appears. In this field, you type the comparison value. If you select Between or Not Between, a third and fourth field appear. In these fields, type the lower and upper values, as shown in Figure 2-13.

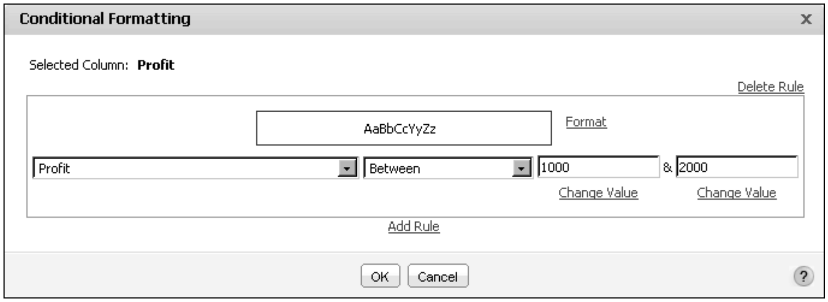


Figure 2-13 Defining a condition that uses the Between operator

Comparing to a literal value

The conditional expression shown in Figure 2-13 evaluates the Profit column and compares each value to determine if it is between 1000 and 2000. The 1000 and 2000 values are literal values that you type. Alternatively, you can select from a list of values from the Profit column. Selecting from a list of values is useful if the comparison value is a product name, and you do not know the exact product names, or if the comparison value is a date, and you do not know the date format to type. If the comparison value is a date, BIRT Studio also provides a calendar tool, which you can use to select a date.

How to select a comparison value from a list of values

1 On Conditional Formatting, below the field that takes a comparison value, choose Change Value.

2 On Value, select Specify literal value, and choose Select Values. The values in the selected column appear. Figure 2-14 shows an example of the values in a product name column.

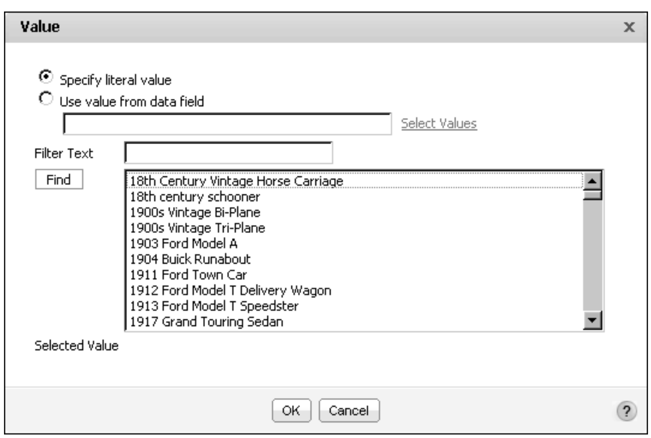


Figure 2-14 Selecting a comparison value

3 If there are too many values in the list, you can type a string in Filter Text to search for values that begin with that string. For example, you can type 18 to view all product names that begin with 18. You cannot search for strings that appear in the middle of a name.

4 Select a value from the list, then choose OK. The value appears in the comparison value field on Conditional Formatting.

Comparing to a value in another column

In a conditional expression, you can compare the values of one column with the values of another column. For example, in a report that displays products, sales prices, and MSRP (Manufacturer Suggested Retail Price), you can create a conditional formatting rule that compares the sale price and MSRP of each product and highlight the names of the products whose sales price is greater than the MSRP.

How to compare to a value in another column

1 On Conditional Formatting, below the field that takes a comparison value, choose Change Value.

2 On Value, select Use value from data field. A list of columns used in the report appears.

3 Select a column from the list, then choose OK. The column name appears in the comparison value field on Conditional Formatting.

Figure 2-15 shows a condition comparing the sale price value with the MSRP value. If the sale price value is greater, the product name appears in bold.

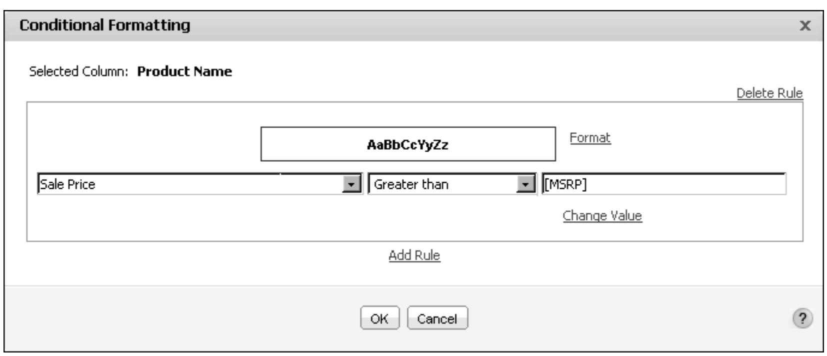
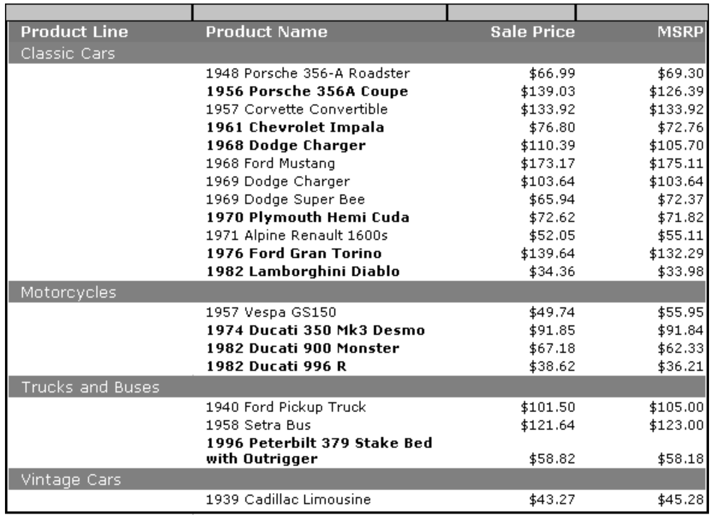


Figure 2-15 Creating a rule that compares values in two columns

Figure 2-16 shows the report design with conditional formatting applied.



Product Line	Product Name	Sale Price	MSRP
Classic Cars			
	1948 Porsche 356-A Roadster	$66.99	$69.30
	1956 Porsche 356A Coupe	$139.03	$126.39
	1957 Corvette Convertible	$133.92	$133.92
	1961 Chevrolet Impala	$76.80	$72.76
	1968 Dodge Charger	$110.39	$105.70
	1968 Ford Mustang	$173.17	$175.11
	1969 Dodge Charger	$103.64	$103.64
	1969 Dodge Super Bee	$65.94	$72.37
	1970 Plymouth Hemi Cuda	$72.62	$71.82
	1971 Alpine Renault 1600s	$52.05	$55.11
	1976 Ford Gran Torino	$139.64	$132.29
	1982 Lamborghini Diablo	$34.36	$33.98
Motorcycles			
	1957 Vespa GS150	$49.74	$55.95
	1974 Ducati 350 Mk3 Desmo	$91.85	$91.84
	1982 Ducati 900 Monster	$67.18	$62.33
	1982 Ducati 996 R	$38.62	$36.21
Trucks and Buses			
	1940 Ford Pickup Truck	$101.50	$105.00
	1958 Setra Bus	$121.64	$123.00
	1996 Peterbilt 379 Stake Bed with Outrigger	$58.82	$58.18
Vintage Cars			
	1939 Cadillac Limousine	$43.27	$45.28

Figure 2-16 A report design with conditional formatting

Specifying multiple conditional formatting rules

You can create up to three conditional formatting rules for a single column. You can, for example, create three rules to set the values of a profit column to one of three colors, depending on the dollar amount. Figure 2-17 shows an example.

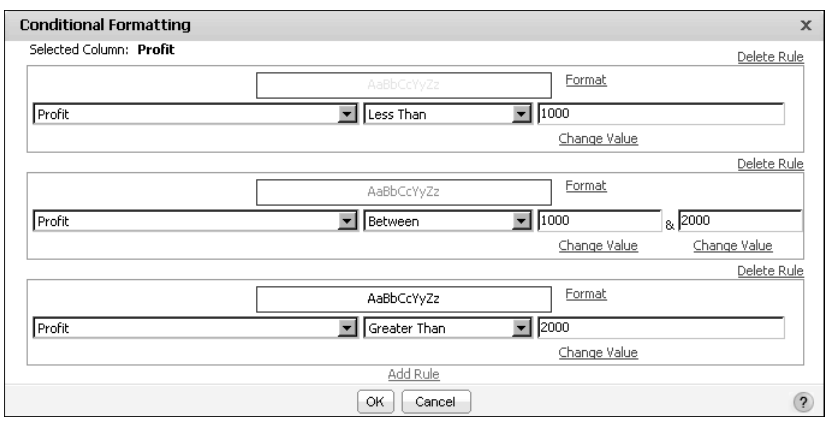


Figure 2-17 Defining multiple conditional formatting rules

For each row of data in the report, BIRT Studio evaluates the rules in the order in which they appear in the list of rules. As it evaluates each rule, BIRT Studio applies the specified format properties if the condition is met.

When creating multiple rules for a column, be careful that the conditions do not cover overlapping values. Consider the following scenario:

- The first rule sets a profit value to blue if the value exceeds 5000.
- The second rule sets the profit value to green if the value exceeds 1000.

If the profit value is 6000, the value appears in green, not blue as you expect, because the condition in both rules is true (6000 exceeds 5000 and 1000), and the second rule supersedes the first rule. For the rules to make sense, the second rule should set the profit value to green if the value is between 1000 and 5000.

Reverting to default formats

When you change the font properties or text alignment for a label or a column of data, your changes override the formats in the template or theme. To revert an element to its default font formats, select the element, choose Font, and set the properties to Auto. To reset text alignment, select the element, and choose an alignment option. You cannot reapply a theme to revert to all the default formats in one step.

If you applied a number, date-and-time, or string format to a column of data, you also can restore these values to the original format that the data source specifies. To do so, select the column, choose Format Data, and select the Unformatted option.

Changing the order of columns

You can change the order of the columns in a table at any time.

How to change the order of columns

Use one of the following methods to change the order of columns:

- Right-click any selected column, and choose Column➛Reorder Columns. On Re-order Columns, shown in Figure 2-18, select each column to move. Then, use the up or down arrow to move the column into the new position. Moving a column up the list moves the column to the left in the table.

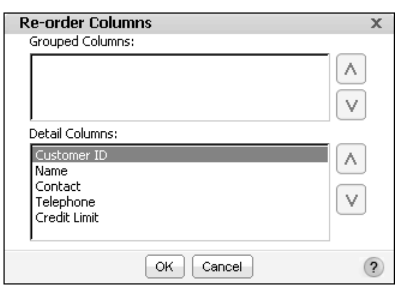
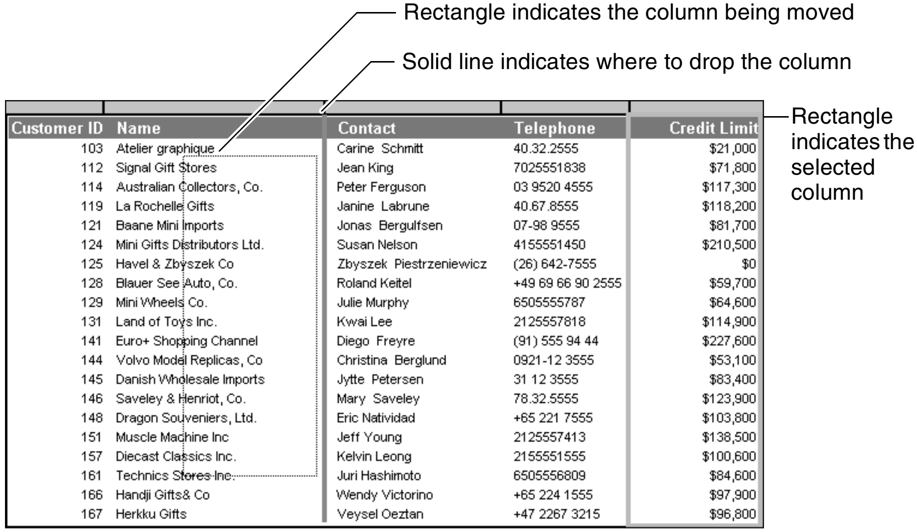


Figure 2-18 Changing the order of columns

- Select the column, and drag it to the new location. As you drag the column, a solid vertical line indicates where you can drop the column, as shown in Figure 2-19.



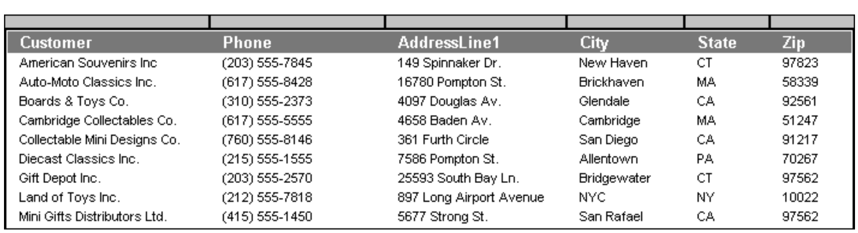
Customer ID	Name	Contact	Telephone	Credit Limit
103	Atelier graphique	Carine Schmitt	40.32.2555	$21,000
112	Signal Gift Stores	Jean King	7025551838	$71,800
114	Australian Collectors, Co.	Peter Ferguson	03 9520 4555	$117,300
119	La Rochelle Gifts	Janine Labrune	40.67.8555	$118,200
121	Baane Mini Imports	Jonas Bergulfsen	07-98 9555	$81,700
124	Mini Gifts Distributors Ltd.	Susan Nelson	4155551450	$210,500
125	Havel & Zbyszek Co	Zbyszek Piestrzeniewicz	(26) 642-7555	$0
128	Blauer See Auto, Co.	Roland Keitel	+49 69 66 90 2555	$59,700
129	Mini Wheels Co.	Julie Murphy	6505555787	$64,600
131	Land of Toys Inc.	Kwai Lee	2125557818	$114,900
141	Euro+ Shopping Channel	Diego Freyre	(91) 555 94 44	$227,600
144	Volvo Model Replicas, Co	Christina Berglund	0921-12 3555	$53,100
145	Danish Wholesale Imports	Jytte Petersen	31 12 3555	$83,400
146	Saveley & Henriot, Co.	Mary Saveley	78.32.5555	$123,900
148	Dragon Souveniers, Ltd.	Eric Natividad	+65 221 7555	$103,800
151	Muscle Machine Inc	Jeff Young	2125557413	$138,500
157	Diecast Classics Inc.	Kelvin Leong	2155551555	$100,600
161	Technics Stores Inc.	Juri Hashimoto	6505556809	$84,600
166	Handji Gifts& Co	Wendy Victorino	+65 224 1555	$97,900
167	Herkku Gifts	Veysel Oeztan	+47 2267 3215	$96,800

Figure 2-19 Moving a column

Merging data into one column

You can merge the data from two or more columns into one column. When you merge data, the data appears on multiple lines. This format is ideal for some types of data, such as addresses. This format also enables you to include more columns without exceeding the width of a page. The following section describes how to merge data into a single column, and how to work with data in a merged column.

Compare the report designs in Figure 2-20 and Figure 2-21. In Figure 2-20, the report design displays each piece of the address information in a separate column. In Figure 2-21, the data in the AddressLine1, City, State, and Zip columns are merged.



Customer	Phone	AddressLine1	City	State	Zip
American Souvenirs Inc	(203) 555-7845	149 Spinnaker Dr.	New Haven	CT	97823
Auto-Moto Classics Inc.	(617) 555-8428	16780 Pompton St.	Brickhaven	MA	58339
Boards & Toys Co.	(310) 555-2373	4097 Douglas Av.	Glendale	CA	92561
Cambridge Collectables Co.	(617) 555-5555	4658 Baden Av.	Cambridge	MA	51247
Collectable Mini Designs Co.	(760) 555-8146	361 Furth Circle	San Diego	CA	91217
Diecast Classics Inc.	(215) 555-1555	7586 Pompton St.	Allentown	PA	70267
Gift Depot Inc.	(203) 555-2570	25593 South Bay Ln.	Bridgewater	CT	97562
Land of Toys Inc.	(212) 555-7818	897 Long Airport Avenue	NYC	NY	10022
Mini Gifts Distributors Ltd.	(415) 555-1450	5677 Strong St.	San Rafael	CA	97562

Figure 2-20 A report design displaying address information in multiple columns

Customer	Phone	AddressLine1 City State Zip
Diecast Classics Inc.	(215) 555-1555	7586 Pompton St. Allentown PA 70267
Land of Toys Inc.	(212) 555-7818	897 Long Airport Avenue NYC NY 10022
Mini Gifts Distributors Ltd.	(415) 555-1450	5677 Strong St. San Rafael CA 97562
Mini Wheels Co.	(650) 555-5787	5557 North Pendale Street San Francisco CA 94217
Muscle Machine Inc	(212) 555-7413	4092 Furth Circle NYC NY 10022
Signal Gift Stores	(702) 555-1838	8489 Strong St. Las Vegas NV 83030

Figure 2-21 A report design displaying address information in a merged column

When you merge data from multiple columns, the column headers also appear on multiple rows, as shown in Figure 2-21. You can improve the format by merging the column headers to remove the City, State, Zip column headers, and editing the remaining column header. Figure 2-22 shows the improved format.

Customer	Phone	Address
Diecast Classics Inc.	(215) 555-1555	7586 Pompton St. Allentown PA 70267
Land of Toys Inc.	(212) 555-7818	897 Long Airport Avenue NYC NY 10022
Mini Gifts Distributors Ltd.	(415) 555-1450	5677 Strong St. San Rafael CA 97562
Mini Wheels Co.	(650) 555-5787	5557 North Pendale Street San Francisco CA 94217
Muscle Machine Inc	(212) 555-7413	4092 Furth Circle NYC NY 10022

Figure 2-22 A report design displaying merged column headers in a single row

Creating a merged column

The following section describes how to create a merged column, and how to merge column headers in a single row.

How to merge data into one column

1 Select all the columns that contain the data to merge. Press Ctrl as you select each column.

2 Right-click one of the selected columns, then choose Column➛Merge Columns, as shown in Figure 2-23.

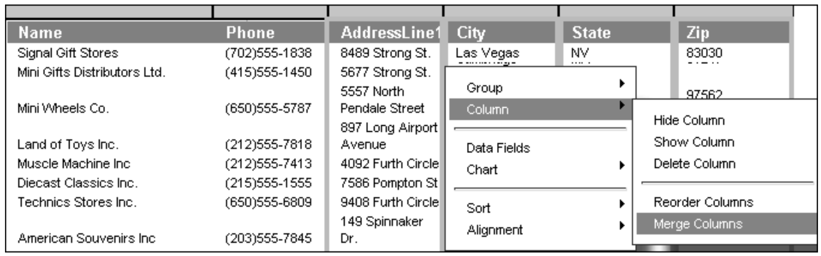


Figure 2-23 Merging columns

How to merge column headers

1 Select the column header in the bottom row, as shown in Figure 2-24. Right-click the selected column header, then choose Cell➛Merge Up.

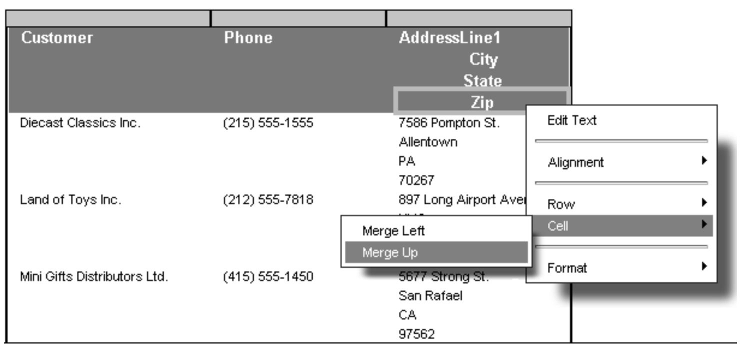


Figure 2-24 Merging column headers

2 Repeat step 1 until only one column header remains.

Working with data in a merged column

When working with a merged column in BIRT Studio, you can perform actions such as modifying font, specifying conditional formatting rules, formatting data strings, creating filters, and so on, just as you would for a regular column.

To work with a merged column, select the column, then choose an option from the context menu. Select Data Item appears, as shown in Figure 2-25. The drop-down list contains a list of columns in the merged column.

Figure 2-25 Selecting a column from the list of merged columns

In Select data, select a column for which you want to perform an action such as formatting, or filtering. Choose OK.

Repeat this task for every column in the merged column for which you want to perform an action.

Adding a new column header row

If you need to add more text or more space in the header area, you can add multiple column header rows. Figure 2-26 shows a row with text added above the default column header row.

U.S. Customers: New accounts signed in quarters 1 and 2, 2008					
Customer	**Phone**	**Street**	**City**	**State**	**Zip**
American Souvenirs Inc	(203) 555-7845	149 Spinnaker Dr.	New Haven	CT	97823
Cambridge Collectables Co.	(617) 555-5555	4658 Baden Av.	Cambridge	MA	51247
Diecast Classics Inc.	(215) 555-1555	7586 Pompton St.	Allentown	PA	70267
Gift Depot Inc.	(203) 555-2570	25593 South Bay Ln.	Bridgewater	CT	97562
Land of Toys Inc.	(212) 555-7818	897 Long Airport Avenue	NYC	NY	10022
Mini Gifts Distributors Ltd.	(415) 555-1450	5677 Strong St.	San Rafael	CA	97562
Mini Wheels Co.	(650) 555-5787	5557 North Pendale Street	San Francisco	CA	94217
Muscle Machine Inc	(212) 555-7413	4092 Furth Circle	NYC	NY	10022
Signal Gift Stores	(702) 555-1838	8489 Strong St.	Las Vegas	NV	83030
Technics Stores Inc.	(650) 555-6809	9408 Furth Circle	Burlingame	CA	94217

Figure 2-26 A report design displaying text in a row above the column header row

When you insert a row, the new row contains the same number of columns as the other rows in the table. To display text that spans multiple columns, as shown in Figure 2-26, you must first merge the columns in that row. As Figure 2-26 shows, you also can add borders to a row.

How to add a new row

1 Select a column header. Right-click the column header, then choose Row.

2 To add a row above the selected column header, choose Insert Row Above. To add a row below the selected column header, choose Insert Row Below. Both options appear in the example shown in Figure 2-27.

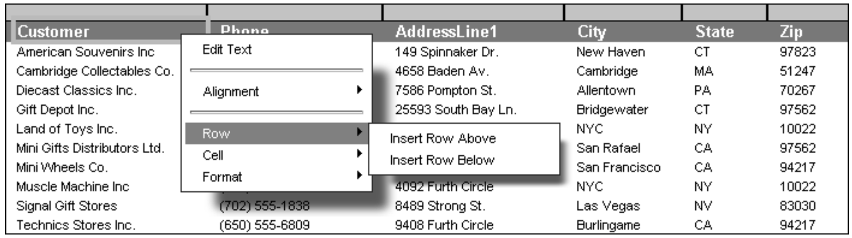


Figure 2-27 Inserting a new row above or below the column header row

A new row appears. Figure 2-28 shows a new row inserted above the default column header row. The row has the same number of columns and the same column widths as the row that follows.

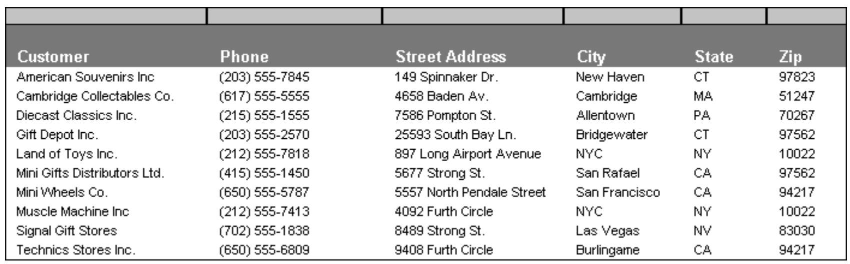


Figure 2-28 A report design displaying an empty row above the column header row

How to merge column headers to type text that spans multiple columns

1 Select the first column header in the new row. Right-click the column header, then choose Cell➛Merge Right, as shown in Figure 2-29. The first cell merges with the second cell to become a single, wider cell.

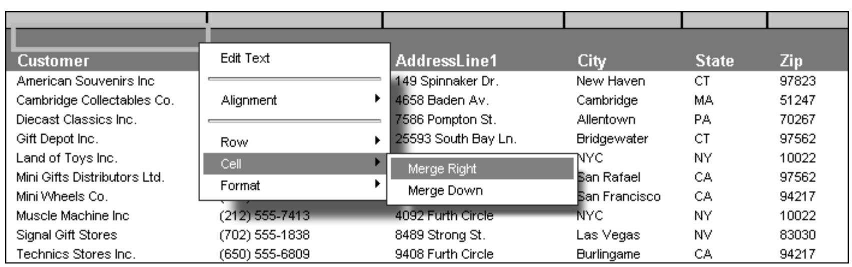
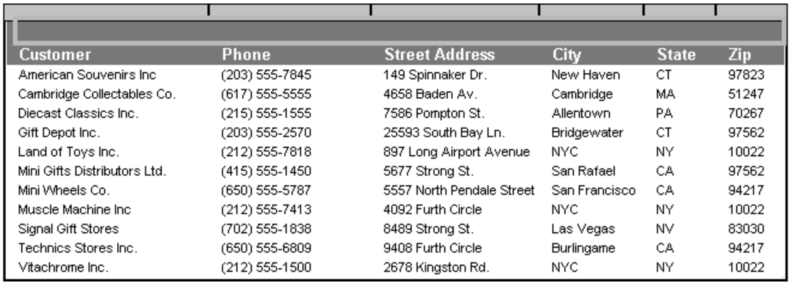


Figure 2-29 Merging column headers

2 Repeat step 1 to merge all cells in the column. Figure 2-30 shows the selection of a single cell, created by merging six cells.



Customer	Phone	Street Address	City	State	Zip
American Souvenirs Inc	(203) 555-7845	149 Spinnaker Dr.	New Haven	CT	97823
Cambridge Collectables Co.	(617) 555-5555	4658 Baden Av.	Cambridge	MA	51247
Diecast Classics Inc.	(215) 555-1555	7586 Pompton St.	Allentown	PA	70267
Gift Depot Inc.	(203) 555-2570	25593 South Bay Ln.	Bridgewater	CT	97562
Land of Toys Inc.	(212) 555-7818	897 Long Airport Avenue	NYC	NY	10022
Mini Gifts Distributors Ltd.	(415) 555-1450	5677 Strong St.	San Rafael	CA	97562
Mini Wheels Co.	(650) 555-5787	5557 North Pendale Street	San Francisco	CA	94217
Muscle Machine Inc	(212) 555-7413	4092 Furth Circle	NYC	NY	10022
Signal Gift Stores	(702) 555-1838	8489 Strong St.	Las Vegas	NV	83030
Technics Stores Inc.	(650) 555-6809	9408 Furth Circle	Burlingame	CA	94217
Vitachrome Inc.	(212) 555-1500	2678 Kingston Rd.	NYC	NY	10022

Figure 2-30 Result of merging all the column headers

3 To display text in this cell, double-click in the cell, type the text, then press Enter.

4 To add borders:

1 Select the cell. Right-click it, and choose Format➛Border, as shown in Figure 2-31.

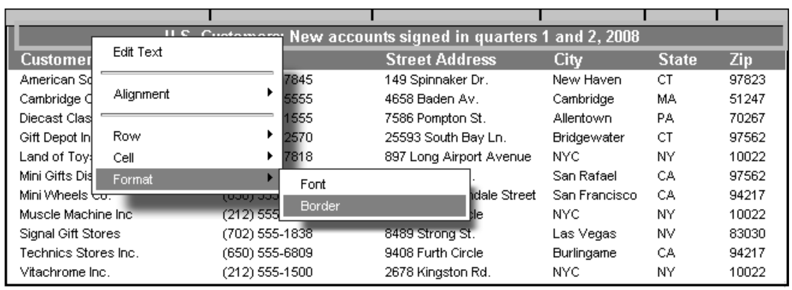


Figure 2-31 Adding borders around the merged column header

2 On Border, specify the type of border or borders to add. You can set the border style, color, and width. Figure 2-32 shows the bottom border set to create a solid, white border that is one pixel wide. Choose OK.

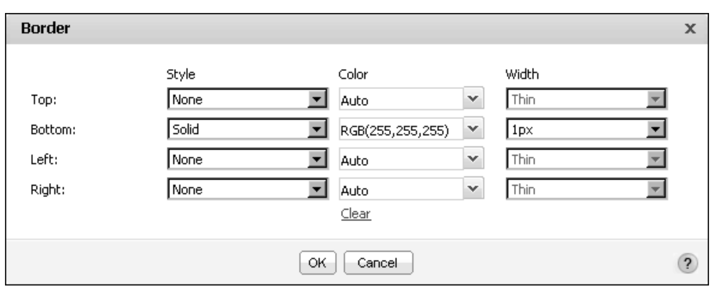


Figure 2-32 Specifying the border type

Changing the page layout to fit data

When you insert data fields in a table, by default, the table expands to accommodate the data. The width of the columns depend on several factors, including the column sizes defined in the template and the data field sizes defined in the original data source.

Figure 2-33 shows an example of a table that uses the default layout. The columns are approximately the same width, even though the data in the state and zip code columns, for example, occupy less space than the data in the Customer and Street Address columns. If you insert many data fields in a table, you probably need to adjust the width of columns to improve the spacing between the columns.

Customer	Customer ID	Credit Limit	Contact	Phone	Street Address	City	State	Zip Code
Signal Gift Stores	112	$71,800	Jean King	(702) 555-1838	8489 Strong St.	Las Vegas	NV	83030
Mini Gifts Distributors Ltd.	124	$210,500	Susan Nelson	(415) 555-1450	5677 Strong St.	San Rafael	CA	97562
Mini Wheels Co.	129	$64,600	Julie Murphy	(650) 555-5787	5557 North Pendale Street	San Francisco	CA	94217
Land of Toys Inc.	131	$114,900	Kwai Lee	(212) 555-7818	897 Long Airport Avenue	NYC	NY	10022
Muscle Machine Inc	151	$138,500	Jeff Young	(212) 555-7413	4092 Furth Circle	NYC	NY	10022
Diecast Classics Inc.	157	$100,600	Kelvin Leong	(215) 555-1555	7586 Pompton St.	Allentown	PA	70267
Technics Stores Inc.	161	$84,600	Juri Hashimoto	(650) 555-6809	9408 Furth Circle	Burlingame	CA	94217

Figure 2-33 A report design displaying data in the default table layout

If you intend to print a report or export the report to a page-based format, such as PDF or Word, check the report output in those formats to ensure that all the data fits on the page. Figure 2-34 shows a portion of the example report in PDF format. The Zip Code column does not fit on the page.

Customer Directory

Customer	Customer ID	Credit Limit	Contact	Phone	Street Address	City	State
Signal Gift Stores	112	$71,800	Jean King	(702) 555-1838	8489 Strong St.	Las Vegas	NV
Mini Gifts Distributors Ltd.	124	$210,500	Susan Nelson	(415) 555-1450	5677 Strong St.	San Rafael	CA
Mini Wheels Co.	129	$64,600	Julie Murphy	(650) 555-5787	5557 North Pendale Street	San Francisco	CA
Land of Toys Inc.	131	$114,900	Kwai Lee	(212) 555-7818	897 Long Airport Avenue	NYC	NY
Muscle Machine Inc	151	$138,500	Jeff Young	(212) 555-7413	4092 Furth Circle	NYC	NY
Diecast Classics Inc.	157	$100,600	Kelvin Leong	(215) 555-1555	7586 Pompton St.	Allentown	PA

Figure 2-34 Viewing the default layout in a PDF document

You can change the page layout to better fit the data in your report by using the following techniques:

- Adjusting the widths of the columns
- Changing the layout type, page size and orientation

Adjusting column widths

If you want to control precisely the widths of columns in a table, first set the Layout Preference property in Page Setup to Fixed Width. While BIRT Studio lets you set a column's width, even if the layout type is Auto Expand Width, the actual width of the column is determined by other factors as described earlier.

How to set a column's width

1 Select the column, right-click it, then choose Column➛Column Width.

2 On Column Properties, shown in Figure 2-35, type the width value. The default unit is inches, but you can select cm, mm, picas, or points.

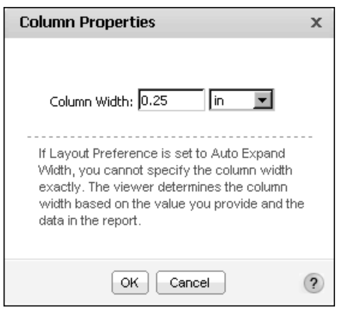


Figure 2-35 Setting a column's width using Column Properties

Changing the layout type, page size, and orientation

As described previously, a table's width expands to display all the columns of data. This feature is typical for viewing a report online. If, however, you want to design a report that shows the same output, whether it is viewed online or in a page-based format such as PDF, you can change the report's layout type from Auto Expand Width to Fixed Width. If you select Fixed Width, you can specify a page width, and this value affects the web report. If you use the default Auto Expand Width setting, the page size values apply only to PDF and printed reports.

How to set the layout type, page size, and orientation

1 In the side menu, choose Page Setup. Figure 2-36 shows an example of settings in Page Setup.

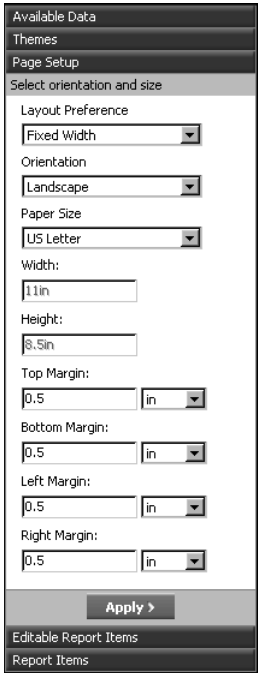


Figure 2-36 Specifying page setup options

2 Set the following options:

1 In Layout Preference:

- Select Auto Expand Width to set the column widths based on the other page options you select and on the number of columns in the table.
- Select Fixed Width to set the page width and precise column widths yourself.

2 In Orientation, accept the default selection of Auto, or select Portrait or Landscape mode. If you choose Auto, BIRT Studio sets the page orientation depending on whether the report is longer or wider. If your table is too wide to print in a standard 8.5 inch page width, select Landscape.

3 In Paper Size, select one of the standard sizes, A4, US Letter, or US Legal. Alternatively, select Custom to set a custom page width and height.

4 If you choose Custom, specify the page size values in Width and Height.

5 In Top Margin, Bottom Margin, Left Margin, and Right Margin, specify the margin sizes to use for each page.

Chapter

3

Inserting calculated data

This chapter contains the following topics:

- About calculated data
- Creating a computed column
- Adding data fields for an expression
- Writing expressions

About calculated data

Most business reports require calculations to track sales, finances, inventory, and other critical business activities. You can keep a count of items in a warehouse, or provide more complex financial data, such as tracking stock portfolio performance over time. Some of this calculated data can be included in the data set if the person who created the data set knew that you and other users need to work with this type of data.

Sometimes, however, a data set does not provide all the data you want to display in your report. In this case, you can create your own calculations. To do so, you create a new field called a computed column. Figure 3-1 shows a report design that uses a computed column, Total, to display the total of each order line item. In this example, the Total values need to be calculated, because the data set does not include this data.

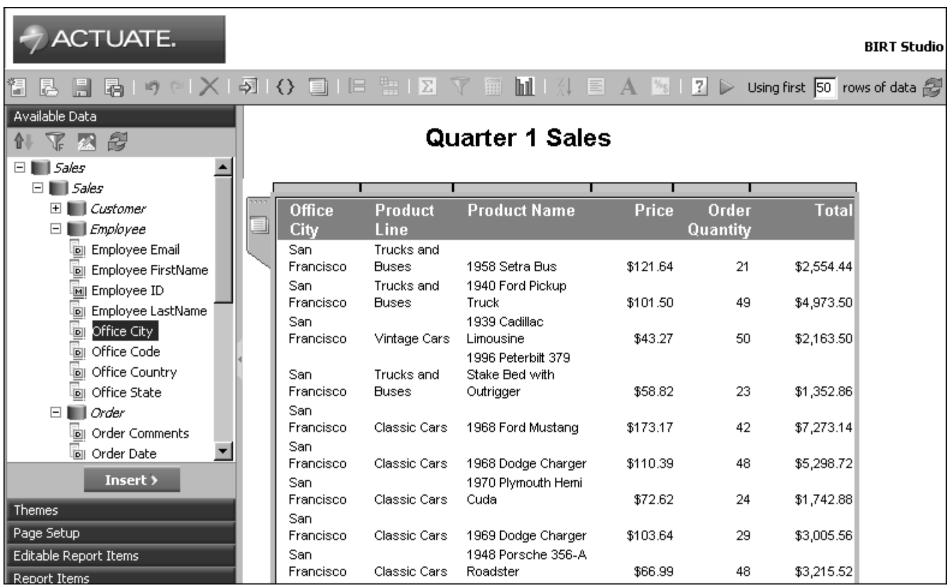


Figure 3-1 Displaying a computed column, Total, in a report design

Creating a computed column

When you create a computed column, you write an expression, which is a statement that indicates how to calculate the data. In the report in Figure 3-1, each value in the Total column is calculated by multiplying the value in the QUANTITYORDERED field with the value in the PRICEEACH field.

Figure 3-2 shows the computed column, Total, using the following expression:

```
[QUANTITYORDERED] * [PRICEEACH]
```

When you refer to a data field in an expression, you must enclose the field name within brackets ([ ]). The field names are case-sensitive.

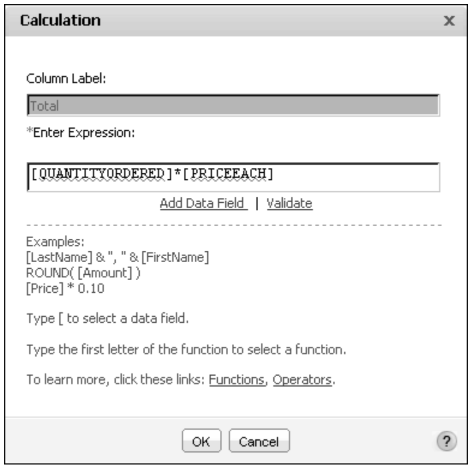


Figure 3-2 An expression for the computed column

About expressions

BIRT Studio supports typical mathematical operations, such as addition, subtraction, multiplication, and division. Computed columns, however, are not limited to mathematical calculations. BIRT Studio also supports many functions for manipulating date-and-time and string data.

A function is a set of instructions that do something and return a result. For example, if a customer name field contains values with leading or trailing blank characters, you can remove the blank characters by using the TRIM( ) function as follows:

```
TRIM([CustomerName])
```

In this example, [CustomerName] is the input value, or argument, you supply to the TRIM( ) function to tell the function to trim the values in the CustomerName field.

More examples of expressions used in computed columns appear later. For descriptions of the supported functions, see Chapter 9, "Functions and operators."

How to create a computed column

1 Select the table column to the left of the new computed column you want to insert.

2 On the toolbar, choose New Computed Column, as shown in Figure 3-3.

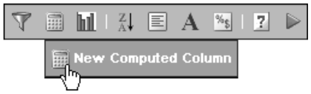


Figure 3-3 Choosing New Computed Column

3 In Calculation, type a name in Column label for the computed column. The name you specify appears in the column header.

4 In Enter Expression, type the expression to calculate the values you want to display:

- To use a data field in the expression, type the left bracket ([), then select a field from the list that appears, as shown in Figure 3-4.

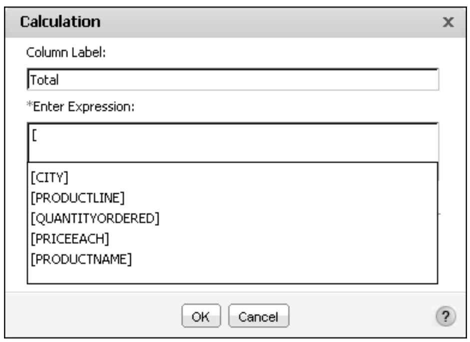


Figure 3-4 Selecting a data field to use in an expression

The list shows only the fields in use in the report design. It does not show all the fields in the data set. To use a field that is not in the list, you must add the field to the report design.

- To use a function, type the first letter of the function, then select a function from the list that appears, as shown in Figure 3-5. The functions indicate the arguments, if any, you need to supply.

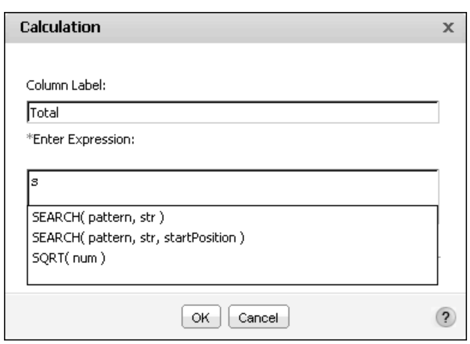


Figure 3-5 Selecting a function to use in an expression

5 When you finish writing the expression, choose Validate. If the expression is syntactically correct, the message, The Expression is valid appears. If the expression contains an error, a message that describes the error appears.

6 After you validate the expression, choose OK. The computed column appears in the report design.

A computed column containing data of numeric or date-and-time type needs to be formatted. For example, the Total column in the example above should be formatted as currency.

Creating aggregate data in a computed column

BIRT Studio enables you to create aggregate data for the values in a computed column. For more information about the types of aggregate calculations you can create, see “Aggregating data,” in Chapter 4, “Organizing data in a report.”

Adding data fields for an expression

It is common to use a data field in an expression for a computed column. As mentioned earlier, a computed column only has access to data fields that are used in the report design. Sometimes, however, you need to write an expression that uses a field provided by the data set, but you do not want to display the field values in the report.

For example, if you insert a CustomerName field and a Phone field in a table, then you insert a computed column, the computed column only has access to the CustomerName and Phone fields. You cannot create an expression that combines, for example, values from the AddressLine1, City, State, and PostalCode fields. To create such a computed column, you must first add the data fields to the table, without actually inserting the fields in the table.

How to add a data field

1 Right-click the table handle, as shown in Figure 3-6.

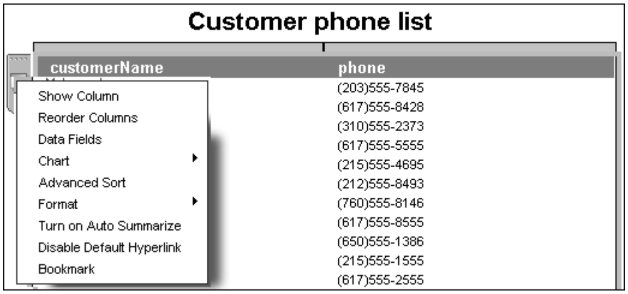


Figure 3-6 Adding a data field to a table

2 Choose Data Fields. Data Fields appears. Figure 3-7 shows an example of Data Fields displaying two fields, customerName and phone, which are currently used in the table.

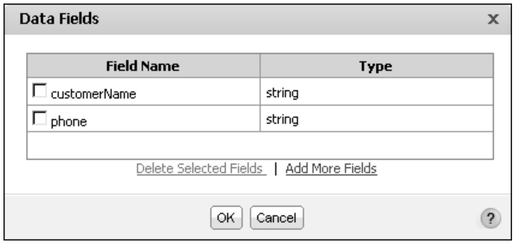


Figure 3-7 Data Fields displaying the fields currently used in the table

3 Choose Add More Fields. Data Set displays all the fields in the data set, as shown in Figure 3-8.

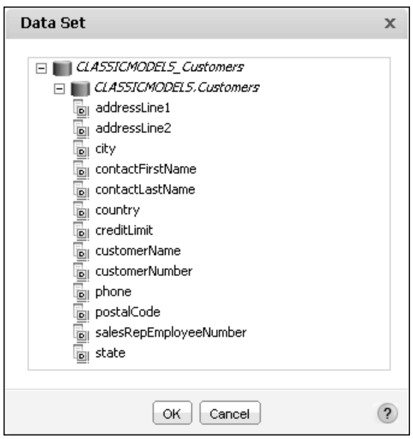


Figure 3-8 Data Set displaying all the available fields

4 Select the field to add. To add multiple fields, press Ctrl as you select each field. Choose OK.

5 On Data Fields choose OK. The selected fields are available to use in any expression.

If you are in the process of defining an expression for a computed column, you can choose the Add Data Field link, as shown in Figure 3-9, then follow steps 3 through 5 to add fields.

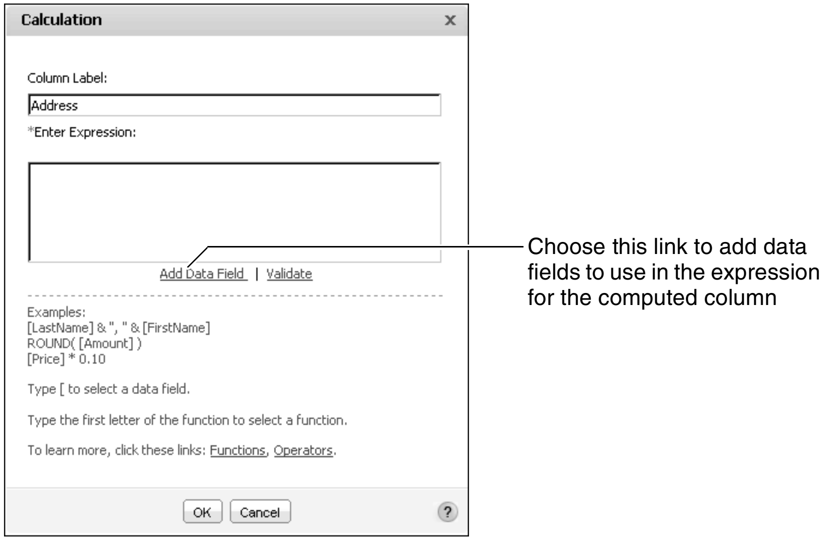


Figure 3-9 Adding data fields to an expression for a computed column

Writing expressions

An expression is a statement that produces a value. An expression can be a literal value, such as:

```
1.23
"Hello, World!"
```

An expression can contain any combination of literal values, operators, functions, and references to data fields, as shown in the following examples.

The following expression displays a customer's first and last names which are stored in two fields. The & operator concatenates string values:

```
[FirstName] & " " & [LastName]
```

The following expression displays a full address by concatenating values from four data fields and adding commas where appropriate:

```
[Address1] & ", " & [City] & ", " & [State] & " " & [Zipcode]
```

The following expression calculates a gain or loss in percent. The expression uses the subtraction, division, and multiplication operators, -, /, and *:

```
([SalePrice] - [UnitPrice])/[UnitPrice] * 100
```

The following expression uses the DIFF_DAY( ) function to calculate the number of days it took to process an order for shipping. The function takes two arguments, a start date and an end date. In the example, two data fields are supplied as the arguments. BIRT Studio runs this expression against each value in the fields and calculates a value for each data row:

```
DIFF_DAY([OrderDate], [ShippedDate])
```

The following expression uses the ADD_DAY( ) function to calculate a payment due date when the payment term is 30 days from the date of the invoice. The function takes two arguments, a start date and the number of days to add:

```
ADD_DAY([InvoiceDate], 30)
```

The following expression uses the IF( ) function to evaluate if the value in the country column is UK. If the condition is true, the function replaces the value with United Kingdom. If the condition is false, the country values appear as stored:

```
IF(([Country]="UK"),"United Kingdom",[Country])
```

Using numbers and dates in an expression

When you create an expression that contains a literal number, you must type the number according to the conventions of the US English locale. In other words, use a period (.), not a comma (,) as the decimal separator, even if you are working in, for example, the French locale. For example:

```
Correct: ([Quantity] * [Price]) * 1.5
Incorrect: ([Quantity] * [Price]) * 1,5
```

Similarly, when you create an expression that contains a literal date, type the date according to the conventions of the US English locale. For example, if you are working in the French locale, type 03/12/2007 to represent March 12, 2007. Do not type 12/03/2007, which is the convention for the French locale. You must enclose literal date values in double quotation marks(" "), as shown in the following expression, which calculates the number of days from the order date to Christmas:

```
DIFF_DAY([OrderDate], "12/25/08")
```

About reserved characters in an expression

Some characters are reserved for internal use and have a special meaning. For example, as you saw in the expression examples in the previous sections, BIRT Studio uses brackets to denote a data field. The following characters are reserved in BIRT Studio:

```
[
]
?
' (single quotation mark)
```

If the name of a data field contains a reserved character, BIRT Studio encloses the reserved character in single quotes (') when you select the data field to use in an expression, for example `'['`. For example, if the name of a data field is OBSOLETE?, BIRT Studio changes it to [OBSOLETE'?'] in the expression. If you type [OBSOLETE?] in the expression, the dialog box displays an error message. To avoid syntax errors, always select the field from the list in the Calculation dialog box and let the software construct the correct expression.

Figure 3-10 shows an example of the Calculation dialog box displaying a list of data fields, three with reserved characters in their names. The first, second, and fifth fields show both versions of their names, the changed name and the original name with reserved characters, as follows:

```
[ORDER''S STATUS] - ORDER'S STATUS
[PRODUCTCODE'['4-digit']'] - PRODUCTCODE[4-digit]
[OBSOLETE'?'] - OBSOLETE?
```

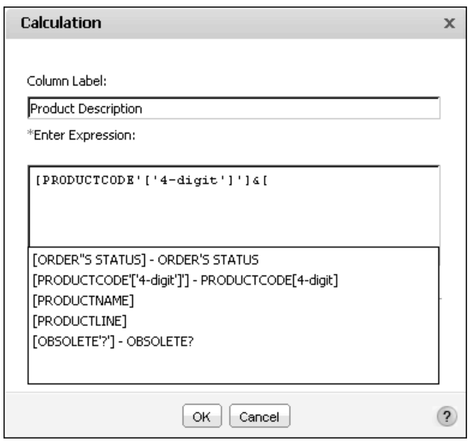


Figure 3-10 Viewing a list of available data fields in Calculation

Chapter

4

Organizing data in a report

This chapter contains the following topics:

- Sorting data
- Organizing data in groups
- Organizing data in sections
- Aggregating data
- Hiding details
- Starting each group or section on a new page

Sorting data

When you insert data in a report design, the data set determines the default sort order for the data rows. If the data set sorts a field in ascending order, the column values appear in ascending order in the design. Typically, however, data appears randomly. A field is likely to display customer names, for example, in the order in which customers were added to the database, rather than in alphabetical order. Sorting data, therefore, is an important task in creating a useful report.

Compare the reports in Figure 4-1.

Country	Customer	Credit Limit
France	Atelier graphique	21000
USA	Signal Gift Stores	71800
Australia	Australian Collectors, Co.	117300
France	La Rochelle Gifts	118200
Norway	Baane Mini Imports	81700
USA	Mini Gifts Distributors Ltd.	210500
Poland	Havel & Zbyszek Co	0
Germany	Blauer See Auto, Co.	59700
USA	Mini Wheels Co.	64600
USA	Land of Toys Inc.	114900
Spain	Euro+ Shopping Channel	227600
Sweden	Volvo Model Replicas, Co	53100
Denmark	Danish Wholesale Imports	83400
France	Saveley & Henriot, Co.	123900
Singapore	Dragon Souveniers, Ltd.	103800

Country	Customer	Credit Limit
Australia	Australian Collectors, Co.	117300
Denmark	Danish Wholesale Imports	83400
France	Atelier graphique	21000
France	La Rochelle Gifts	118200
France	Saveley & Henriot, Co.	123900
Germany	Blauer See Auto, Co.	59700
Norway	Baane Mini Imports	81700
Poland	Havel & Zbyszek Co	0
Singapore	Dragon Souveniers, Ltd.	103800
Spain	Euro+ Shopping Channel	227600
Sweden	Volvo Model Replicas, Co	53100
USA	Signal Gift Stores	71800
USA	Mini Gifts Distributors Ltd.	210500
USA	Mini Wheels Co.	64600
USA	Land of Toys Inc.	114900

Figure 4-1 Displaying unsorted and sorted data in report designs

The report on the left displays the data rows in the order the data set returns them. The report on the right displays the same data, except that the rows are sorted by country in alphabetical order.

You can sort data in ascending or descending order, and you can sort data by multiple columns. For example, Figure 4-2 shows the results of sorting by country, then by customer name. For rows where the country names repeat, France and USA, the customer names appear in alphabetical order.

Country	Customer	Credit Limit
Australia	Australian Collectors, Co.	117300
Denmark	Danish Wholesale Imports	83400
France	Atelier graphique	21000
France	La Rochelle Gifts	118200
France	Saveley & Henriot, Co.	123900
Germany	Blauer See Auto, Co.	59700
Norway	Baane Mini Imports	81700
Poland	Havel & Zbyszek Co	0
Singapore	Dragon Souveniers, Ltd.	103800
Spain	Euro+ Shopping Channel	227600
Sweden	Volvo Model Replicas, Co	53100
USA	Land of Toys Inc.	114900
USA	Mini Gifts Distributors Ltd.	210500
USA	Mini Wheels Co.	64600
USA	Signal Gift Stores	71800

Figure 4-2 Displaying data sorted by country and customer

Sorting on a single column

To sort data on a single column, first select the column. Then, choose Sort Ascending or Sort Descending, as shown in Figure 4-3.

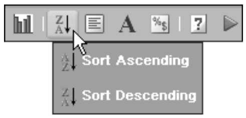


Figure 4-3 Selecting a sort option

Sorting on multiple columns

You can sort data by up to three columns. When you sort by multiple columns, you do not use the Sort Ascending or Sort Descending buttons. Instead, you use the Advanced Sort dialog box, shown in Figure 4-4.

When you sort on multiple columns, it is important to understand the order of precedence for the sort. On Advanced Sort, select the columns in the order in which to sort data. For example, to sort data by city first, then by customer name, you must select the columns in that order on Advanced Sort.

How to sort data on multiple columns

1. Select a column. You can select any column in the table.
2. Right-click the selected column, then choose Sort➛Advanced Sort from the context menu.
3. On Advanced Sort, select a column from the first drop-down list, and choose either Ascending or Descending order.
4. Select another column from the second drop-down list, and choose the sort order.
5. Optionally, select a third column on which to sort. Figure 4-4 shows the sort criteria for the report design in Figure 4-2.

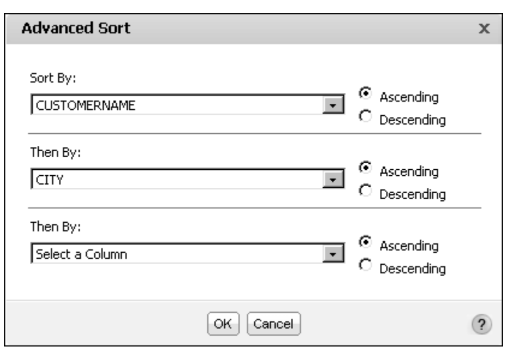


Figure 4-4 Specifying advanced sorting

Restoring data to its original order

To restore data to its original, unsorted order, complete the following steps.

1 Select a column. You can select any column in the table. Right-click the column, then choose Sort➛Advanced Sort.

2 In Advanced Sort, for each column specified for a sort, select Select a Column from the drop-down list, as shown in Figure 4-5. Choose OK.

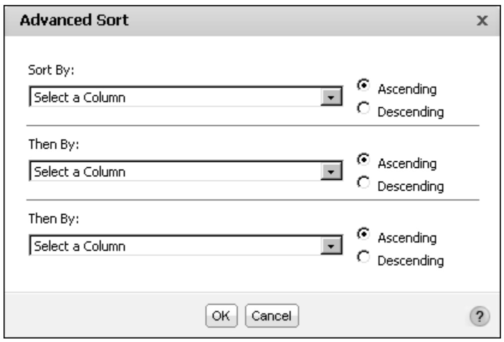


Figure 4-5 Removing sorting criteria

Organizing data in groups

As the previous section describes, sorting data makes a report more readable and useful. Sorting, however, is only one of the ways to organize data in a report. It is common for reports to present data that is organized into meaningful groups, especially reports that contain large amounts of data.

Consider the task of listing every item a corporation owns, along with information such as the category, purchase price, purchase date, inventory code, location, and supplier. If a report presents all these items in an unorganized list or even in a sorted list, there is no way to determine how much the corporation has spent, for example, on office furniture or computer hardware, because this information cannot be calculated. The report cannot help you see which year had the most purchases, nor is it easy to tell which items are located in a field office and which items are in the warehouse.

To organize this information into a useful inventory report, you create data groups. Data groups contain related data rows. For example, you can create a report that lists items by category—all office furniture in one group, all computer hardware in another group, all computer software in a third group, and so on. For each group, you can calculate aggregate data, such as the total purchase price or the count of items in a group. Organizing data in groups makes it easier to compare and analyze information.

Compare the reports in Figure 4-6 and Figure 4-7. The report design in Figure 4-6 displays sales information in a simple list. The data rows are sorted by sales office, then by product line. Notice the repeated sales office name and product lines.

Sales Office	Product Line	Product Name	Total	Profit
San Francisco	Classic Cars	1948 Porsche 356-A Roadster	$3,215.52	$628.32
San Francisco	Classic Cars	1956 Porsche 356A Coupe	$6,534.41	$1,914.31
San Francisco	Classic Cars	1957 Corvette Convertible	$4,151.52	$1,983.69
San Francisco	Classic Cars	1961 Chevrolet Impala	$1,766.40	$1,022.81
San Francisco	Classic Cars	1968 Dodge Charger	$5,298.72	$1,691.04
San Francisco	Classic Cars	1968 Ford Mustang	$7,273.14	$3,268.86
San Francisco	Classic Cars	1969 Corvair Monza	$4,636.50	$1,694.88
San Francisco	Classic Cars	1969 Dodge Charger	$3,005.56	$1,302.39
San Francisco	Classic Cars	1969 Dodge Super Bee	$2,967.30	$760.05
San Francisco	Classic Cars	1969 Ford Falcon	$7,720.18	$3,899.88
San Francisco	Classic Cars	1970 Plymouth Hemi Cuda	$1,742.88	$976.80
San Francisco	Classic Cars	1970 Triumph Spitfire	$2,929.92	$723.84
San Francisco	Classic Cars	1971 Alpine Renault 1600s	$1,405.35	$363.69
San Francisco	Classic Cars	1976 Ford Gran Torino	$5,864.88	$2,778.30
San Francisco	Classic Cars	1982 Lamborghini Diablo	$687.20	$362.40
San Francisco	Classic Cars	2001 Ferrari Enzo	$5,485.92	$2,331.45
San Francisco	Motorcycles	1957 Vespa GS150	$2,238.30	$755.55

Figure 4-6 Displaying data in a simple list

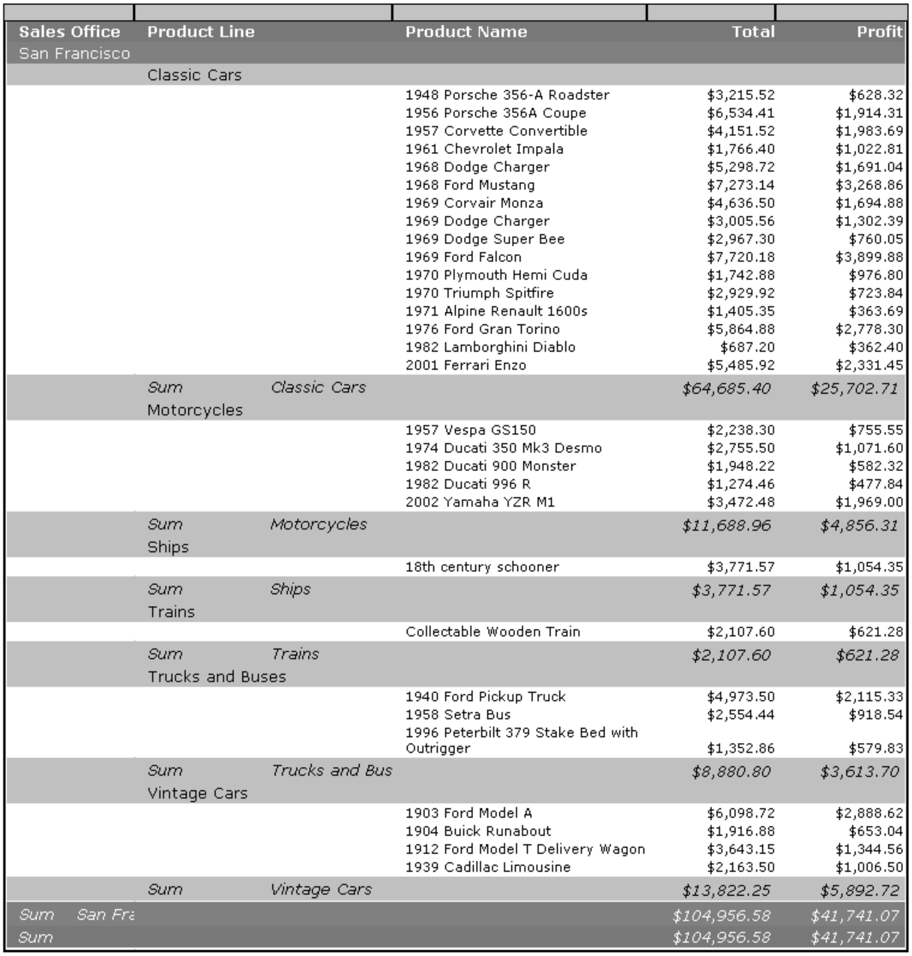
The report design in Figure 4-7 shows the same data. Unlike the report design in Figure 4-6, this design groups the rows by sales office then by product line. Data appears in a hierarchy. The report design does not contain repeated values for office names and product lines. In addition, totals, displayed in italics, appear at the end of each product line and sales office group.

As the example reports Figure 4-6 and Figure 4-7 show, a report design that groups data provides a more effective way to view data. When you group data, you can

- Add subtotals, counts, averages, or other aggregate information at the beginning or end of each group.
- Insert a page break before or after each group to start each group of data on a new page.
- Hide the details of each group to view a summary report.

In addition, BIRT Studio completes the following actions when you group data:

- Removes duplicate values.
- Sorts the values of each group. For example, a product line group displays the product lines in alphabetical order (Classic Cars to Vintage Cars), as shown in Figure 4-7.
- Generates a table of contents when you view the finished report in the viewer. The table of contents, which displays the group values, supports navigating to specific locations in the report. This feature is particularly useful when a report contains many pages.



Sales Office	Product Line	Product Name	Total	Profit
San Francisco				
	Classic Cars			
		1948 Porsche 356-A Roadster	$3,215.52	$628.32
		1956 Porsche 356A Coupe	$6,534.41	$1,914.31
		1957 Corvette Convertible	$4,151.52	$1,983.69
		1961 Chevrolet Impala	$1,766.40	$1,022.81
		1968 Dodge Charger	$5,298.72	$1,691.04
		1968 Ford Mustang	$7,273.14	$3,268.86
		1969 Corvair Monza	$4,636.50	$1,694.88
		1969 Dodge Charger	$3,005.56	$1,302.39
		1969 Dodge Super Bee	$2,967.30	$760.05
		1969 Ford Falcon	$7,720.18	$3,899.88
		1970 Plymouth Hemi Cuda	$1,742.88	$976.80
		1970 Triumph Spitfire	$2,929.92	$723.84
		1971 Alpine Renault 1600s	$1,405.35	$363.69
		1976 Ford Gran Torino	$5,864.88	$2,778.30
		1982 Lamborghini Diablo	$687.20	$362.40
		2001 Ferrari Enzo	$5,485.92	$2,331.45
	Sum Classic Cars		*$64,685.40*	*$25,702.71*
	Motorcycles			
		1957 Vespa GS150	$2,238.30	$755.55
		1974 Ducati 350 Mk3 Desmo	$2,755.50	$1,071.60
		1982 Ducati 900 Monster	$1,948.22	$582.32
		1982 Ducati 996 R	$1,274.46	$477.84
		2002 Yamaha YZR M1	$3,472.48	$1,969.00
	Sum Motorcycles		*$11,688.96*	*$4,856.31*
	Ships			
		18th century schooner	$3,771.57	$1,054.35
	Sum Ships		*$3,771.57*	*$1,054.35*
	Trains			
		Collectable Wooden Train	$2,107.60	$621.28
	Sum Trains		*$2,107.60*	*$621.28*
	Trucks and Buses			
		1940 Ford Pickup Truck	$4,973.50	$2,115.33
		1958 Setra Bus	$2,554.44	$918.54
		1996 Peterbilt 379 Stake Bed with Outrigger	$1,352.86	$579.83
	Sum Trucks and Bus		*$8,880.80*	*$3,613.70*
	Vintage Cars			
		1903 Ford Model A	$6,098.72	$2,888.62
		1904 Buick Runabout	$1,916.88	$653.04
		1912 Ford Model T Delivery Wagon	$3,643.15	$1,344.56
		1939 Cadillac Limousine	$2,163.50	$1,006.50
	Sum Vintage Cars		*$13,822.25*	*$5,892.72*
Sum San Fra			*$104,956.58*	*$41,741.07*
Sum			*$104,956.58*	*$41,741.07*

Figure 4-7 A report design displaying data grouped by sales office and product line

How to group data

1 Select the column that contains the data by which to group.

2 Choose Add Group, as shown in Figure 4-8.

Figure 4-8 Adding a group

If the column you select contains string or numeric data, BIRT Studio groups the data by each unique value in the column. If the column you select contains date-and-time data, you have additional grouping options described in the following section.

Grouping data on a date-and-time column

When you group data on a date-and-time column, you have two options. You can show every individual date or time value, or you can group the data by a specific time interval. The second option is more typical. A shipping report, for example, can organize shipment data by month or by quarter, rather than by date.

Figure 4-9 shows three report designs that contain the same data. The first report design displays data that has not been grouped, the second design groups shipment dates using individual date values, and the third design groups shipment dates by month.

Shipment Date	Order	Shipped To
10 Jan 2003	10100	Online Diecast Creations Co.
11 Jan 2003	10101	Blauer See Auto, Co.
14 Jan 2003	10102	Vitachrome Inc.
1 Feb 2003	10104	Euro+ Shopping Channel
2 Feb 2003	10103	Baane Mini Imports
12 Feb 2003	10105	Danish Wholesale Imports
21 Feb 2003	10106	Rovelli Gifts
26 Feb 2003	10107	Land of Toys Inc.
8 Mar 2003	10108	Cruz & Sons Co.
11 Mar 2003	10109	Motor Mint Distributors Inc.

Shipment Date	Order	Shipped To
Jan 10, 2003		
	10100	Online Diecast Creations Co.
Jan 11, 2003		
	10101	Blauer See Auto, Co.
Jan 14, 2003		
	10102	Vitachrome Inc.
Feb 1, 2003		
	10104	Euro+ Shopping Channel
Feb 2, 2003		
	10103	Baane Mini Imports
Feb 12, 2003		
	10105	Danish Wholesale Imports
Feb 21, 2003		
	10106	Rovelli Gifts
Feb 26, 2003		
	10107	Land of Toys Inc.
Mar 8, 2003		
	10108	Cruz & Sons Co.
Mar 11, 2003		
	10109	Motor Mint Distributors Inc.

Shipment Date	Order	Shipped To
January, 2003		
	10100	Online Diecast Creations Co.
	10101	Blauer See Auto, Co.
	10102	Vitachrome Inc.
February, 2003		
	10103	Baane Mini Imports
	10104	Euro+ Shopping Channel
	10105	Danish Wholesale Imports
	10106	Rovelli Gifts
	10107	Land of Toys Inc.
March, 2003		
	10108	Cruz & Sons Co.
	10109	Motor Mint Distributors Inc.

Figure 4-9 Comparing grouping options for date-and-time data

How to group date-and-time data

1 Select the column that contains the date-and-time values by which to group.

2 Choose Add Group.

3 On Group Details, select one of the group options. If you opt to group by interval:

- Select an interval type, such as weeks, months, or quarters, from the drop-down list.
- Type a number by which to group the selected interval type. For example, if you selected Weeks, type 2 to group data in two-week periods.

Figure 4-10 shows grouping the SHIPPEDDATE column by month.

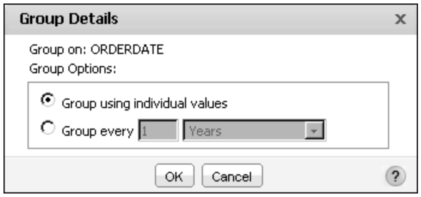


Figure 4-10 Grouping dates by month

Grouping on multiple columns

Just as you do when you sort on multiple columns, consider order of precedence before creating groups. To group customer data by state, then by city, create the groups in that order. For example, the report design in Figure 4-7 groups sales data first by sales office, then by product line. You can identify the order of the data groups by the order of the columns in the table. The first group appears as the first column in the table, the second group appears next to it, and so on.

Changing the grouping order

After you create groups, you can change their order. Consider the effect of changing the grouping order. For example, changing the order of the state and city groups to city and state produces an illogically organized report. On the other hand, changing the order of the sales office and product line groups to product line and sales office provides a different perspective on the sales data. The focus shifts from the sales office performance to the product line performance.

How to change the grouping order

1 Select any column.

2 Right-click the selected column, then choose Column➛Reorder Columns. Reorder Columns, shown in Figure 4-11, displays the grouped columns in the order in which the report design currently groups the data.

3 Under Grouped Columns, select a column and use the up or down buttons to move the column up or down the list. Choose OK.

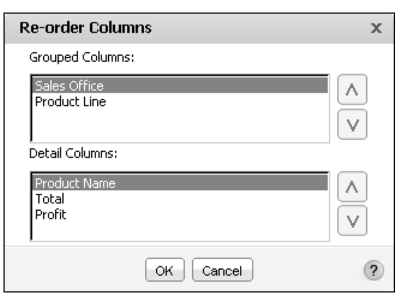


Figure 4-11 Changing the grouping order

Deleting a group

To re-display the values in the detail row, delete a group. Select the column whose values you want to ungroup, then choose Delete Group, as shown in Figure 4-12.

Figure 4-12 Selecting Delete Group

Deleting a group deletes all information associated with the group. For example, if the report design displays aggregate data at the group level, this data is removed.

Organizing data in sections

A section is functionally equivalent to a group. When you create a section, you are also grouping data. Like groups, you can create multiple sections, calculate aggregate data for each section, start each section on a new page, and hide the details of each section.

One of the obvious differences between a section and a group is how the information is arranged. Compare the report designs in Figure 4-13 and Figure 4-14. The report in Figure 4-13 organizes data in two groups: sales office and product line. The report in Figure 4-14 organizes the same data in two sections: sales office and product line.

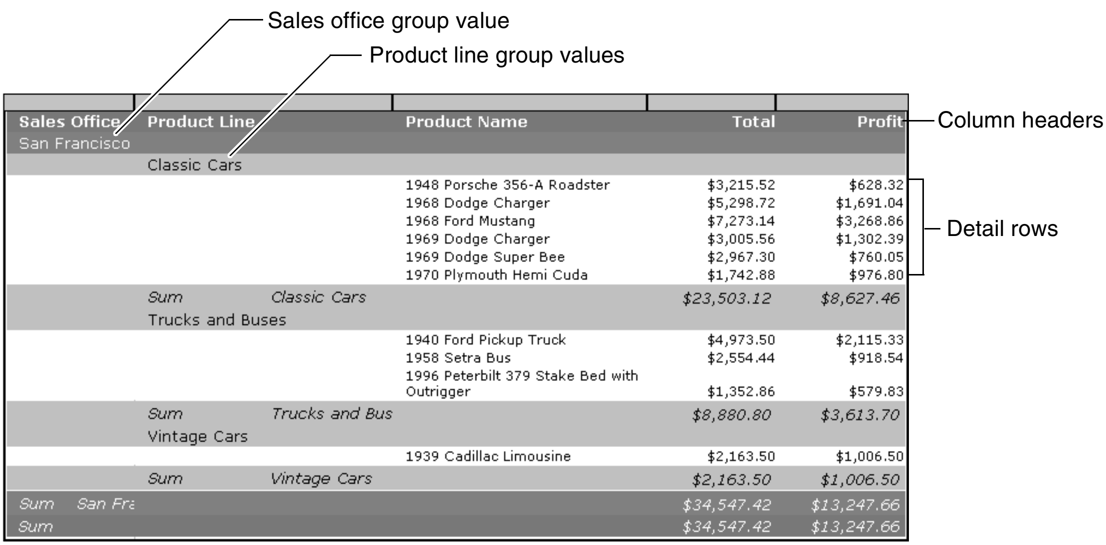


Figure 4-13 Using groups to organize data

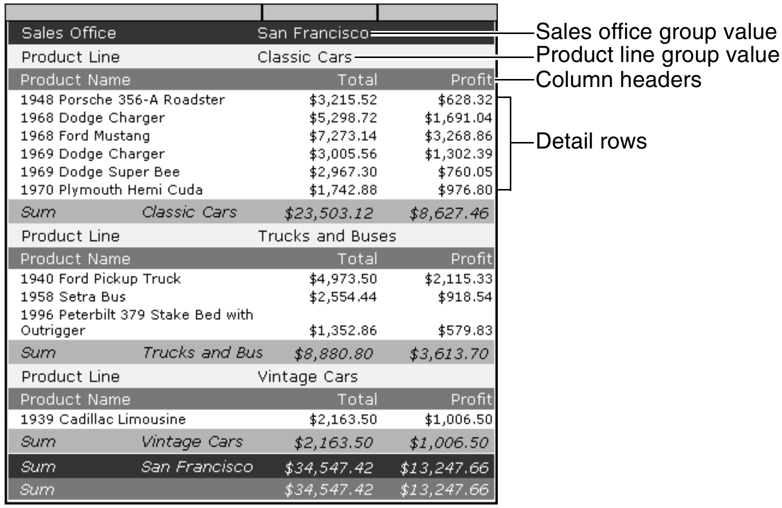


Figure 4-14 Using sections to organize data

In Figure 4-13, the report design with groups displays the data from five fields in five columns. The sales office and product line values appear in the initial columns, below the column headers.

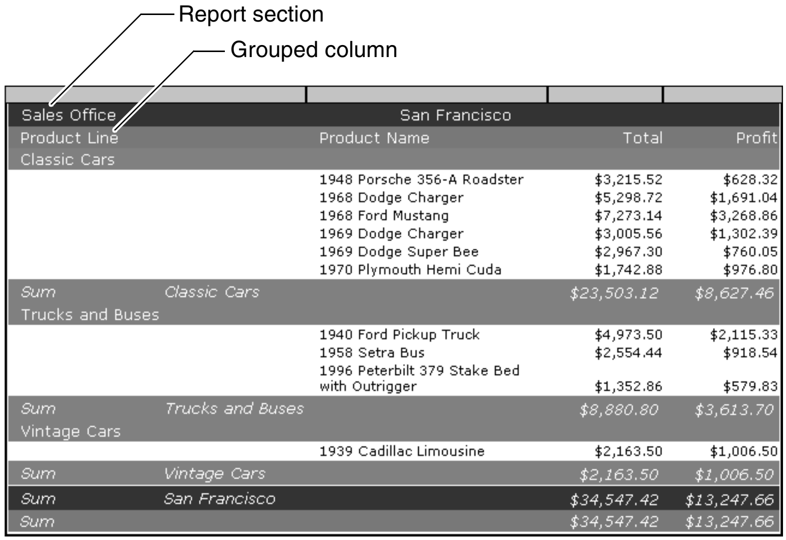
In Figure 4-14, the report design with sections, the sales office and product line values appear above the column headers.

As you can see when you compare the examples, organizing data in sections reduces the overall width of the table. If your report design displays many columns, and space is tight, this difference can be reason enough to create sections instead of groups.

Aside from the differences in the way information is arranged, organizing data in sections provides the following benefits over organizing data in groups:

- You can insert multiple charts, one in each section, to reflect the aggregated data in each section. In a table that contains data groups, you can display only one chart, which presents the aggregated data for the overall table. For more information about charts, see Chapter 8, “Presenting data in a chart.”
- You can display additional data fields in a section heading. By default, a section heading displays only a label and the section value, as shown in the report design in Figure 4-14.

You can use both groups and sections in a single report design. Figure 4-15 shows a report design that contains the same data as the previous examples, except the sales office column is a section, and the product line column is a group.



Sales Office		San Francisco		
Product Line		Product Name	Total	Profit
Classic Cars				
		1948 Porsche 356-A Roadster	$3,215.52	$628.32
		1968 Dodge Charger	$5,298.72	$1,691.04
		1968 Ford Mustang	$7,273.14	$3,268.86
		1969 Dodge Charger	$3,005.56	$1,302.39
		1969 Dodge Super Bee	$2,967.30	$760.05
		1970 Plymouth Hemi Cuda	$1,742.88	$976.80
Sum	*Classic Cars*		*$23,503.12*	*$8,627.46*
Trucks and Buses				
		1940 Ford Pickup Truck	$4,973.50	$2,115.33
		1958 Setra Bus	$2,554.44	$918.54
		1996 Peterbilt 379 Stake Bed with Outrigger	$1,352.86	$579.83
Sum	*Trucks and Buses*		*$8,880.80*	*$3,613.70*
Vintage Cars				
		1939 Cadillac Limousine	$2,163.50	$1,006.50
Sum	*Vintage Cars*		*$2,163.50*	*$1,006.50*
Sum	*San Francisco*		*$34,547.42*	*$13,247.66*
Sum			*$34,547.42*	*$13,247.66*

Figure 4-15 Using a group and a section to organize data

How to create a section

1 Select the column that contains the data you want to organize in a section.

2 Choose Add Section, as shown in Figure 4-16.

Figure 4-16 Adding a section

Just as with groups, if the column you selected contains string or numeric data, BIRT Studio creates a section for each unique value in the column. If the column you selected contains date-and-time data, you can create sections based on individual date values, or you can create sections that group dates by interval.

Displaying additional information in a section heading

As described previously, when you create a section, BIRT Studio creates for each section, a section heading that contains a label and the section value. The label displays the column name. Figure 4-17 shows an example of a section heading.

Figure 4-17 A section heading

You can add the contents of other data fields in a section heading. For example, in the sales office section heading, you can add the address of the sales office or the e-mail of the sales representative. Figure 4-18 shows an example.

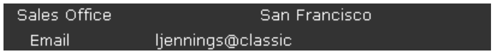


Figure 4-18 A section heading with an additional field

The data fields that you can add to a section heading are the data fields that are currently in use in the table in the report design. If a data field is not in the table, but is included in the data set, you can add the data field to the table. This task is described in Chapter 3, "Inserting calculated data."

How to add content to a section heading

1. Select the section heading.
2. Right-click the section heading, then choose Section Heading. A dialog box displays a list of the data fields that are in the report design.
3. Select the data field to add to the section heading, as shown in Figure 4-19.

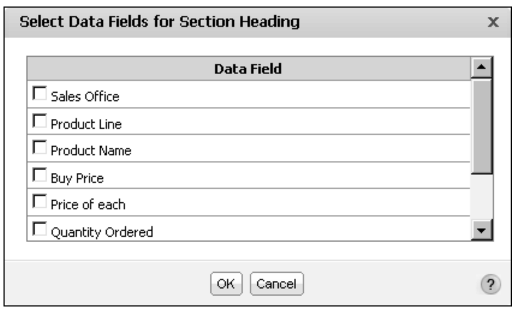


Figure 4-19 Selecting data fields to add to a section heading

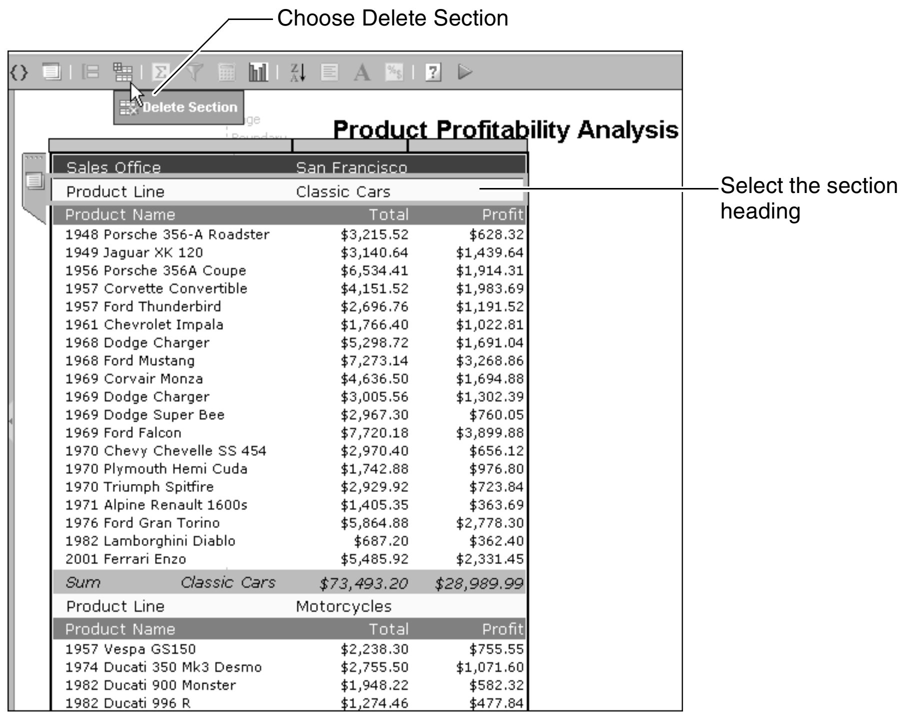
4. Choose OK. The data field appears in the section heading.

Deleting a section

You can delete a section to reorganize data. When you delete a section, the values in the section do not reappear in the detail rows. Instead, the values appear in a grouped column. Any aggregate data for the section is deleted. Any chart you inserted in the section, is also deleted.

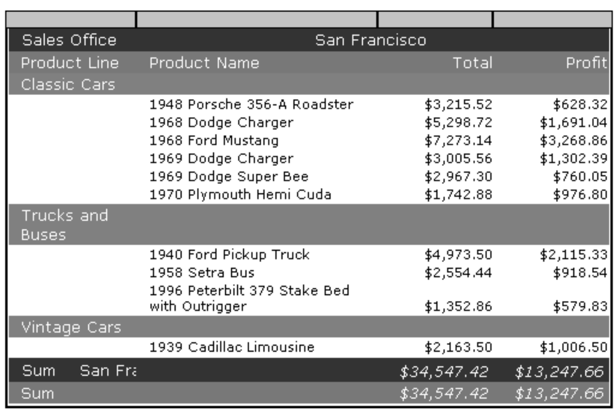
To delete a section, select the section heading, then choose Delete Section, as shown in Figure 4-20.

Figure 4-21 shows the report design after the product line section is deleted from the design in Figure 4-20. The product line section becomes a grouped column.



Sales Office	San Francisco	
Product Line	Classic Cars	
Product Name	Total	Profit
1948 Porsche 356-A Roadster	$3,215.52	$628.32
1949 Jaguar XK 120	$3,140.64	$1,439.64
1956 Porsche 356A Coupe	$6,534.41	$1,914.31
1957 Corvette Convertible	$4,151.52	$1,983.69
1957 Ford Thunderbird	$2,696.76	$1,191.52
1961 Chevrolet Impala	$1,766.40	$1,022.81
1968 Dodge Charger	$5,298.72	$1,691.04
1968 Ford Mustang	$7,273.14	$3,268.86
1969 Corvair Monza	$4,636.50	$1,694.88
1969 Dodge Charger	$3,005.56	$1,302.39
1969 Dodge Super Bee	$2,967.30	$760.05
1969 Ford Falcon	$7,720.18	$3,899.88
1970 Chevy Chevelle SS 454	$2,970.40	$656.12
1970 Plymouth Hemi Cuda	$1,742.88	$976.80
1970 Triumph Spitfire	$2,929.92	$723.84
1971 Alpine Renault 1600s	$1,405.35	$363.69
1976 Ford Gran Torino	$5,864.88	$2,778.30
1982 Lamborghini Diablo	$687.20	$362.40
2001 Ferrari Enzo	$5,485.92	$2,331.45
Sum Classic Cars	*$73,493.20*	*$28,989.99*
Product Line	Motorcycles	
Product Name	Total	Profit
1957 Vespa GS150	$2,238.30	$755.55
1974 Ducati 350 Mk3 Desmo	$2,755.50	$1,071.60
1982 Ducati 900 Monster	$1,948.22	$582.32
1982 Ducati 996 R	$1,274.46	$477.84

Figure 4-20 Deleting a section



Sales Office	San Francisco		
Product Line	Product Name	Total	Profit
Classic Cars			
	1948 Porsche 356-A Roadster	$3,215.52	$628.32
	1968 Dodge Charger	$5,298.72	$1,691.04
	1968 Ford Mustang	$7,273.14	$3,268.86
	1969 Dodge Charger	$3,005.56	$1,302.39
	1969 Dodge Super Bee	$2,967.30	$760.05
	1970 Plymouth Hemi Cuda	$1,742.88	$976.80
Trucks and Buses			
	1940 Ford Pickup Truck	$4,973.50	$2,115.33
	1958 Setra Bus	$2,554.44	$918.54
	1996 Peterbilt 379 Stake Bed with Outrigger	$1,352.86	$579.83
Vintage Cars			
	1939 Cadillac Limousine	$2,163.50	$1,006.50
Sum San Fra		*$34,547.42*	*$13,247.66*
Sum		*$34,547.42*	*$13,247.66*

Figure 4-21 A report design displaying data after a section is deleted

Aggregating data

One of the key features of any report is the ability to display summary, or aggregate, data. Many of the example report designs shown in this document contain aggregate data. The product profitability report design, for example, shows sales and profit subtotals for each product line by sales office, a sales and profit total for each sales office, and grand totals for the company.

Aggregating data involves performing a calculation on a set of values. For a simple listing report that does not organize data in groups or sections, aggregate calculations are performed on values in a selected column, over all the data rows in a table. The listing report design in Figure 4-22 displays aggregate data for the Total and the Profit columns at the end, or footer, of the table.

Product Name	Total	Profit
1939 Cadillac Limousine	$2,163.50	$1,006.50
1940 Ford Pickup Truck	$4,973.50	$2,115.33
1948 Porsche 356-A Roadster	$3,215.52	$628.32
1956 Porsche 356A Coupe	$6,534.41	$1,914.31
1957 Corvette Convertible	$4,151.52	$1,983.69
1957 Vespa GS150	$2,238.30	$755.55
1958 Setra Bus	$2,554.44	$918.54
1961 Chevrolet Impala	$1,766.40	$1,022.81
1968 Dodge Charger	$5,298.72	$1,691.04
1968 Ford Mustang	$7,273.14	$3,268.86
1969 Dodge Charger	$3,005.56	$1,302.39
1969 Dodge Super Bee	$2,967.30	$760.05
1970 Plymouth Hemi Cuda	$1,742.88	$976.80
1971 Alpine Renault 1600s	$1,405.35	$363.69
1974 Ducati 350 Mk3 Desmo	$2,755.50	$1,071.60
1976 Ford Gran Torino	$5,864.88	$2,778.30
1982 Ducati 900 Monster	$1,948.22	$582.32
1982 Ducati 996 R	$1,274.46	$477.84
1982 Lamborghini Diablo	$687.20	$362.40
1996 Peterbilt 379 Stake Bed with Outrigger	$1,352.86	$579.83
Sum	*$63,173.66*	*$24,560.17*
Average	*$3,158.68*	*$1,228.01*
Max	*$7,273.14*	*$3,268.86*

Aggregate data

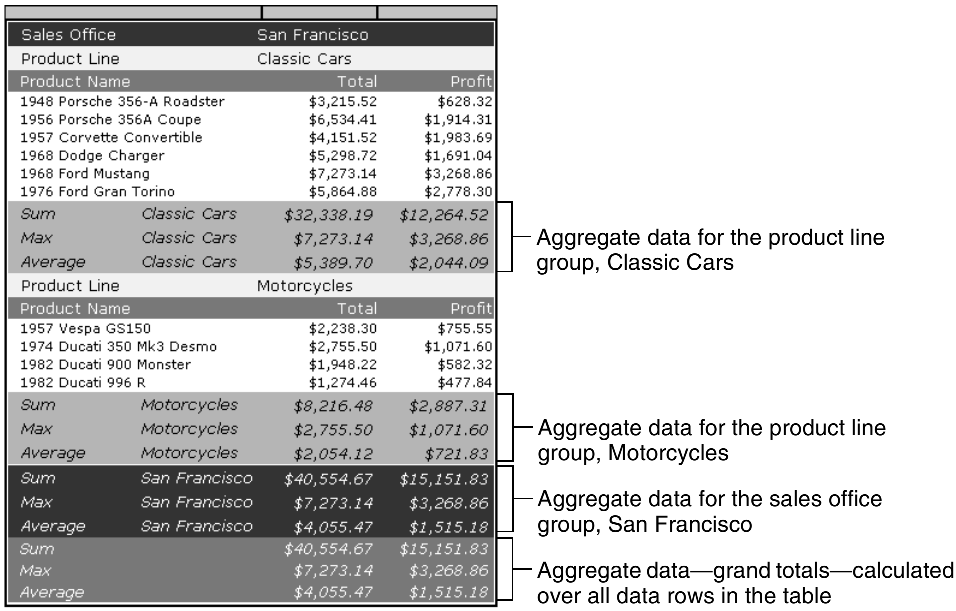
Figure 4-22 Displaying aggregate data in a simple listing report

For a report design that organizes data in groups or sections, as shown in Figure 4-23, you can display aggregates for a selected column or columns, for each group of data rows, and for all the data rows in the table. In this report design, the aggregate data appears in the footer of each group and in the footer of the table.

When you aggregate data in a selected column, you specify the following information:

- The type of aggregate calculation. For both the Total and Profit columns, the report design in Figure 4-23 displays the sum of the values, the highest (max) value, and the average value. For each column, you can display a maximum of three aggregations. Each aggregation appears on a separate row.

- Whether to perform the aggregate calculation over all the data rows in the table, or over the data rows in each group, or both. Aggregating data for groups applies only to report designs that organize data in groups or sections.
- Whether to display the aggregate data in the footer or header of the table or the groups. The previous examples show aggregate data in the footers, which is typical. Select header to display the aggregate data at the beginning of the table or group.



Sales Office	San Francisco		
Product Line	Classic Cars		
Product Name		Total	Profit
1948 Porsche 356-A Roadster		$3,215.52	$628.32
1956 Porsche 356A Coupe		$6,534.41	$1,914.31
1957 Corvette Convertible		$4,151.52	$1,983.69
1968 Dodge Charger		$5,298.72	$1,691.04
1968 Ford Mustang		$7,273.14	$3,268.86
1976 Ford Gran Torino		$5,864.88	$2,778.30
Sum	*Classic Cars*	*$32,338.19*	*$12,264.52*
Max	*Classic Cars*	*$7,273.14*	*$3,268.86*
Average	*Classic Cars*	*$5,389.70*	*$2,044.09*
Product Line	Motorcycles		
Product Name		Total	Profit
1957 Vespa GS150		$2,238.30	$755.55
1974 Ducati 350 Mk3 Desmo		$2,755.50	$1,071.60
1982 Ducati 900 Monster		$1,948.22	$582.32
1982 Ducati 996 R		$1,274.46	$477.84
Sum	*Motorcycles*	*$8,216.48*	*$2,887.31*
Max	*Motorcycles*	*$2,755.50*	*$1,071.60*
Average	*Motorcycles*	*$2,054.12*	*$721.83*
Sum	*San Francisco*	*$40,554.67*	*$15,151.83*
Max	*San Francisco*	*$7,273.14*	*$3,268.86*
Average	*San Francisco*	*$4,055.47*	*$1,515.18*
Sum		*$40,554.67*	*$15,151.83*
Max		*$7,273.14*	*$3,268.86*
Average		*$4,055.47*	*$1,515.18*

Figure 4-23 Displaying aggregate data for groups and sections

The aggregate calculations you can perform on a column depend on the column's data type. You can perform aggregate functions that involve a mathematical calculation, such as Sum and Average, only on numeric data. The most common aggregate functions you perform on string data are Count and Count Value. Count returns the number of values in a column, including duplicate values. Count Value returns the number of distinct values in the column.You can use Count Value on a customer name column to get the number of customers. For date-and-time data columns, you can use the Min and Max functions to get the earliest and the latest date, respectively, from a column displaying order dates. For descriptions of the supported aggregate functions, see Chapter 9, "Functions and operators."

How to aggregate data

1 Select the column that contains the data to aggregate.

2 Choose Aggregation. Aggregation appears, as shown in Figure 4-24.

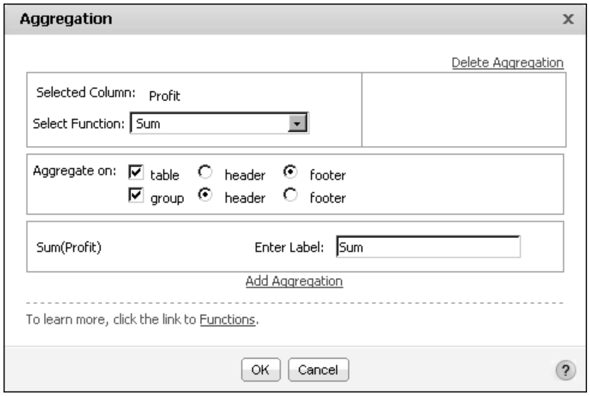


Figure 4-24 Creating an aggregation

3. In Select Function, select the aggregate function to use.
4. In Aggregate on, do the following:
   - Select table to aggregate data over all the data rows in the table. If you select table, select either header or footer as the location in which to display the aggregate data.
   - Select group to aggregate data at the group level. If you select group, select either header or footer as the location in which to display the aggregate data.
5. In Enter Label, type the text to display next to the aggregate value. By default, BIRT Studio uses the name of the function you selected. You can, for example, replace Max with the following text:

   ```
   Highest value
   ```
6. If you want to calculate and display a second aggregation, choose Add Aggregation. Follow steps 3 to 5 to define the next aggregation.
7. Choose OK. Figure 4-25 shows the three aggregations that are defined for the report design in Figure 4-21.

Formatting aggregate values

Using BIRT Studio, you can format aggregate values in the following ways:

- Specify alignment properties.
- Modify font type, size, color, and style.
- Apply a format to the aggregate values.

To format an aggregate value, select the value, and choose a formatting option from the context menu.

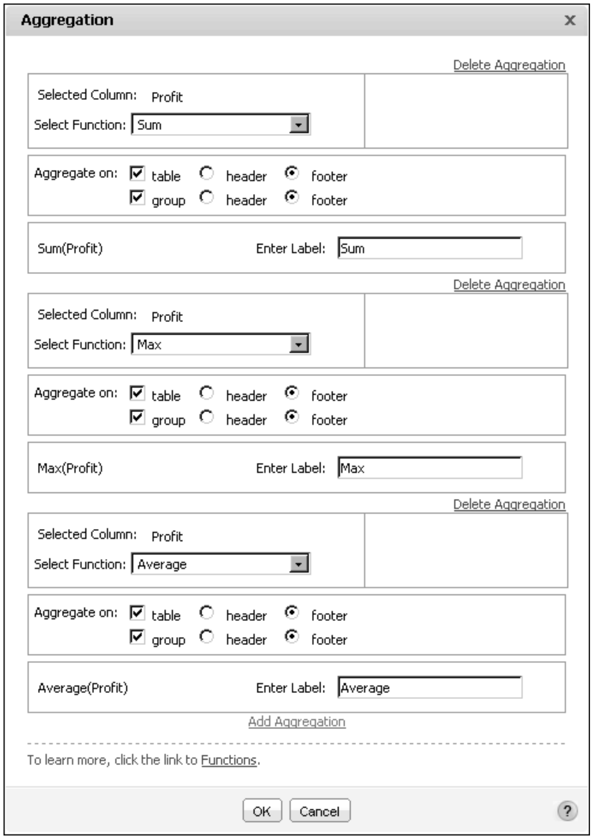


Figure 4-25 Defining three aggregations

Filtering aggregate values

You can use filters to display aggregate values according to certain conditions. To create a filter on aggregate values, complete the following steps:

1 In BIRT Studio, select the aggregate value, then choose Filter➛Filter, or Filter➛Filter Top/Bottom N from the context menu, as shown in Figure 4-26.

2 Define a filter condition. For more information about creating filter conditions, see “Creating a filter” in Chapter 5, “Filtering data.”

BIRT Studio executes the filter condition, and displays the resulting data in the report design.

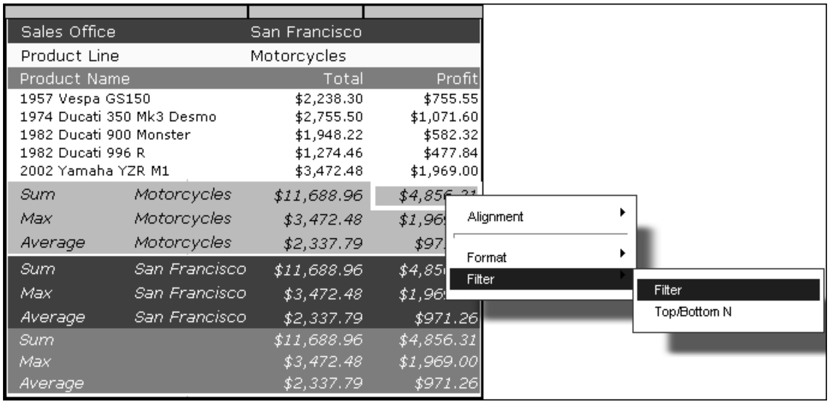


Figure 4-26 Selecting the filter option for an aggregate value

Because BIRT Studio displays only a preview of up to 200 rows of the actual data, when you perform actions such as aggregating data, or filtering aggregate data values, the resulting report design displays different results in BIRT Studio and in Actuate BIRT Viewer. This disparity occurs because BIRT Studio applies the specified aggregation function to the previewed data, and applies the filter condition to this subset of data.

When you view a report created in BIRT Studio, in the BIRT Viewers, all data rows are retrieved and displayed in the report design. If you now create an aggregate calculation in Interactive Viewer, the aggregate function is applied across all rows of data in the report design. If you create a filter condition for the aggregate data, the filter is also applied across all rows of data, resulting in more accurate results. Always view the report design in BIRT viewer to verify that the results are what you expect.

Hiding details

When you create a report that organizes data in groups or sections, you can change the appearance of the report to a summary report by hiding the details of a group or section. Hiding details, particularly for a report that runs into hundreds of pages, shows key information at a glance. For reports containing large amounts of aggregate data, you can also choose to create a summary report. A summary report aggregates data at the data-source level, providing a concise view of report data, improving response time, and reducing the load on the server. For more information, see Chapter 7, “Working with summary tables.”

Figure 4-27 shows two report designs that contain the same data. The first design shows all details. The next design hides the details in each product-line section.

Sales Office	San Francisco		
Product Line	Classic Cars		
Product Name		Total	Profit
1956 Porsche 356A Coupe		$6,534.41	$1,914.31
1957 Corvette Convertible		$4,151.52	$1,983.69
1968 Dodge Charger		$5,298.72	$1,691.04
1968 Ford Mustang		$7,273.14	$3,268.86
1976 Ford Gran Torino		$5,864.88	$2,778.30
Sum	*Classic Cars*	*$29,122.67*	*$11,636.20*
Max	*Classic Cars*	*$7,273.14*	*$3,268.86*
Average	*Classic Cars*	*$5,824.53*	*$2,327.24*
Product Line	Motorcycles		
Product Name		Total	Profit
1957 Vespa GS150		$2,238.30	$755.55
1974 Ducati 350 Mk3 Desmo		$2,755.50	$1,071.60
1982 Ducati 900 Monster		$1,948.22	$582.32
1982 Ducati 996 R		$1,274.46	$477.84
Sum	*Motorcycles*	*$8,216.48*	*$2,887.31*
Max	*Motorcycles*	*$2,755.50*	*$1,071.60*
Average	*Motorcycles*	*$2,054.12*	*$721.83*
Product Line	Trucks and Buses		
Product Name		Total	Profit
1940 Ford Pickup Truck		$4,973.50	$2,115.33
1958 Setra Bus		$2,554.44	$918.54
1996 Peterbilt 379 Stake Bed with Outrigger		$1,352.86	$579.83
Sum	*Trucks and Bus*	*$8,880.80*	*$3,613.70*
Max	*Trucks and Bus*	*$4,973.50*	*$2,115.33*
Average	*Trucks and Bus*	*$2,960.27*	*$1,204.57*
Sum	*San Francisco*	*$46,219.95*	*$18,137.21*
Max	*San Francisco*	*$7,273.14*	*$3,268.86*
Average	*San Francisco*	*$3,851.66*	*$1,511.43*
Sum		*$46,219.95*	*$18,137.21*
Max		*$7,273.14*	*$3,268.86*
Average		*$3,851.66*	*$1,511.43*

Sales Office	San Francisco		
Product Line	Classic Cars		
Product Name		Total	Profit
Sum	*Classic Cars*	*$29,122.67*	*$11,636.20*
Max	*Classic Cars*	*$7,273.14*	*$3,268.86*
Average	*Classic Cars*	*$5,824.53*	*$2,327.24*
Product Line	Motorcycles		
Product Name		Total	Profit
Sum	*Motorcycles*	*$8,216.48*	*$2,887.31*
Max	*Motorcycles*	*$2,755.50*	*$1,071.60*
Average	*Motorcycles*	*$2,054.12*	*$721.83*
Product Line	Trucks and Buses		
Product Name		Total	Profit
Sum	*Trucks and Bus*	*$8,880.80*	*$3,613.70*
Max	*Trucks and Bus*	*$4,973.50*	*$2,115.33*
Average	*Trucks and Bus*	*$2,960.27*	*$1,204.57*
Sum	*San Francisco*	*$46,219.95*	*$18,137.21*
Max	*San Francisco*	*$7,273.14*	*$3,268.86*
Average	*San Francisco*	*$3,851.66*	*$1,511.43*
Sum		*$46,219.95*	*$18,137.21*
Max		*$7,273.14*	*$3,268.86*
Average		*$3,851.66*	*$1,511.43*

Figure 4-27 Showing and hiding detail rows

How to hide the details of a group

1. Select the grouped column for which to hide details.
2. Right-click. Then, choose Group➛Hide Detail, as shown in Figure 4-28.

   In the example shown in Figure 4-28, all the detail rows for every product line group are hidden.

How to hide the details of a section

1. Select the section heading of the section for which to hide details.
2. Right-click. Then, choose Section➛Hide Detail, as shown in Figure 4-29.

How to re-display the details of a group or section

Select the grouped column or section heading, right-click the object, then choose Group➛Show Detail or Section➛Show Detail.

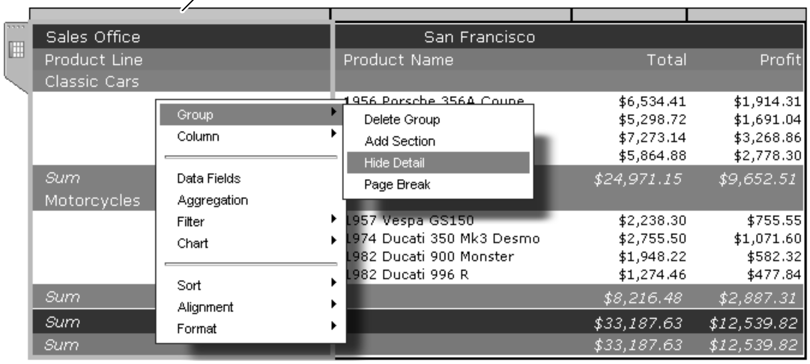


Figure 4-28 Hiding the details of a selected grouped column

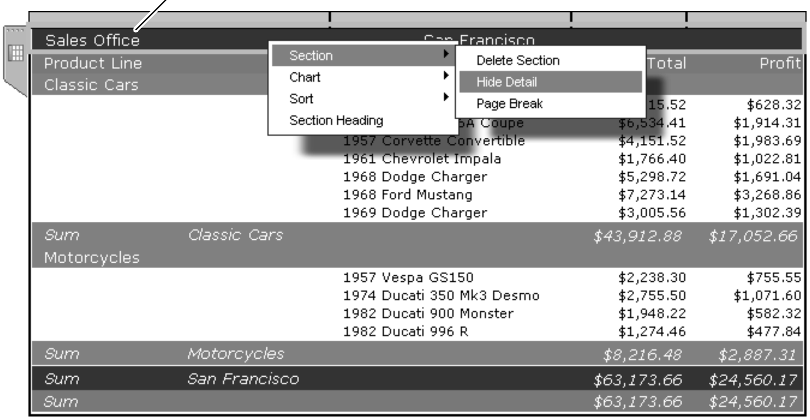


Figure 4-29 Hiding the details of a selected section

Starting each group or section on a new page

Lengthy reports or reports that consist of distinct sections typically look more organized if each section appears on a separate page. The product profitability report design example appears compact, because it is previewed with a small number of rows. For example, the report design in Figure 4-29 is set to display only 20 rows of data. When the report is run and appears in the viewer, it contains 60 pages of data.

As you design a report, make sure you periodically view the generated report to see if you need to set page breaks to make the report more usable. You can set a page break for each group and section you create. In the report design in

Figure 4-29, you can start each sales office section and each product line group within the sales office section on a new page.

How to set a page break

1. Select a grouped column or section heading. Right-click the object, then choose Group➛Page Break or Section➛Page Break. Page Break appears, as shown in Figure 4-30.

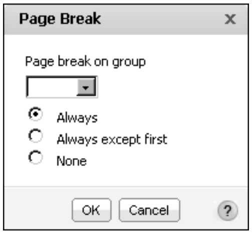


Figure 4-30 Page Break

2. On Page Break, select After or Before from the drop-down list to insert a page break after or before a group or section.
   - If you select After, do one of the following:
     - Choose Always to always insert a page break after each group or section.
     - Choose Always except last to always insert a page break after each group or section, but not after the last one. This is the typical option to avoid a blank page at the end of the report.
   - If you select Before, do one of the following:
     - Choose Always to always insert a page break before each group or section.
     - Choose Always except first to always insert a page break before each group or section, but not before the first one. This is the typical option to avoid a blank page at the beginning of the report.
   - Choose None to remove an existing page break.
3. Choose OK. Run the report to view the modified design.

Chapter

5

Filtering data

This chapter contains the following topics:

- About data filtering
- About filtering options
- Creating a filter
- Prompting for filter values at run time

About data filtering

A data set often provides more information than your report needs. You can select specific data rows to use in a report by using a filter. For example, rather than list all customer sales, you can create a filter to select only the sales data for a particular week or only the sales data for a particular region.

Filtering data helps you work effectively with large amounts of data. It enables you to find the necessary pieces of information to answer specific business questions, such as which sales representatives generated the top ten sales accounts, which products generated the highest profit in the last quarter, which customers have not made a purchase in the past 90 days, and so on.

Filtering data can also have a positive effect on processing speed. Limiting the number of data rows used in the report can reduce measurably the load on the database server, because it does not need to return all the rows every time the report is run.

Actuate BIRT Studio supports the use of static filters and dynamic filters. You can use a static filter to define a specific filter condition during report design. The data displayed in the report depends on the filter condition specified in the report design. You use a dynamic filter to prompt a user to specify values for which BIRT Studio displays data, when they run the report. The data that appears in the report depends on the values the user specifies when running the report.

When you use filters in a report design, BIRT Studio typically evaluates the filter condition against the rows of data previewed in the report design, 50 rows by default. Increase this number to 200 rows, when creating filters in a report design. If you use the default value of 50 data rows to test the filter condition, BIRT Studio might frequently not be able to return any data that matches the filter you define. Also run the report in BIRT Viewer to view the actual data, and verify that the result is what you expect.

About filtering options

You can filter data at the data-set level and at the table level. Filtering at the data-set level narrows the scope of data available to a report design and can improve design-time performance if the data set returns a particularly large amount of data. Filtering at the table level narrows the scope of data displayed in a table and is the typical filtering option.

If you filter data at both the data set and table levels, BIRT Studio executes the filter at the data-set level first, then at the table level. Design the filters accordingly. For example, if a data set returns all customer sales for ten years, you can create a filter at the data-set level to limit the sales data to the current year. Then, when you design your report, you can create a filter at the table level to display sales data for a particular quarter in the current year. You cannot display sales data for the previous year, because the filter at the data-set level excludes this data.

Creating a filter

When you create a filter, you define a condition that specifies which data rows to include in the report. A filter condition is an If expression that must evaluate to true in order to include a data row. For example:

```
If the order total is greater than 10000
If the sales office is San Francisco
If the order date is between 4/1/2010 and 6/30/2010
```

Figure 5-1 shows an example of a condition defined in Filter. As the illustration shows, Filter helps you define the condition by breaking it down to the following parts:

- The column to evaluate, such as Total.
- The comparison operator that specifies the type of comparison test, such as Greater Than. For information about all the operators, see Chapter 9, "Functions and operators."
- The value to which all values in the column are compared, such as 10000.

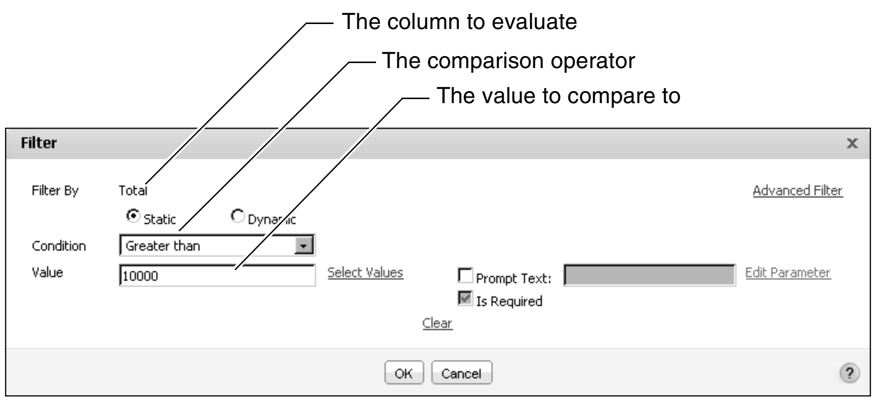


Figure 5-1 Filter displaying a filter condition

Defining a filter at the report-table level

This section describes the procedure to create a filter at the table level. You can create a simple filter as described in the previous section. In addition, you can also choose multiple values for a filter condition by using the In operator, choose to exclude specific data, create a filter on empty values in the data fields, display the top or bottom N values, or specify a date as the comparison value. You can create a filter condition that compares values in a column containing string data to a string pattern instead of to a specific value. You can also create a filter condition that compares values in one column to values in another column.

How to create a filter at the table level

1 Select the column in the report design, containing the value that determines when the filter takes effect. For example, to create a filter that retrieves data rows where the sales office is Boston, select the sales office column.

2 Choose Filter. Filter appears, as shown in Figure 5-2.

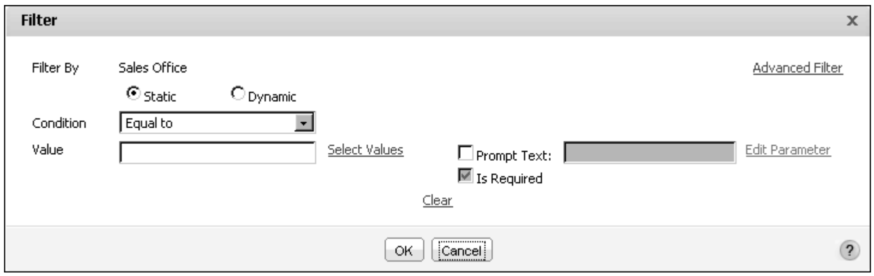


Figure 5-2 Filter displaying the selected column

Filter By displays the column on which the filter is applied.

3 Select Static, as the type of filter, if necessary.

4 In Condition, select the comparison test, or operator, to apply to the selected column. Depending on the operator you select, Filter displays one or two Value fields, or a completed filter condition.

5 If you select an operator that requires a comparison value, you can specify the value in one of the following ways:

- In Value, type the value.
- To select from a list of values, choose Select Values. A list of values appears. Select a value from the list. Figure 5-3 shows the selection of Boston from the list of available sales office values.

  If the list of values is long, type a value in Filter Text and choose Find. If found, BIRT Studio displays this value in the list. When you select the value, it appears in Value.

6 Choose OK. The filter takes effect. In some cases, the sample data in the report design does not meet the filter condition. When this happens, the report design does not show any data. The best way to test a filter is to run the report and view the generated report.

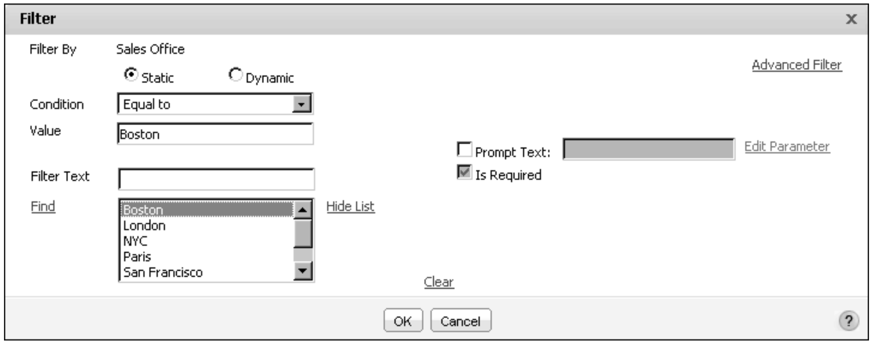


Figure 5-3 Specifying values

BIRT Studio displays the message in Figure 5-4.

Figure 5-4 The message displayed when the sample data in the report design does not meet the filter condition

Selecting multiple values for a filter condition

The preceding filter examples specify one comparison value. Sometimes you need to view more data, for example, sales details for several sales offices, not for only one office. To select more than one comparison value, select the In operator, choose Select Values, then select each value. To select multiple values, press Ctrl as you select each value. To select contiguous values, select the first value, press Shift, and select the last value. This action selects the first and last values and all values between them. Figure 5-5 shows London and Paris selected from a list of sales office values.

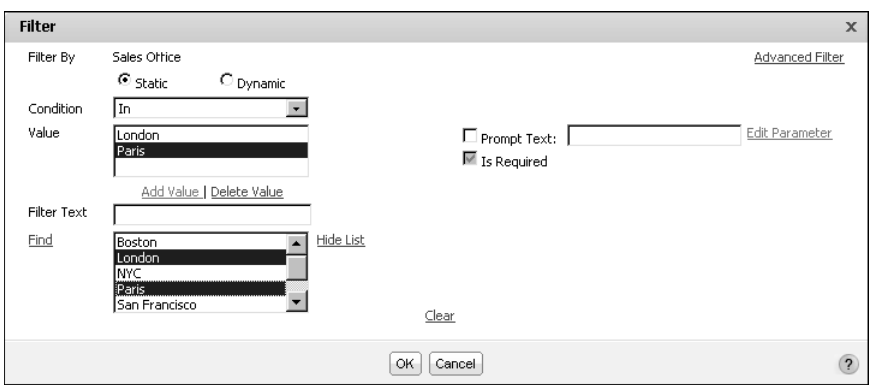


Figure 5-5 Selecting multiple comparison values

Excluding data

You use comparison operators, such as Equal to, Greater Than, or Less Than, to evaluate the filter condition to determine which data to include. Sometimes it is more efficient to specify a condition that excludes a small set of data. For example, you need sales data for all countries except the USA. Instead of selecting all the available countries and listing them in the filter condition, simply use the Not Like operator. Similarly, use Not Between to exclude data in a specific range, and Not Like to exclude data that matches a string pattern.

For example, the following filter condition excludes orders with amounts between 1000 and 5000:

```
OrderAmount Not Between 1000 and 5000
```

The filter condition in the next example excludes products with codes that start with MS:

```
ProductCode Not Like 'MS%'
```

Filtering empty or blank values

Sometimes, rows display nothing for a particular column. For example, suppose a customer database table contains an e-mail field. Some customers, however, do not supply an e-mail address. In this case, the e-mail field might contain an empty value or a blank value. An empty value, also called a null value, means no value is supplied. Null values apply to all data types. You can create a filter to exclude data rows where a particular column has null values.

When filtering to exclude null values, use the Is Not Null operator. To view only rows that have null values in a particular column, use Is Null. For example, the following filter condition excludes customer data where the e-mail column contains null values:

```
email Is Not Null
```

The following filter condition displays only rows where the e-mail column contains null values:

```
email Is Null
```

Displaying top or bottom values

For a report that presents a large amount of numeric data, you might find it useful to view, for example, the top 50 order totals or the counties whose median home prices are in the bottom 10 percent. To display top or bottom values, use the Top N, Top Percent, Bottom N, and Bottom Percent operators.

For example, the following filter condition displays the top 50 orders:

```
OrderAmount Top N 50
```

The filter condition in the next example displays the median prices in the bottom 10 percent. If there are 1000 rows of data, 100 rows appear.

```
MedianPrice Bottom Percent 10
```

You can display top or bottom values for the entire report table, or for each group in the report design. For example, in a report design that lists profit and revenue for each product line and sales office, you can define a filter condition that displays profit values in the bottom 10 percent for each sales office city. On Filter, define the condition in the same way you did earlier. In Filter on, select On Group, then select a grouped column from the drop-down list, as shown in Figure 5-6.

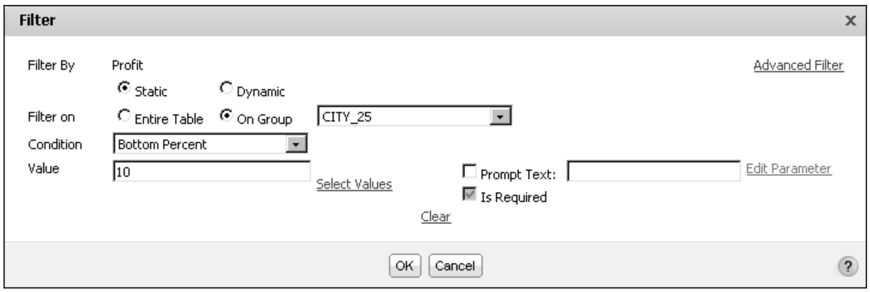
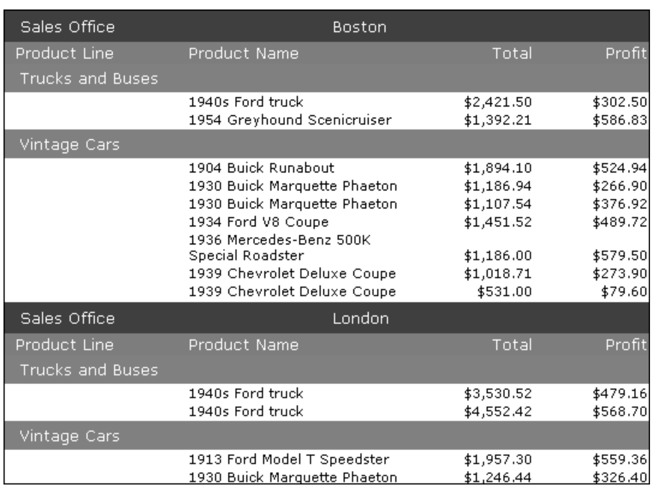


Figure 5-6 Specifying a bottom percent filter condition for a group

Choose OK. The report design displays the data rows that match the filter condition you defined, as shown in Figure 5-7.



Sales Office	Boston		
Product Line	Product Name	Total	Profit
Trucks and Buses			
	1940s Ford truck	$2,421.50	$302.50
	1954 Greyhound Scenicruiser	$1,392.21	$586.83
Vintage Cars			
	1904 Buick Runabout	$1,894.10	$524.94
	1930 Buick Marquette Phaeton	$1,186.94	$266.90
	1930 Buick Marquette Phaeton	$1,107.54	$376.92
	1934 Ford V8 Coupe	$1,451.52	$489.72
	1936 Mercedes-Benz 500K Special Roadster	$1,186.00	$579.50
	1939 Chevrolet Deluxe Coupe	$1,018.71	$273.90
	1939 Chevrolet Deluxe Coupe	$531.00	$79.60
Sales Office	London		
Product Line	Product Name	Total	Profit
Trucks and Buses			
	1940s Ford truck	$3,530.52	$479.16
	1940s Ford truck	$4,552.42	$568.70
Vintage Cars			
	1913 Ford Model T Speedster	$1,957.30	$559.36
	1930 Buick Marquette Phaeton	$1,246.44	$326.40

Figure 5-7 Displaying profits in the bottom 10 percent for each city

Specifying a date as a comparison value

When you create a filter condition that compares the date-and-time values in a column to a specific date, the date value you supply must be in one of the following formats, regardless of your locale:

```
3/26/2008
3/26/2008 2:30:00 PM
```

For your convenience, BIRT Studio provides a calendar you can use to select a date. Figure 5-8 shows the calendar and how to access it.

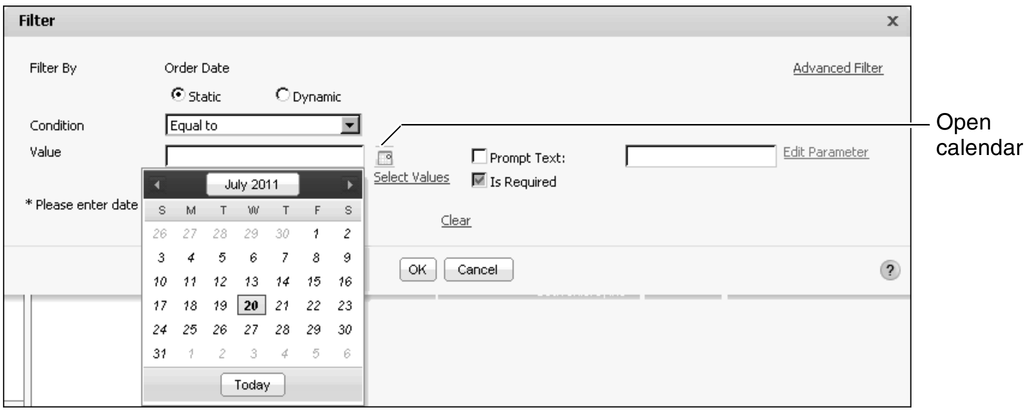


Figure 5-8 Selecting a date using the calendar

Choose the calendar icon to open the calendar, as shown in Figure 5-8. The calendar appears displaying the current date, month, and year. The current date is highlighted by default.

- To accept the current date, choose Today.
- To choose another date, select it.

The selected value appears in Value, in Filter. Choose OK for the filter condition to take effect.

Comparing to a string pattern

For a column that contains string data, you can create a filter condition that compares each value to a string pattern instead of to a specific value. For example, to display only customers whose names start with M, use the Like operator and specify the string pattern, M%, as shown in the following filter condition:

```
Customer Like M%
```

You can use the following special characters in a string pattern:

- % matches zero or more characters. For example, %ace% matches any value that contains the string ace, such as Ace Corporation, Facebook, Kennedy Space Center, and MySpace.
- _ matches exactly one character. For example, t_n matches tan, ten, tin, and ton. It does not match teen or tn.

To match the percent sign (%) or the underscore character (_) in a string, precede those characters with two backslash characters (\\). For example, to match S_10, use the following string pattern:

```
S\\_10
```

To match 50%, use the following string pattern:

```
50\\%
```

Comparing to a value in another column

Use a filter condition to compare the values in one column with the values of another column. For example, in a report that displays products, sale prices (Price), and MSRP (Manufacturer Suggested Retail Price), as shown in Figure 5-9, you can create a filter condition to compare the Price and MSRP of each product. You can then create a filter to display rows where the Price is greater than MSRP.

How to compare to a value in another column

1 Select the column that contains the values to compare, then choose Filter.

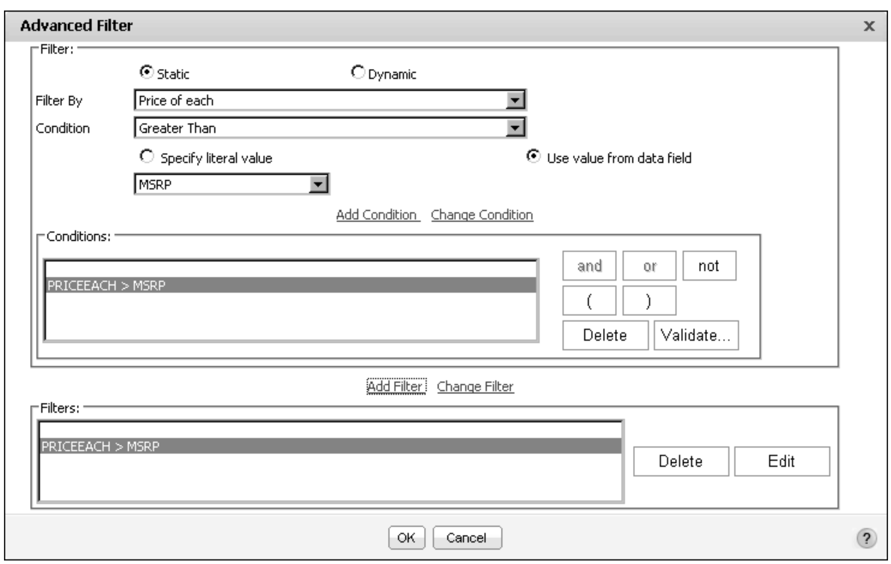


Figure 5-9 Building a comparison filter using Advanced Filter

2 On Filter, choose Advanced Filter.

3 In Condition, select a comparison operator.

4 Choose Use value from data field.

5 From the drop-down list that appears, choose the column that contains the values to compare to the first column, as shown in Figure 5-9.

6 Choose Add Condition.

7 Choose Add Filter, then choose OK.

Defining multiple filter conditions

When you create a filter, you define at least one filter condition, but there is no limit on the number of filter conditions that you can add. Each condition you add narrows the scope of data further.

For example, you can create a filter that returns rows where the customer's credit rank is either A or B and whose open orders total between $250,000 and $500,000. Each condition you add also adds complexity to the filter. Design and test filters with multiple conditions carefully.

Adding a condition

There are two ways to define multiple filter conditions. You can select a column and define a filter, then select the next column and define a filter, and so on. Figure 5-10 shows the definition of two filters, the first on the SALES OFFICE column, the second on the PRODUCTLINE column.

When you use this method to define multiple filters, BIRT Studio constructs the following filter expression:

```
SALES OFFICE = San Francisco
and PRODUCTLINE = Vintage Cars
```

This filter returns only data rows that meet both conditions. Sometimes, you need to create a filter to return data rows when either condition is true, or you need to create a more complex filter. To accomplish either task, use Advanced Filter, shown in Figure 5-11. As the illustration shows, this dialog box provides more options for defining a filter with multiple conditions. It also shows all the filter conditions for the table.

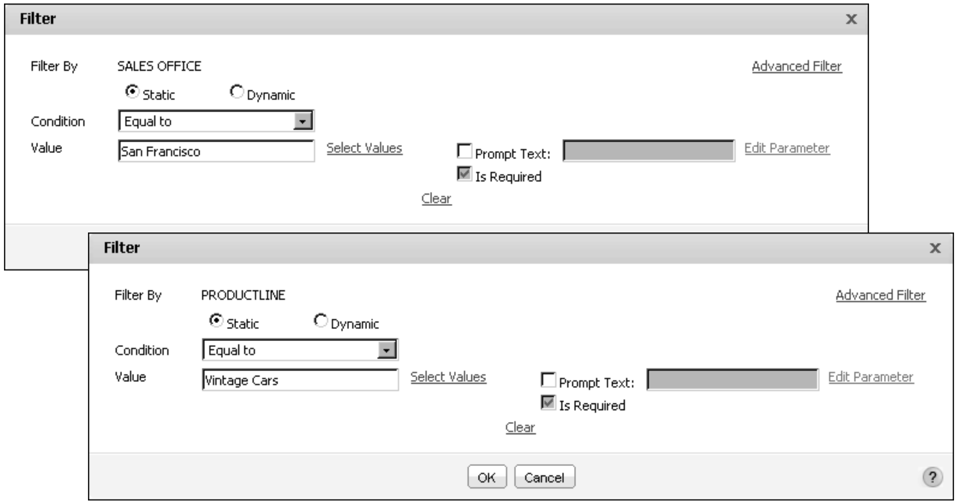


Figure 5-10 Creating multiple filter conditions

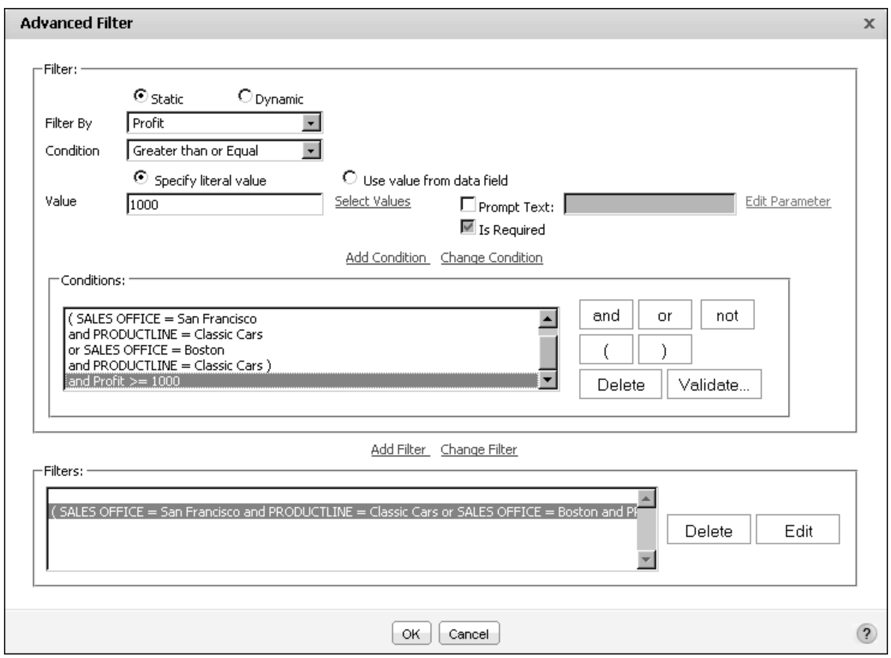


Figure 5-11 Displaying multiple conditions in a complex filter

How to define multiple filter conditions

This procedure shows how to define multiple filter conditions using Advanced Filter:

1. In BIRT Studio, select the column that contains the values to filter by, then choose Filter.
2. On Filter, choose Advanced Filter.
3. On Advanced Filter, select Static, if necessary.

   Advanced Filter displays the selected column in Filter By. You can change this column by selecting a different column from the drop-down list.
4. Define a filter condition as follows:
   1. In Condition, select a comparison operator.
   2. Specify a value as follows:
      - To type or choose a specific value, choose Specify a literal value.
        - In Value, type the value, then proceed to step 3.
        - To select from a list of values, choose Select Values. A list of values appears. Select a value from the list, then proceed to step 3.
        - If the list of values is long, type a value in Filter Text, then choose Find. If found, BIRT Studio highlights this value in the list. Proceed to step 3.
      - To compare values in the selected column, to values in another column, choose Use value from data field. Select a column from the drop-down list.
   3. Choose Add Condition.

   The filter condition appears in the Conditions area.
5. Define the next filter condition on Advanced filter as follows:
   1. In Filter By, select another column.
   2. In Condition, select a comparison operator.
   3. Specify a value.
      - To type or choose a specific value, choose Specify a literal value.
        - In Value, type the value. Proceed to step 4.
        - To select from a list of values, choose Select Values. A list of values appears. Select a value from the list. Proceed to step 4.
        - If the list of values is long, type a value in Filter Text, then choose Find. If found, BIRT Studio highlights this value in the list. Proceed to step 4.
      - If you choose Use value from data field, select a column from the drop-down list.

4 Choose Add Condition.

In the Conditions area, the second filter condition appears after the first condition, as shown in Figure 5-12. By default, the second condition is preceded by the logical operator, And.

6 Select a different logical operator, if necessary.

7 To add additional filter conditions repeat step 4.

8 If you create more than two filter conditions and you use different logical operators, you can use the parentheses buttons to group conditions to determine the order in which conditions are evaluated.

9 Choose Validate to verify the syntax of the filter conditions, then choose Add Filter.

The defined filter conditions appear in the Filters area, as shown in Figure 5-12. Choose OK.

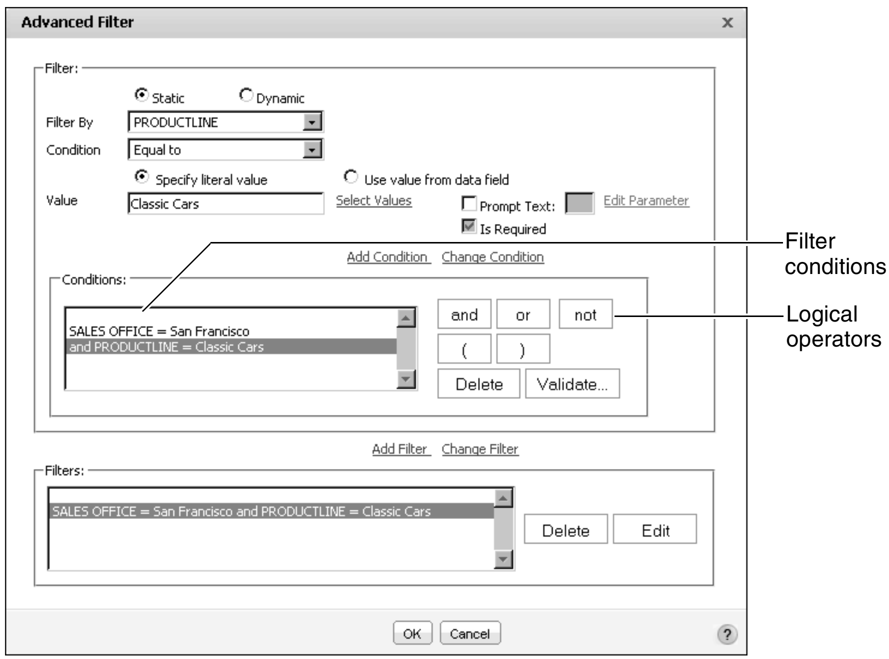


Figure 5-12 Advanced Filter displaying two conditions

10 Run the report to verify that it displays the expected results.

Selecting a logical operator

As you add each filter condition, the logical operator And is inserted between each filter condition. You can change the operator to Or. The And operator means both filter conditions must be true to include a data row in the report. The Or operator means only one condition has to be true to include a data row. You also can add the Not operator to either the And or Or operators to exclude a small set of data.

For example, the following filter conditions return only sales data for classic car items for the San Francisco office:

```
Sales Office = San Francisco
And Product Line = Classic Cars
```

The following filter conditions return all sales data for the San Francisco and Boston offices:

```
Sales Office = San Francisco
Or Sales Office = Boston
```

The following filter conditions return sales data for all product lines, except classic cars, for the San Francisco office:

```
Sales Office = San Francisco
And Not (Product Line = Classic Cars)
```

Specifying the evaluation order

BIRT Studio evaluates filter conditions in the order in which they appear. If you define more than two conditions, you can use parentheses to group conditions to get the results you expect. For example, A And B Or C is evaluated in that order, so A and B must be true or C must be true to include a data row. In A And (B Or C), B Or C is evaluated first, so A must be true and B Or C must be true to include a data row. To illustrate the difference a pair of parentheses can make, compare the following examples.

The following filter contains four conditions and none of the conditions are grouped:

```
Sales Office = San Francisco
and ProductLine = Classic Cars
and Total >= 8000
or Profit >= 4500
```

Figure 5-13 shows the generated report. Although the filter specifies the San Francisco office and the Classic Cars product line, the report displays data for other sales offices and product lines. Without any grouped conditions, the filter includes rows that meet either conditions 1, 2, and 3 or just condition 4.

Sales Office	Product Line	Product Name	Total	Profit
Paris				
	Classic Cars			
		1952 Alpine Renault 1300	$10,072.00	$5,143.00
		1952 Alpine Renault 1300	$9,467.68	$4,834.42
		1969 Dodge Charger	$11,170.52	$5,473.71
		1992 Ferrari 360 Spider red	$9,449.40	$4,775.40
		2001 Ferrari Enzo	$8,434.52	$4,515.33
		2001 Ferrari Enzo	$9,974.40	$5,386.08
	Vintage Cars			
		1928 Mercedes-Benz SSK	$8,353.00	$4,725.00
San Francisco				
	Classic Cars			
		1952 Alpine Renault 1300	$10,286.40	$5,554.56
		1972 Alfa Romeo GTA	$8,616.96	$3,133.44
	Motorcycles			
		2003 Harley-Davidson Eagle Drag Bike	$9,394.28	$4,934.30

Figure 5-13 A report displaying the results of a complex filter that does not use parentheses to group conditions

The following filter contains the same four conditions, but this time the third and fourth conditions are grouped:

```
Sales Office = San Francisco
and ProductLine = Classic Cars
and (Total >= 8000
or Profit >= 4500)
```

Figure 5-14 shows the generated report. This time, only two rows meet the conditions. The Sales Office = San Francisco and ProductLine = Classic Cars conditions must be true, and either the Total >= 8000 condition or the Profit >= 4500 condition is true.

Sales Office	Product Line	Product Name	Total	Profit
San Francisco				
	Classic Cars			
		1952 Alpine Renault 1300	$10,286.40	$5,554.56
		1972 Alfa Romeo GTA	$8,616.96	$3,133.44

Figure 5-14 A report displaying the results of a complex filter that uses parentheses to group conditions

Changing a condition

To change a filter condition, in the Conditions area of Advanced Filter, select the condition. Modify the condition by changing the values in Filter By, Condition, or Value. Select Change Condition, then select Change Filter, and choose OK, as shown in Figure 5-15.

Deleting a filter condition

To delete a filter condition, in the Filters area of Advanced Filter, select the Filter. Then, select Delete. Then, select the Condition and select Delete. Choose OK. Verify that any remaining filter conditions display the expected results.

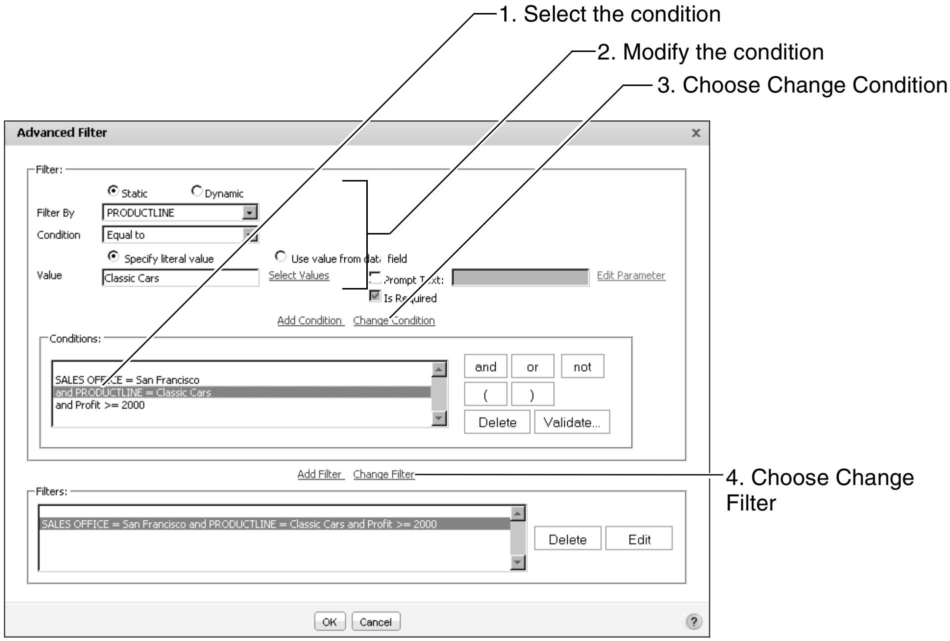


Figure 5-15 Changing a filter condition in Advanced Filter

Defining a filter at the data-set level

When you use a template containing a data set, an information object, or a data object data source, the procedure to create a filter at the data-set level is similar to the process to create a filter in Advanced Filter. The primary difference is how you access Advanced Filter.

How to create a data-set filter using Switch Filter Pane

1 In Available Data, choose Filter, as shown in Figure 5-16.

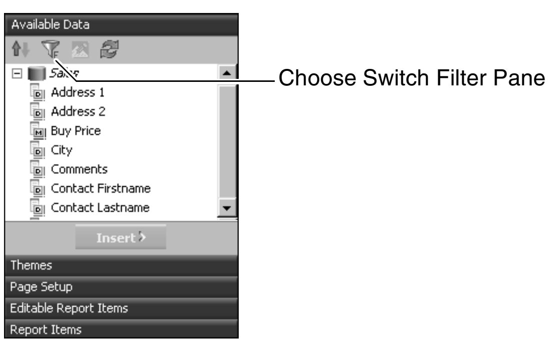


Figure 5-16 Choosing Filter

Filter Pane appears below the report design. It is empty, as shown in Figure 5-17, if you have not defined any filters at the data-set level.

2 Choose Add to create a new filter. Advanced Filter appears.

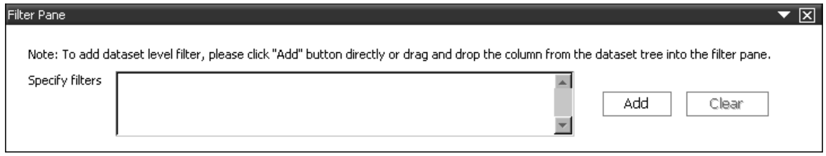


Figure 5-17 Filter Pane

3 Define the filter condition. For more information about how to define a filter condition in Advanced Filter, see “How to define multiple filter conditions,” earlier in this chapter.

How to create a data-set filter using Table Builder

1 Create a new report using a data set in a template, an information object or a data object data source.

2 On Table Builder—Data, select the data fields to use in the report design.

3 Choose Filter. On Table Builder—Filter, shown in Figure 5-18, to create a filter condition, complete the following steps:

   1 In Filter By, select a column from the drop-down list.

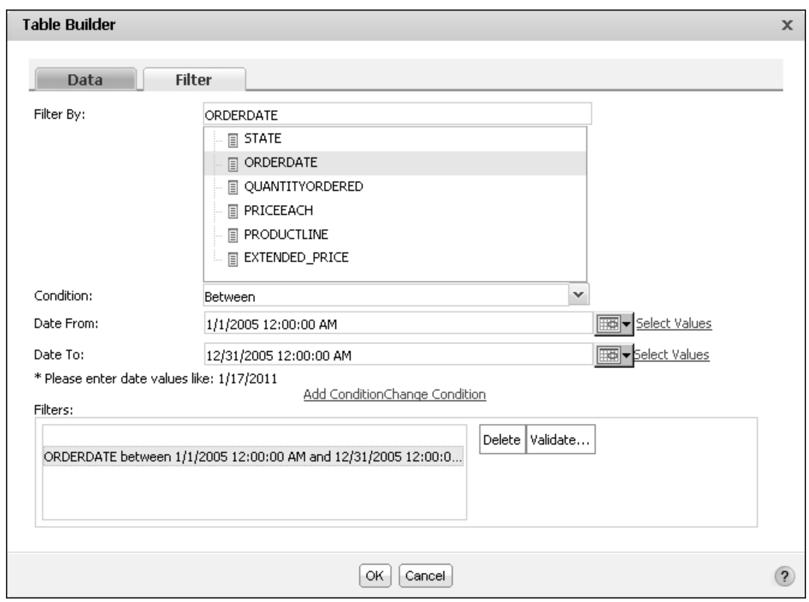


Figure 5-18 Specifying a data-set filter in Table Builder

2 In Condition, select an operator. Depending on the operator you choose, one or two value fields appear. The example in Figure 5-18 shows the selection of the operator Between, and two corresponding value fields.

3 In Value, do one of the following:

- Type a value.
- Choose Select Values, then select a value from the list of values that appears.

4 Choose Add Condition. Filters displays the new condition as shown in Figure 5-18.

5 Choose Validate to verify the syntax of the new condition.

Choose OK. BIRT Studio displays data fields that meet the filter condition in the report design.

How to modify a data-set filter

To modify an existing filter condition, complete the following steps on Table Builder—Filter:

1 In Filter By, select a new column from the drop-down list.

2 In Condition, select an operator. Depending on the operator you choose, one or two value fields appear.

3 In Value, do one of the following:

- Type a value.
- Choose Select Values, then select a value from the list of values that appears.

4 Choose Change Condition. Filters displays the modified condition.

5 Choose Validate to verify the syntax of the modified condition.

Choose OK. BIRT Studio displays data fields that meet the filter condition in the report design.

Prompting for filter values at run time

You can create a prompt that accepts a value or multiple values for a filter when a report runs. A prompt, also called a parameter, enables users reading your report to control the content of the report without having to edit the report. For example, in a report that displays sales data by sales office, instead of creating a filter that returns data for a specific office, you can create a prompt that asks the user to select the office for which they want to view data in the report. Even if you are the only one reading the report, a prompt makes it easy for you to view the report with different content. Without a prompt, you must edit the filter condition to supply a different value.

About report parameters

Report parameters provide a mechanism for collecting values from a report user or a program. You typically use report parameters in filters to collect information that determines the data to display in a report.

In BIRT Studio, you can create a static filter parameter or a dynamic filter parameter at the data-set level, or at the report table level. You can typically create static or dynamic parameters at the data-set level for reports using templates with data sets, as well as information objects or data objects.

To construct a filter parameter, complete the following tasks:

- Choose a filter parameter type.
- Define a condition.
- Specify whether the parameter is required or optional. Dynamic filter parameters are always optional.
- Select a Display Type from the list of available display options described in Table 5-1.
- Specify a list of values from which a user can select values. You can specify a static list, or a dynamic list. You can create a dynamic list of values at the data set level, or at the current column level.

Choosing a parameter type

A static filter parameter is a filter condition that the report developer defines, that prompts the user to specify one or more values when running a report. The filter condition typically consists of an operator, specified by the report developer, and corresponding values, to be specified by the report user when running a report.

A dynamic filter parameter differs from a static filter parameter in one important aspect. Using a dynamic parameter, the report developer can provide report users with a list of operators to use to construct a filter condition. The user selects an operator, then specifies one or more corresponding values when running the report. The report displays data for the filter parameter condition the user specifies.

Making a filter parameter optional

When you create a prompt, you can require the user to specify a value, or you can make the value optional. It is typically good practice to make the value optional, so the user can view the report with all the data. For example, if Parameters contains a list of sales offices for the user to choose from, if the prompt is not required, the user can leave the field blank to view sales data for all the sales offices.

On the other hand, you can require that the user supply a value if displaying all the data results in a very long report. A report that runs into hundreds of pages is not only difficult to read, but takes longer to generate.

In Actuate BIRT Studio, a dynamic filter parameter is always optional.

How to make a prompt value optional

To make a prompt value optional, in Filter, select Prompt Text. Prompt Text and Is Required appear highlighted. Deselect Is Required, as shown in Figure 5-19.

No Value appears in Filter. Select No Value if you do not want to supply any default values for the parameter. If you select No Value, any Default Value that you might have supplied is no longer valid and does not appear when you specify parameter values when you run the report.

You can also use the No Value field to add an empty value to a list of values when using the In operator.

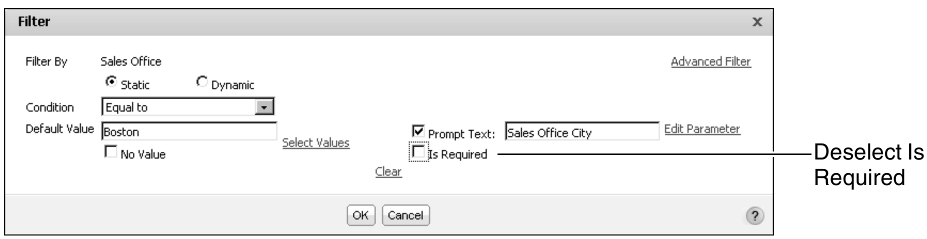


Figure 5-19 Specifying an optional prompt

Accepting multiple values

Users often want to select any number of values for a filter condition. In an inventory report, for example, the user might need to view data for several vendors. To support the selection of multiple values, create a filter parameter as follows:

- Select list box as the display type.
- Select the In operator as one of the operators to provide to the user.
- Create a list of values.

Specifying a display type

When specifying properties for a static filter parameter, you can select the display types shown in Table 5-1 for the list of values.When specifying properties for a dynamic filter parameter, you can specify List Box or Text Box as the display type.

Table 5-1 Choosing a display type for the list of values

Option	Description	Use
Combo Box	Provides a combination of a drop-down list and a text box, where the user can either select a value from the list or type a value.	Enables the user to select one or more values, or type a specific value for which BIRT Studio displays data.
List Box	Displays available values in a drop-down list.	Enables users to select one or more values. If you use the In operator, the list box is the only available display type for the list of values.
Radio Button	Displays the list of values as radio buttons.	Enables the user to select one value at a time. Additionally, this option enables a report developer to provide the user with a limited number of values from which to choose.
Text Box	Requires that the user type a value for which to display data.	Enables the user to type a value for which BIRT Studio displays data.
Text Box-Auto Suggest	Provides the user with the available values that match a certain number of typed characters.	Activate auto suggest after any number of typed characters. For example, if you activate auto suggest after one character, and if the user types S, all the values beginning with that character appear in a list from which the user selects a value. In case there is no match, BIRT Studio displays the message No Suggestions.

Providing the user with a list of values

To create a helpful prompt, you can provide the user with a list of values from which to choose. Do not assume that the report user knows what values to supply. For example, a user probably does not know that an order status field takes one of three values, Open, Closed, and In Process. In some cases, providing a list of values, such as customer names or invoice numbers, is necessary.

You create a list of values using one of the following techniques:

- Create a static list of values. In a static list, the values you select to display to the report user are fixed in the report design. You can either select from a list of values, or type values to populate the list of values.
- Create a dynamic list of values. In a dynamic list, the software generates the list of values when the report runs, using the current values in the report table or the data set. You can create a dynamic list of values at the data set level, or at the report column level.

Create a dynamic list for values that are frequently updated in the data source. For example, fields such as new customer names or product names are periodically updated in data sources. If you create a static list of these values, you need to update the list manually to match the values in the data source. Create a static list if you want to control the list of values displayed to the report user, for example, if you want to display only some of the values.

Creating a static filter parameter

For a static filter parameter, the report developer specifies the filter condition, consisting of an operator and display type to specify the value for which BIRT Studio displays data. The report developer also creates a list of values. A static filter parameter can be required or optional. This section describes how you can create a static filter parameter at the report table and data-set level.

How to create a static filter parameter at the report-table level

1 In BIRT Studio, select the column that contains the values for which you want to create a static prompt, then choose Filter. Filter appears, as shown in Figure 5-20.

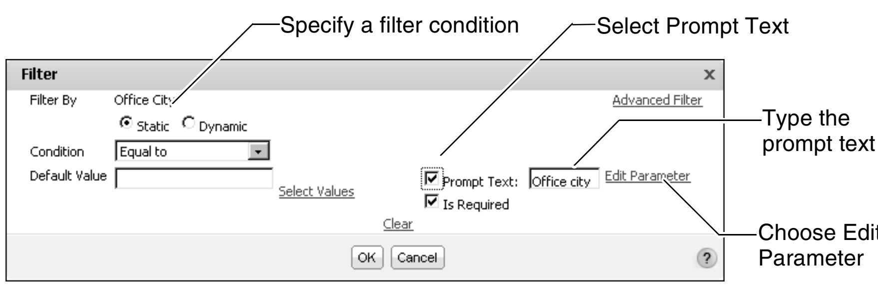


Figure 5-20 Creating a filter that prompts for a value when the report runs

2 In Filter, select Static, if necessary.

3 In Condition, select an operator.

4 Select Prompt Text, then type the text to display.

5 Choose Edit Parameter, as shown in Figure 5-20.

6 Edit Parameter appears. In Edit Parameter, complete the following steps:

1 In Prompt Text modify the existing text if necessary.

2 In Help Text, optionally type a Tooltip to assist the user in selecting the values for which BIRT Studio displays data.

3 Accept the default selection of Is Required. To make the parameter optional, deselect Is Required.

4 In Display Type, select an option from the list. The example in Figure 5-21 shows the display type set to Text Box.

5 Select the values to make available to the report user when the report runs. Choose Select Values. Select a value from the list of values that appears in Find, then press Ctrl and select the each additional value you want to display.

6 If you selected Text Box as the Display Type, select a value to set as the default. If you do not specify a default value, the first value you selected is set as the default value. The example in Figure 5-21 displays Boston as the default value.

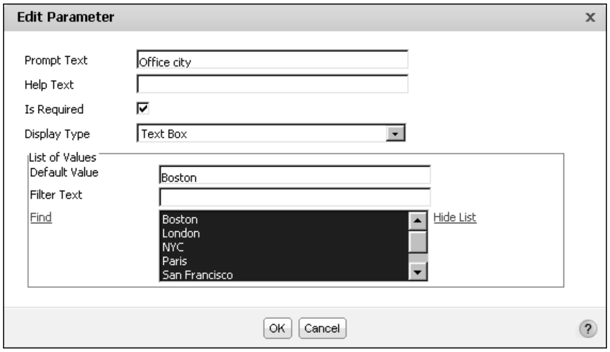


Figure 5-21 Creating a static parameter

Choose OK. The specified default value appears in Default Value, on Filter, as shown in Figure 5-22. Choose OK, the filter condition takes effect immediately.

Figure 5-22 Filter displaying the specified parameter properties

How to create a static filter parameter with a list of values

When you create a static filter parameter, and select an available display type other than Text Box, you can create a static or dynamic list of data values to provide to the user. To create a static filter parameter with a list of values, on Edit Parameter, complete the following steps:

1. In Prompt Text, if necessary, type the display text for the prompt.
2. In Help Text, optionally type a Tooltip to assist the user in selecting values for which BIRT Studio displays data.
3. Accept the default selection of Is Required. To make the parameter optional, deselect Is Required.
4. In Display Type, select an option other than Text Box from the list of options. The example in Figure 5-23 shows the selection of List Box.

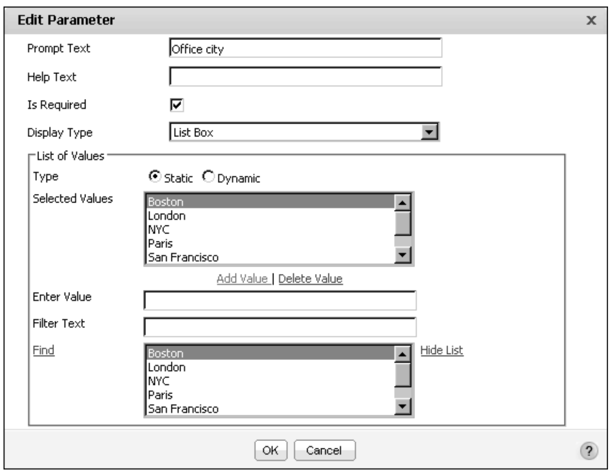


Figure 5-23 Creating a static list of values

5. In List of Values, do one of the following:
   - Select Static to create a static list of values, that the report developer specifies. If you select Static, proceed to step 6.
   - Select Dynamic to create a list of values that is retrieved from the data source when the report runs. If you select Dynamic, proceed to step 8.
6. Select the values to make available to the report user when the report runs. Choose Select Values.

   Select a value from the list of values that appears in Find, then press Ctrl and select any additional values you want to display. You can also type each value in Enter value, then choose Add value. The added values appear in Selected Values.

7 In Selected Values, select a value to set as the default value, as shown in Figure 5-23.

- If the parameter is optional, you do not need specify a default display value.
- If the parameter is optional, No Value appears on Edit Parameter. If you do not want to supply any values for the parameter, select No Value. If you select No Value, any default value that you might have supplied earlier is no longer valid. When the user selects No Value at run time, all the data is displayed in the report.

Choose OK. Proceed to step 10.

8 In Data Set, from the drop-down list, select Current column (No Data Set) to create the list of values from the report table, or select an available data set to create the list of values at the data set level. If you selected the current column do the following, as shown in Figure 5-24:

- Choose Select Values. Select a value from the list of values that appears in Find, then press Ctrl and select each additional value to display. The first value you select is set as the default value.
- In Sort, in Sort Direction, optionally select Ascending or Descending. If you do not specify a sort direction, BIRT Studio displays the default value first, and arranges the remaining values in ascending order below the default value.

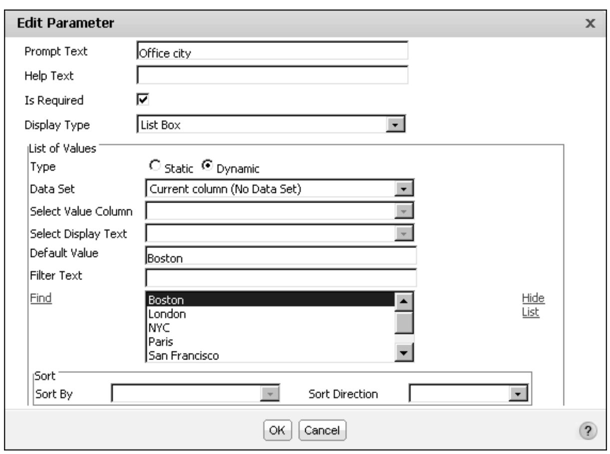


Figure 5-24 Creating a dynamic list of values

Choose OK. Proceed to step 10.

9 In Data Set if you selected an available data set, do the following:

- In Select Value Column, select the column whose values are filtered according to the parameter condition.

- In Select Display Text, select a column from the drop-down list, containing the values displayed to the user. For example, you can choose to display the Office Code values for the user to choose from, instead of the names of sales office cities.
- Choose Select Values. Select a value from the list of values that appears in Find, then press Ctrl and select the remaining values you want to display. The first value you select is set as the default.
- In Sort, specify a field to sort the list of values by.
- In Sort direction, select Ascending, or Descending from the drop-down list.

  If you do not specify a sort direction, BIRT Studio displays the default value first, and arranges the remaining values in ascending order below the default value.

Choose OK.

10 On Filter, choose OK.

11 Test the prompt in BIRT Studio by choosing Parameter. Parameters displays the prompt, as shown in Figure 5-25. In this example, the default parameter value is Boston.

Figure 5-25 Parameters displaying the specified prompt

How to create a static filter parameter at the data-set level

1 Create a new report by selecting a template containing a data set, or an information object.

2 Select the data fields to insert in the report, and lay out the report.

3 In BIRT Studio, in Available Data, choose Switch Filter Pane.

4 Filter Pane appears, as shown in Figure 5-26. Choose Add to define a filter condition.

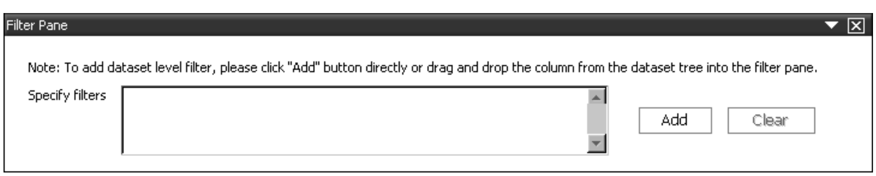


Figure 5-26 Filter Pane

5 Advanced Filter appears, as shown in Figure 5-27. On Advanced Filter, in Filter By, select a column from the drop-down list.

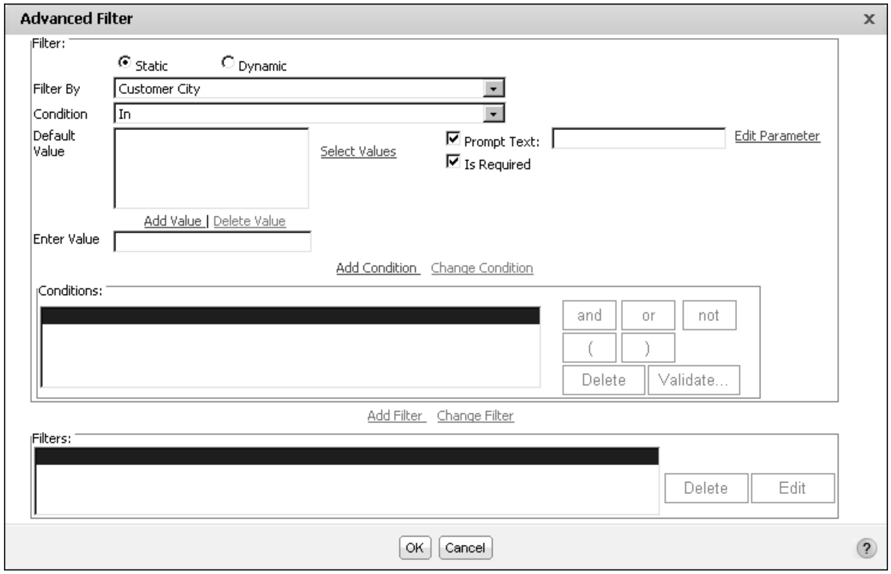


Figure 5-27 Defining a static filter parameter in Advanced Filter

6 In Condition, select an operator from the drop-down list.

7 Select Prompt Text, then type the text to display. Edit Parameter is highlighted, and Is Required appears selected.

8 Choose Edit Parameter. In Edit Parameter, create the parameter condition as follows:

- To create a static filter parameter using a list of values, complete steps 1 to 9 in “How to create a static filter parameter with a list of values,” earlier in this chapter.
- To create a static filter parameter displaying a text box where the user types the value, complete steps 1 to 6 in “How to create a static filter parameter at the report-table level,”earlier in this chapter.

Choose OK. Figure 5-28 displays the completed filter parameter condition.

The selected default value appears in Default Value, as shown in Figure 5-28.

9 Choose Add Condition. The condition appears in Conditions, as shown in Figure 5-28.

10 Choose Add Filter. The static filter parameter condition appears in Filters.

Choose OK. Filter Pane displays the specified condition.

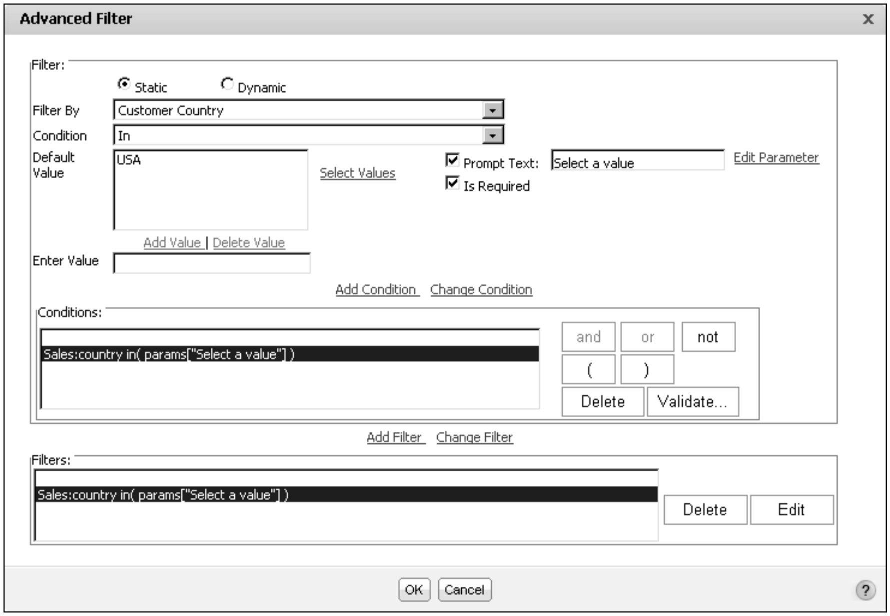
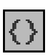


Figure 5-28 Advanced Filter displaying the static filter parameter condition



11 Test the prompt in BIRT Studio by choosing Parameter. Parameters displays the prompt, as shown in Figure 5-29. The default value specified in the example condition is USA.

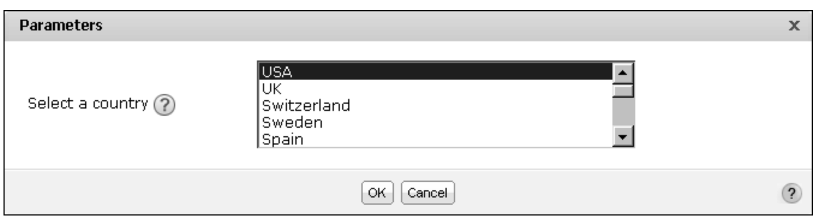


Figure 5-29 Parameters displaying that static parameter condition

Creating a dynamic filter parameter

Dynamic filter parameters provide users with more control over the data they view in a report. Instead of specifying only the value on which to filter, the report user can specify conditions, such as Profit Less than 1000, Profit Between 1000 and 2000, or Profit Greater than 2000. The user can also choose to view all totals, omitting any filter conditions. A dynamic filter parameter is always optional. Figure 5-30 shows an example of a dynamic filter parameter.

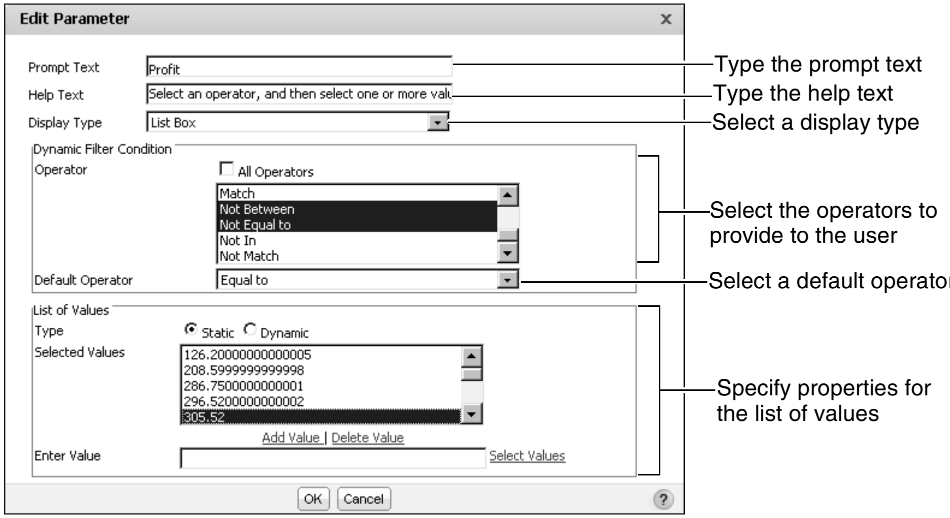


Figure 5-30 Specifying properties for a dynamic filter parameter

In BIRT Studio, dynamic filter parameters modify the underlying query, so filtering occurs at the data-source level. This method of querying the data source for the most recent information also provides users with more accurate information. Only data rows that meet the filter criteria are retrieved from the data source. Retrieving a limited number of rows improves performance and reduces the time it takes for the data to appear in the report. The basic properties you specify when you create a dynamic filter parameter, are similar to the properties for a static filter parameter, with one exception. On Edit Parameter, you specify the operators to provide to users, and set a default operator. Like a static filter parameter, a dynamic filter parameter can also provide the user with a list of values. However, to provide the user with a static or dynamic list, values must be presented in a list box. Figure 5-30 shows an example of a dynamic filter parameter that includes a list of values. The display type is set to List Box.

How to create a dynamic filter parameter at the report-table level

1. Select the column for which you want to create a prompt, and choose Filter. Filter appears. The selected column appears in Filter By.
2. Select Dynamic, as shown in Figure 5-31.

Figure 5-31 Specifying filter parameter properties

3 In Prompt Text, type a description for the prompt. Then, choose Edit Parameter.

4 In Edit Parameter, complete the following steps:

    1 In Prompt Text, if required, modify the text you typed earlier.

    2 In Help Text, optionally type a Tooltip that assists the user in selecting parameter values, for which BIRT Studio displays data.

    3 In Display Type, choose List Box.

    4 In Dynamic Filter Condition, in Operator, select one or more operators from the drop-down list to provide to the user. By default, all operators are selected. To select operators from the list, deselect All Operators. Then, press Ctrl and select each operator from the list of available operators.

    5 Select a default operator from the drop-down list.

    6 In List of values, do one of the following:

        - Select Static to create a static list of values. A static list contains values specified by the report developer. If you selected Static, proceed to step 7.
        - Select Dynamic to create a dynamic list of values. A dynamic list contains values that are generated at run time from the report table or the data set. If you selected Dynamic, proceed to step 9.

    7 Choose Select Values, and select the values from the list of values in Find. You can also type each value in Enter value, then choose Add value. The added values appear in Selected Values.

    8 In Selected values, select a value to set as the default.

    Choose OK. Proceed to step 5.

    9 In Data Set, select Current Column (No data set), to create the list of values from the report table, or select an available data set to create the list of values at the data set level. If you selected the current column, do the following:

        - Choose Select Values. Select a value from the list of values that appears in Find to set as the default value.
        - In Sort, in sort direction, select Ascending or Descending. If you do not specify a sort direction, BIRT Studio displays the default value first, and arranges the other values in ascending order below the default value.

        Choose OK. Proceed to step 5.

    10 In Data Set, if you selected an available data set, do the following:

- In Select Value Column, select the column whose values are filtered according to the parameter condition. The example shown in Figure 5-32 uses the Product Name column.
- In Select Display Text, optionally select a column from the drop-down list, containing the values displayed to the user. In the example shown in Figure 5-32, displayed values are selected from the Product Code column.

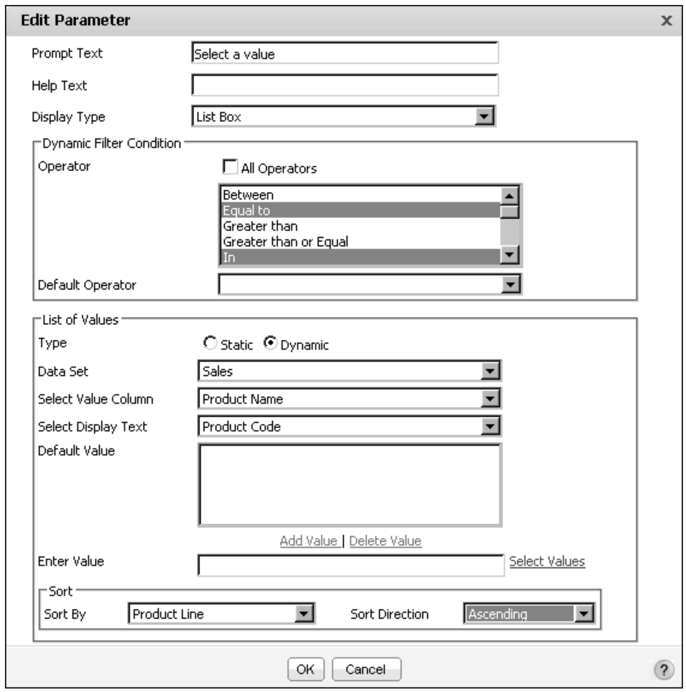
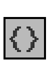


Figure 5-32 Specifying a dynamic list of values

- In Default value, optionally specify a default value.
- In Sort By, select a sort field, by which the data in the list of values is sorted. In the example shown in Figure 5-32, displayed values are sorted by Product Line.
- In Sort Direction, select Ascending or Descending.

  If you do not specify a sort direction, BIRT Studio displays the default value first, and arranges the remaining values in ascending order below the default value. Choose OK.

5 On Filter, choose OK.



6 To test the prompt in BIRT Studio choose Parameter. Parameters displays the prompt, as shown in Figure 5-33.

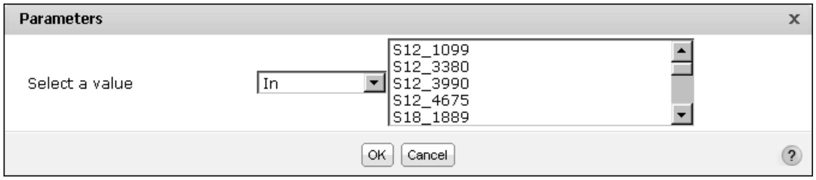


Figure 5-33 Parameters displaying a dynamic prompt

How to create a dynamic filter parameter at the data-set level

1 Create a new report by selecting a template containing a data set, an information object, or a data object.

2 Select the data fields to insert in the report.

3 In BIRT Studio, in Available Data, choose Switch Filter Pane.

4 Filter Pane appears. Choose Add to define a filter condition. Advanced Filter appears.

5 On Advanced Filter, select Dynamic, as shown in Figure 5-34.

6 In Filter By, select a column from the drop-down list.

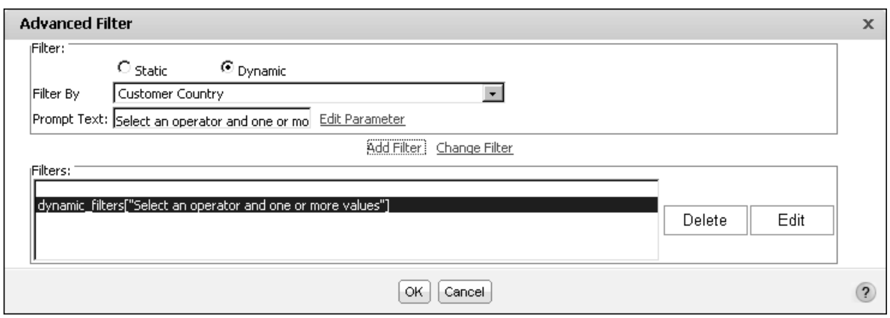
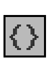


Figure 5-34 Advanced Filter displaying the dynamic filter parameter

7 Choose Edit Parameter. In Edit Parameter, create the parameter condition by completing step 4 in "How to create a dynamic filter parameter at the report-table level," earlier in this chapter.

In Advanced Filter, Prompt Text displays the text you entered earlier.

8 Choose Add Filter. The dynamic filter parameter condition appears in Filters, as shown in Figure 5-34. Choose OK. Filter Pane displays the condition.



9 To test the prompt in BIRT Studio choose Parameter. Parameters displays the prompt, as shown in Figure 5-35. The default operator specified in the example condition is In, and the default value is USA.

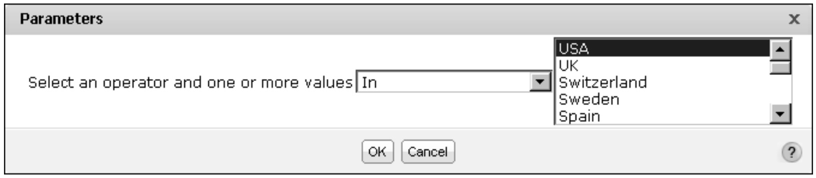


Figure 5-35 Parameters displaying the dynamic prompt

When working with a dynamic prompt set on a column containing numeric data, that uses the Match or Not Match operators, if you use the value 1 as the string value to match you must use double quotation marks (" ") to enclose the value. Type the value as "1" in the text box or combo box, so that BIRT Studio evaluates it accurately as a literal string. If you use other values such as 2, 3, and so on, you do not need to enclose them in double quotation marks.

Creating multiple filter parameters

A report developer can also define multiple conditions, for which a user specifies values, further narrowing the scope of data displayed in a report. For example, you can define the first condition such that the user specifies a Sales Office value, the next condition where the user specifies one or more Product Lines, and a third condition where the user specifies a Profit value or range. The user can specify that the Sales Office value is equal to Boston, the Product Line is In Classic Cars, and Motorcycles, and that the Profit is less than or equal to 1001.10, as shown in the example. BIRT Studio displays the data rows that fulfill all three conditions.

You can specify multiple static filter parameters, dynamic filter parameters, or a combination of both types. You create multiple filter parameters on Advanced Filter. The following section describes how to define multiple filter parameters in a report design.

How to create multiple filter parameters

1 Select the first column for which you want to create a prompt, and choose Filter. Filter appears. The selected column appears in Filter By.

2 Select Static or Dynamic.

- To create a static filter parameter, complete steps 1 to 9 in “How to create a static filter parameter with a list of values.”
- To create a dynamic filter parameter, complete steps 1 to 10 in “How to create a dynamic filter parameter at the report-table level,” earlier in this chapter.

3 On Filter, choose Advanced Filter.

4 On Advanced Filter, choose Add Condition. The filter parameter condition you previously specified appears in the Filters area in Advanced Filter.

5 Choose Add Filter.

6 To define the next condition, in Advanced Filter complete the following steps:

1 In Filter, select Static or Dynamic.

2 In Filter By, select a column from the drop-down list.

3 In Prompt Text, type the display text for the prompt.

4 Choose Edit Parameter.

- If you selected Static in step 1, complete steps 1 to 9 in "How to create a static filter parameter with a list of values," earlier in this chapter.

- If you selected Dynamic in step 1, complete steps 1 to 10 in "How to create a dynamic filter parameter at the report-table level," earlier in this chapter.

5 In Advanced Filter, choose Add Filter.

7 To define additional filter parameters, in Advanced Filter, repeat step 6.

Choose OK. The example in Figure 5-36 shows Advanced Filter displaying three filter parameter conditions.

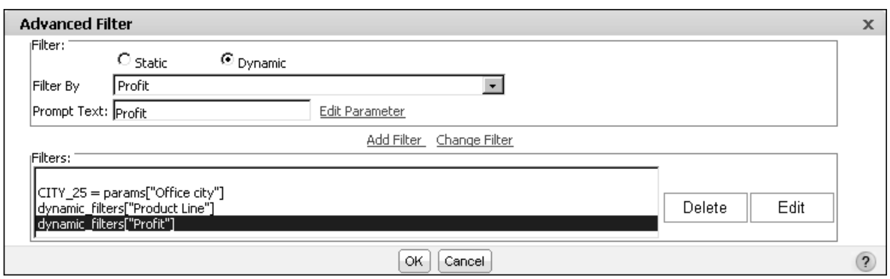
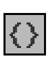


Figure 5-36 Defining multiple filter parameters in Advanced Filter



8 To test the prompt in BIRT Studio choose Parameter. Parameters displays the prompt with all the defined conditions. The example shown in Figure 5-37, displays three conditions for which the user can supply values. The first displayed condition is a static filter parameter, and the remaining two conditions are dynamic filter parameters.

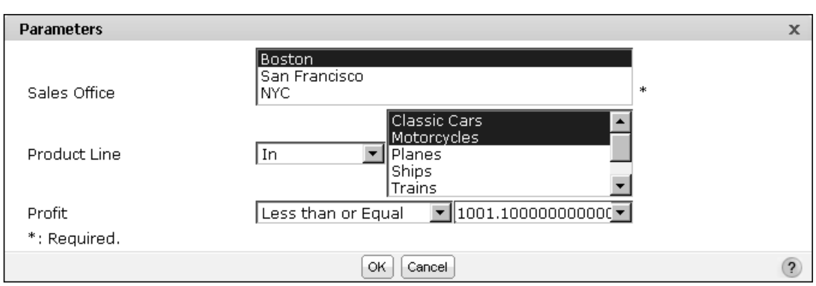


Figure 5-37 Specifying multiple filter parameter values

Chapter

6

Presenting data in a cross tab

This chapter contains the following topics:

- About cross tabs
- Obtaining data for a cross tab
- Creating a cross tab
- Formatting data in a cross tab
- Analyzing data in a cross tab

About cross tabs

A cross tab displays data in a row-and-column matrix that has a spreadsheet-like appearance. The cross tab is ideal for summarizing data in a compact and concise format, and displays summary, or aggregate, values such as sums, counts, or averages. The cross tab groups these values by one set of data listed down the left side of the matrix and another set of data listed across the top of the matrix.

Figure 6-1 shows a cross tab that displays sales totals by state and by product line.

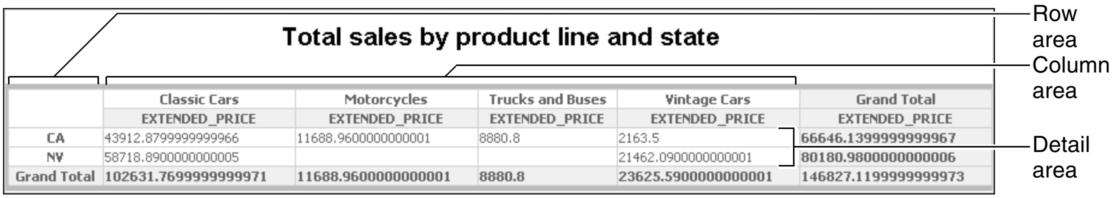


Figure 6-1 Displaying sales by state and product line in a cross tab

The cross tab uses data from the state, product line, and extended price fields. A cross tab typically uses data from at least three fields as follows:

- One field populates the column headings in the cross tab. There is one column for each unique value in the field. In the example shown in Figure 6-1, five unique values from the productline field; Classic Cars, Motorcycles, Planes, Ships, and Trains, appear in an area called the column area.
- One field populates the row headings in the cross tab. There is one row for each unique value in the field. In this example, eight unique values from the state field; CA, CT, MA, NH, NJ, NV, NY, and PA, appear in the row area.
- BIRT Studio aggregates the values in one field and displays these values in the cross-tab cells. In this example, each cell displays a revenue total by productline and state. Revenue totals are calculated using the SUM function on the values in the extendedprice field. Totals appear in the detail area.

You can use BIRT Studio to insert a cross tab in a report design, select data for the cross tab, and specify the aggregate data to display. In addition, you can use the Actuate BIRT Data Analyzer, which requires a separate license, to edit data in the cross tab, format the data values, and modify cross tab data. You can access the Actuate BIRT Data Analyzer from BIRT Studio and Interactive Viewer.

Obtaining data for a cross tab

A cross tab typically derives data from a cube. A cube is a multidimensional data structure that is optimized for analysis and reporting. In BIRT Studio, a cross tab retrieves data from a data object, which typically contains at least one data set and can contain at least one cube, but can contain many data sets and many cubes. Ensure that you select a data object that contains one or more cubes to create the cross tab.

Working with cubes

A cube organizes data into dimensions and measures as follows:

- Measures represent values that are counted or aggregated, such as costs or units of products.
- Dimensions are categories, such as products, customers, or sales periods, used to aggregate measures. Dimensions can be hierarchical and contain multiple levels. For example, a region dimension can contain a region-country-state hierarchy. Similarly, a time dimension can contain a year-quarter-month-week hierarchy. Most cubes include a time dimension because, for most reports, showing measures by day, week, month, quarter, or year is essential to analyzing data.

For example, a retail cube can contain data that supports viewing sales volume and cost of goods, which are measures, by store location, time period, and product lines, which are dimensions.

In a cross tab, the row and column areas display the dimensions from a cube. The dimension values form the row and column headings of the cross tab. The detail area contains one or more measures from a cube, displaying the aggregate data.

Enabling users to drill through data categories

Sometimes data objects created in BIRT Designer Professional, contain links to enable users to drill through the data hierarchy in the data object to view either summary or detail information. For example, consider data organized in years, that contains a quarter-month-week hierarchy. If the data object contains defined hyperlinks for this hierarchy, choosing a year displays the quarters that make up the year, choosing a quarter displays the months in the quarter, and so on.

Creating a cross tab

In BIRT Studio, complete the following tasks to create a cross tab:

- Select data for the cross tab.
- Lay out the cross tab.
- Specify display options for the aggregate values.

The following section describes each of these tasks in detail.

Selecting data for a cross tab

When building a cross tab, you group the aggregate data by at least two dimensions, for example, sales totals by year and product line or sales totals by product line and state. Often, report users need to view aggregate data by more than two dimensions.

Figure 6-2 shows data selected for a cross tab displaying the sales total by state and product line. To create the cross tab, one dimension (State) is inserted in the row area, and one dimension (Productlines) is inserted in the column area.

Each additional dimension by which you group data appears as a column or row, and each additional dimension provides a more comprehensive and detailed view of the data. Just as you can define an unlimited number of dimensions for a cube, you can build a cross tab that displays aggregate data by as many dimensions as needed.

Calculating aggregate data by too many dimensions, however, can result in many empty cells, a problem commonly referred to as data sparsity. When designing a cross tab that contains more than two dimensions, make sure that report processing time is not spent calculating zeros. In addition, a cross tab that contains more than two or three dimensions in either the row or column area is difficult to read. Rather than displaying data by too many dimensions in a single cross tab, consider dividing the data into multiple cross tabs, so the data is easier for the user to understand quickly.

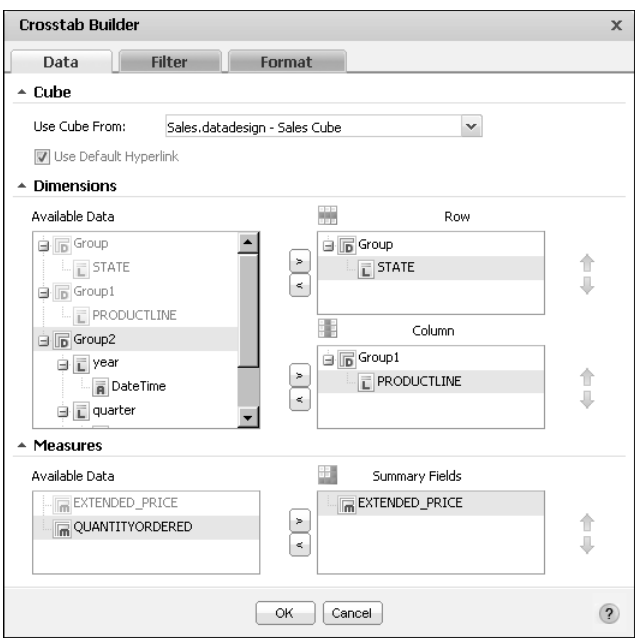


Figure 6-2 Displaying the dimensions and measures for a cross tab

How to select data for a cross tab

1 In BIRT Studio, choose New. Report Template appears.

2 In Report Template, complete the following steps:

1. In Category, select Standard from the drop-down list.
2. In Available Templates, select Crosstab, as shown in Figure 6-3.
3. In Themes, select a theme from the drop-down list if your Crosstab template contains associated themes.

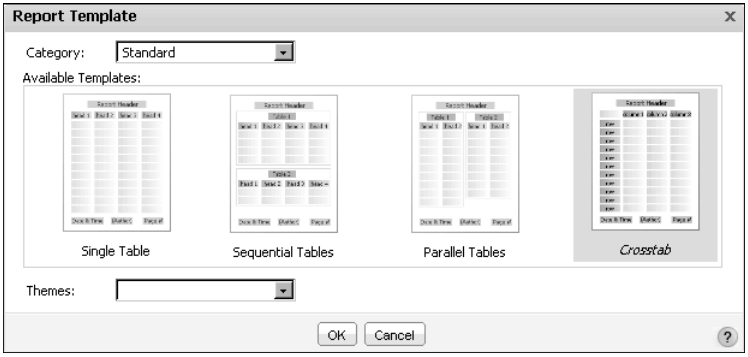


Figure 6-3 Choosing the cross tab template

Choose OK. Crosstab Builder—Data appears.

3 In Cube, in Use Cube From, choose New Data from the drop-down list, as shown in Figure 6-4.

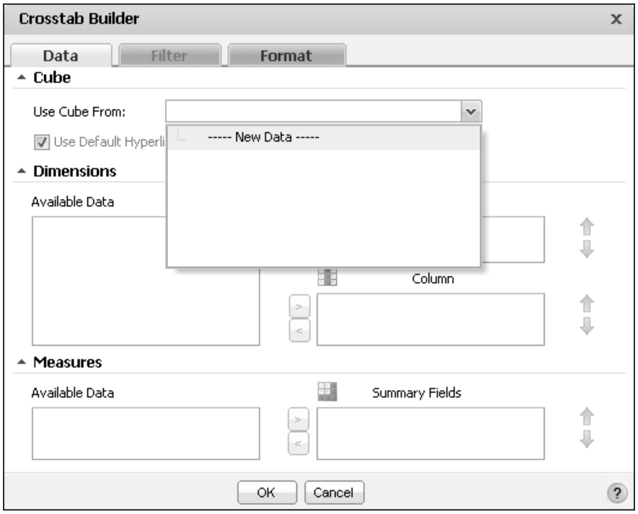


Figure 6-4 Selecting a data object data source

Select Data appears, displaying the available data objects. If you use customized data objects, ensure that they are in the /Resources folder of the BIRT iServer Encyclopedia volume.

4 In Available Data, select a data object design or data object store file. You can select a data object design (.datadesign) file to retrieve data on demand, or a data object store (.data) file to use cached data. Select a data object from the list, and choose the + symbol next to the selection, as shown in Figure 6-5.

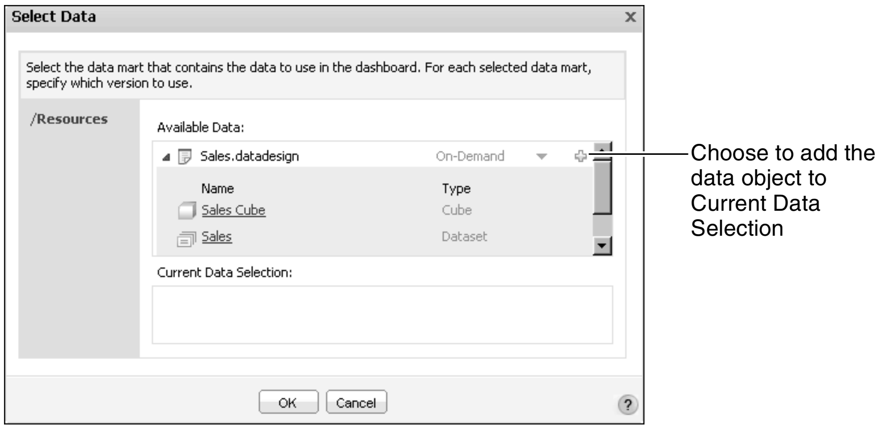


Figure 6-5 Selecting a data object

The data object appears in Current Data Selection, as shown in Figure 6-6. Choose OK. You can modify your selection by choosing Delete, as shown in Figure 6-6, and selecting a new data object in Available Data.

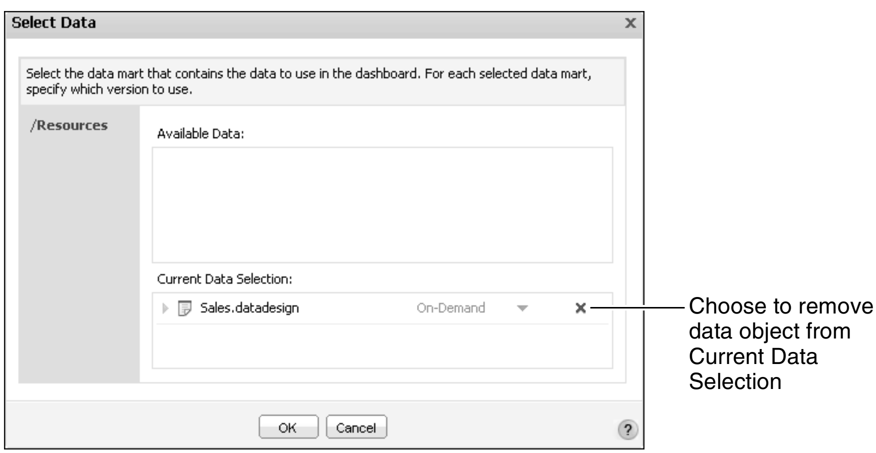


Figure 6-6 Current data selection displaying the data object

Crosstab Builder—Data displays the data from the selected data object, organized in dimensions and measures, as shown in Figure 6-7.

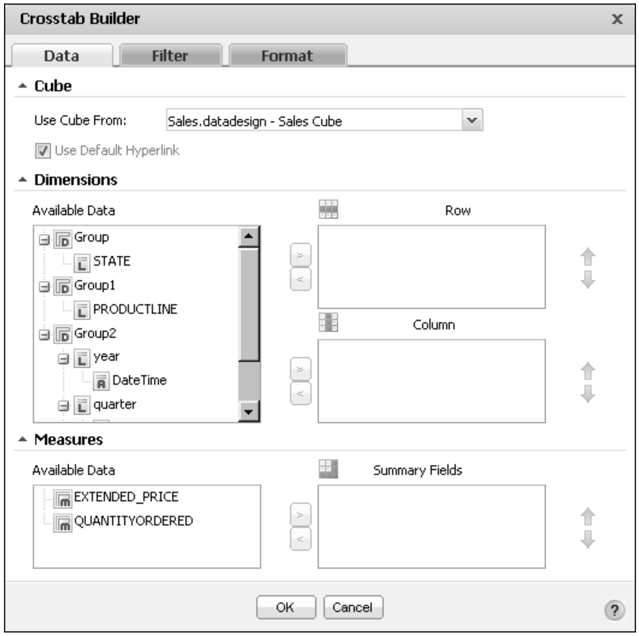


Figure 6-7 Displaying dimensions and measures in the data object

5 To display a subset of the available data in the cross tab, you can specify a filter condition for the data object. To create a filter at the data-object level, choose Filter and specify one or more filter conditions, as you would do for a BIRT report. For example, to view data only for Pennsylvania (PA) create a filter on the state dimension, using the Equal To operator, as shown in Figure 6-8.

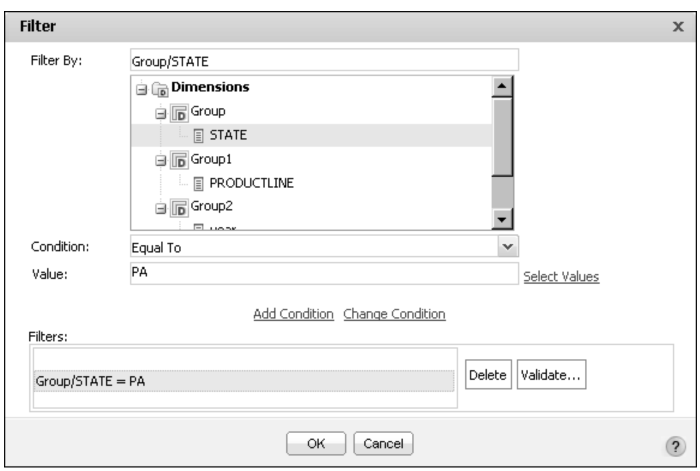


Figure 6-8 Specifying a data object filter

The cross tab displays aggregate extended price data by product line, for only the state of Pennsylvania, as shown in Figure 6-9.

	Classic Cars	Motorcycles	Vintage Cars	Grand Total
	EXTENDED_PRICE	EXTENDED_PRICE	EXTENDED_PRICE	EXTENDED_PRICE
PA	102856.2400000000026	39025.0900000000004	134925.009999999999	**240024.6200000000037**
Grand Total	**102856.2400000000026**	**39025.0900000000004**	**134925.009999999999**	**240024.6200000000037**

Figure 6-9 A cross tab displaying filtered data rows

You can also select the In operator in Condition to set a filter condition for more than one value. For example, you can define a filter condition that displays data for Pennsylvania (PA), California (CA), and Massachusetts (MA) in the cross tab.

Laying out data in a cross tab

When you lay out data in a cross tab, provide the following information:

- In Dimensions, specify:
  - The grouped data fields to display as rows
  - The grouped data fields to display as columns
- In Measures, specify the summary fields to display in the cross tab.

How to lay out data in a cross tab

1. Select the cross-tab template, then select the cube to use.
   Crosstab Builder—Data appears, as shown in Figure 6-10.
2. In Crosstab Builder—Data, in Dimensions, in Available Data, select a grouped data field. To select multiple data fields press Ctrl as you select each field.
3. Choose the right arrow next to Row to set the selected field as a row, or choose the right arrow next to Column to set the selected field as a column. The example shown in Figure 6-10, specifies State as a Row, and Product Line as a column.
4. In Measures, in Available Data, select a data field to aggregate, then choose the right arrow to set the selected field as a summary field. The example in Figure 6-10 sets the Extended Price field as the summary field.

   Choose OK. The selected fields appear in a cross tab in BIRT Studio.

Displaying aggregate values

You can display totals for each dimension that you add to a cross tab and for each level within a multilevel dimension. The cross tab in Figure 6-10 displays sales data by product line, and state. The rows and columns that display the subtotals and grand totals are highlighted in the cross tab. In the example cross tab you just created, no subtotals appear. Subtotals are displayed when you set a multidimensional data field, as a row or column.

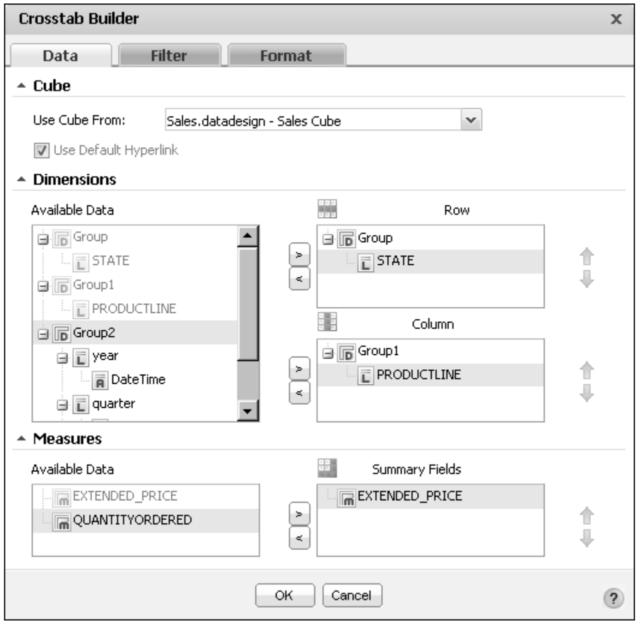


Figure 6-10 Selecting data to display in the cross tab

Each number displayed in a cross tab represents an aggregate total. Typically Grand totals display the total sales of all products for each state, the total sales of each product, or the total of all sales across products and states. Sub totals display the sales of each product in each state and so on. BIRT Studio does not create the subtotal and grand total rows and columns by default. You specify the aggregate totals that appear in a cross tab.

How to display aggregate values in a cross tab

1 To specify display properties for the aggregate values in the cross tab, in Crosstab Builder, choose Crosstab Builder—Format, as shown in Figure 6-11.

2 In Grand Totals, do the following:

- Select Show Grand Totals for Rows to display grand totals for each row.
- Select Show Grand Totals for Columns to display grand totals for each column.

3 You can display subtotals for multilevel dimensions used in rows or columns. Because the example in Figure 6-10 does not contain multilevel dimensions, the Sub Totals section is not highlighted. In Sub Totals, if highlighted, do the following:

- Select Show Sub Totals for Rows.
- Select Show Sub Totals for Columns

4 In Page Break, select Enable Page Break to specify pagination properties for the cross tab, do the following, as shown in Figure 6-11:

- In Row Interval, type a value such as 40.
- In Column Interval, type a value, such as 10.

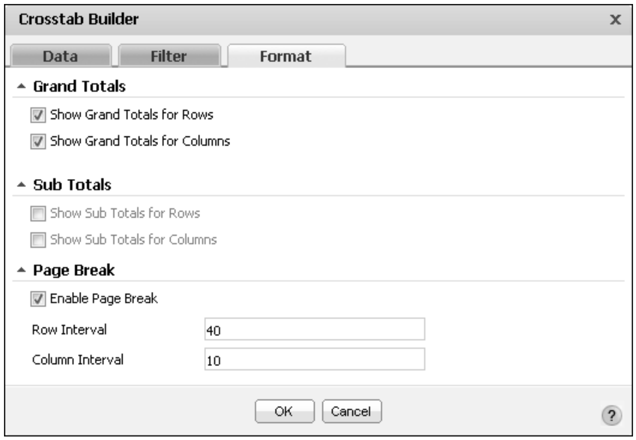
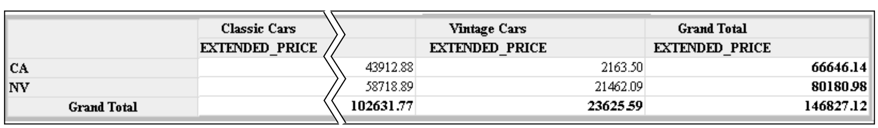


Figure 6-11 Specifying display options for the aggregate values

Choose OK. Sample data appears in the cross tab, displaying grand totals for rows and columns, as shown in Figure 6-12.



	Classic Cars		Vintage Cars	Grand Total
	EXTENDED_PRICE		EXTENDED_PRICE	EXTENDED_PRICE
CA		43912.88	2163.50	66646.14
NV		58718.89	21462.09	80180.98
Grand Total		102631.77	23625.59	146827.12

Figure 6-12 Displaying grand totals for rows and columns in a cross tab

Saving and viewing a cross tab

As with other report design files, BIRT Studio displays only a preview of the actual data in the cross tab. You save and view a cross tab in the same way that you save and view a BIRT report design, as described in earlier sections of this document.

How to save and view a cross tab

1 In BIRT Studio, choose Save and view.

2 Save Report Design appears. Specify a file name, and a folder location in which to store the saved file. Choose OK.

The cross tab appears in Actuate BIRT Viewer, as shown in Figure 6-13.

	Vintage Cars	Grand Total
	EXTENDED_PRICE	EXTENDED_PRICE
CA	366355.3700000000039	992815.49
CT	14101.07	119443.84
MA	105384.1799999999988	387238.75
NH	39376.6500000000002	116449.29
NJ	9035.3600000000003	9035.36
NV	21462.0900000000001	80180.98
NY	62342.2800000000003	400958.01
PA	34925.009999999999	175264.34
Grand Total	652982.01	2281386.06

Figure 6-13 Viewing the cross tab in Actuate BIRT Viewer

Formatting data in a cross tab

As the examples in this document reflect, the data in a cross tab appears as it is stored in the data object. To customize formatting properties for a cross tab, you can use Interactive Viewer. You can use Interactive Viewer to complete the following tasks in a cross tab:

- Display a detail Tooltip or a description of the aggregate data values.
- Modify font type, size, color, and style of data.
- Modify alignment properties of text.
- Format data based on its type, for example, format currency as US dollars, as shown in Figure 6-14.

	Trains	Trucks and Buses	Vintage Cars	Grand Total
	EXTENDED_PRICE	EXTENDED_PRICE	EXTENDED_PRICE	EXTENDED_PRICE
CA	$17,965.32	$167,896.48	$366,355.37	$1,348,882.59
CT	$9,548.53	$15,671.49	$14,101.07	$215,771.06
MA	$12,184.49	$58,487.98	$105,384.18	$613,791.30
NH		$7,922.29	$39,376.65	$116,449.29
NJ			$9,035.36	$81,806.55
NV			$21,462.09	$80,180.98
NY	$15,033.47	$77,996.00	$62,342.28	$576,373.66
PA	$4,862.02	$37,483.09	$34,925.01	$240,024.62
Grand Total	$59,593.83	$365,457.33	$652,982.01	3273280.0500000000085

Figure 6-14 Cross tab displaying applied currency format

- Launch BIRT Data Analyzer to analyze cross-tab data.
- Export data and charts to other file formats.

Analyzing data in a cross tab

From BIRT Studio, you can launch BIRT Data Analyzer if you have purchased a license for it, to manipulate and further analyze cross-tab data. Using BIRT Data Analyzer, you can answer questions such as:

- What the sales for a product are over time

- How total sales break down by product
- How many units of a product shipped to all locations in April

You can add, remove, reorganize, and customize the data and its appearance in the cross tab to examine relationships and trends. Using BIRT Data Analyzer, you can perform the following tasks:

- Filter, group, and sort data.
- Format a cross tab.
- Perform calculations.

To launch BIRT Data Analyzer, select any cell in the cross tab, then choose Analyze from the context menu, as shown in Figure 6-15.

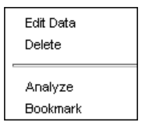


Figure 6-15 Launching BIRT Data Analyzer from BIRT Studio

Chapter

7

Working with summary tables

This chapter contains the following topics:

- About summary tables
- Creating a summary table
- Modifying a summary table
- Inserting a chart in a summary table
- Changing table types

About summary tables

A summary table presents aggregate data information in a report, providing users with a concise view of the data. A summary table helps improve response time, reduce network traffic, and enables customers to view the data they need in a format that is easily understood. The summary table feature in BIRT Studio enables report users to display only aggregate data from data sets that contain a large volume of data. The aggregate calculation occurs at the data-source level, which enables users to quickly view summary information at a glance.

When a report design presents all detail rows of a data set, with or without aggregate data, the table is called a detail table. By hiding the details for a group in a detail table, you can provide the user with a simplified view of the aggregate data. Using a summary table to achieve the same result improves performance, and reduces the load on the server, when dealing with large volumes of data.

For example, from a data source that contains two fields Order Status and Amount, using a summary table, the user can view the Total Amount for each Status. In an ordinary report design, BIRT Studio retrieves the sales for each status and calculates the sum of the Amount. When you use a summary table, BIRT Studio calculates the sum of the Amount field at the data-source level, and presents the corresponding values in the report design. You can use a template containing one or more data sets, an information object, or a data object to create a summary table. This section describes how you can use BIRT Studio to create and modify a summary table.

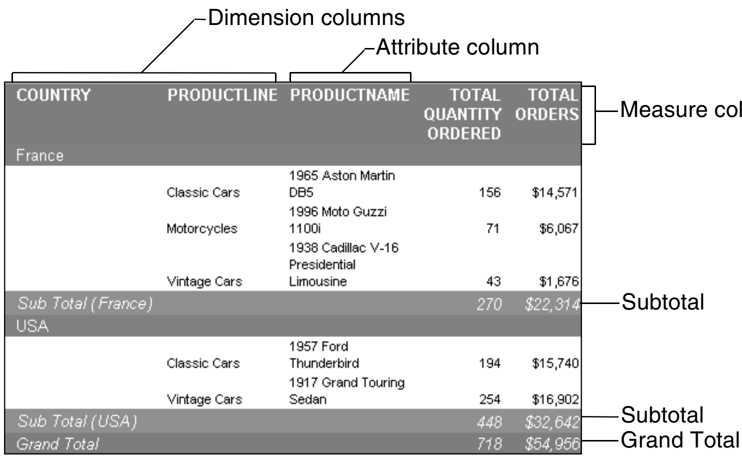
About columns in a summary table

The columns in a summary table are dimension columns, attribute columns, and measure columns. Dimension columns group data in other columns. Measure columns contain the aggregated values that are evaluated. Some examples of dimension columns include order date, country, state, or product line. Dimension columns containing date-and-time data can be grouped in intervals, in a summary table. Depending on the type of data in a measure column, BIRT Studio makes available specific aggregate functions to use on the column. Some examples of measure columns include quantity ordered, profit, and revenue.

In a summary table, columns that provide additional information about a dimension field, are called attribute columns. For example a product name, or product code data field can be attributes of the Product Line dimension. Each dimension can contain several attribute fields.

Designing a summary table

When you design a summary table, first evaluate which dimension, attribute, and measure columns to select, based on the aggregate data you want to display. In the example report shown in Figure 7-1, data in the measure column is grouped by the dimension, and displays subtotals in the group footer. Duplicate values in a grouped dimension column do not appear in the summary table. Although there are two dimension columns in the example in Figure 7-1, the outermost column is used to group data in the other columns. The attribute column, Product Name, is an attribute of the Product Line dimension. You can perform multiple aggregate calculations for data in a single measure column. A grand total is displayed in the report table footer.



COUNTRY	PRODUCTLINE	PRODUCTNAME	TOTAL QUANTITY ORDERED	TOTAL ORDERS
France				
	Classic Cars	1965 Aston Martin DB5	156	$14,571
	Motorcycles	1996 Moto Guzzi 1100i	71	$6,067
	Vintage Cars	1938 Cadillac V-16 Presidential Limousine	43	$1,676
Sub Total (France)			*270*	*$22,314*
USA				
	Classic Cars	1957 Ford Thunderbird	194	$15,740
	Vintage Cars	1917 Grand Touring Sedan	254	$16,902
Sub Total (USA)			*448*	*$32,642*
Grand Total			*718*	*$54,956*

Figure 7-1 Example of a summary table

Although each attribute column must be associated with a dimension column for it to be useful in a summary table, you do not need a dimension column to create a meaningful summary table. The example shown in Figure 7-2 displays order totals, the measure column, for each state, the attribute column, and a grand total for all states.

STATE	SUM (ORDERTOTAL)
Victoria	$180,585
	$180,887
	$104,225
NV	$164,670
Grand Total	*$630,367*

Figure 7-2 Summary table without groups displaying the grand total

The example shown in Figure 7-3 displays the grand total for two peer attribute columns and a measure column. Because these are peer attribute columns, and the summary table does not contain the dimension field they are associated with, data is not grouped, and subtotals are not displayed. The classification of data fields into dimensions, attributes, and measures is defined in the data set at the time of developing the template, information object, or data object.

PRODUCTNAME	PRODUCTCODE	SUM (QUANTITYORDERED)
1965 Aston Martin DB5	S18_1589	767
1996 Moto Guzzi 1100i	S10_2016	283
P-51-D Mustang	S18_2581	192
1938 Cadillac V-16 Presidential Limousine	S24_2022	439
Grand Total		*1681*

Figure 7-3 Grand totals displayed for attribute columns

Creating a summary table

To create a summary table, complete the following steps:

1 Select a template.

2 Select one of the following data sources:

- A data set included in a template
- An information object
- A data object

3 Select the dimension, attribute, and measure fields to use in the summary table.

As you did when creating a detail report, you use Table Builder to select the data fields for the summary table, and specify the order of appearance of the selected fields. Using Table Builder, you can also specify the following information:

- Group date-and-time dimension columns by an interval.
- Specify one or more aggregate functions to use for the selected measure columns.
- Optionally create a filter condition at the data-set level on any dimension or attribute column to limit the data displayed in the summary table.

The following section describes the process to create a summary table using BIRT Studio.

How to create a summary table

The process to create a summary table using a data set in a template, an information object, or a data object data source is the same. This section describes how to create a summary table using any of these data sources.

1 Launch BIRT Studio, select a template, then select a data source. For more information, see "Using a data source," in Chapter 1, "Getting started."

2 Table Builder appears. In Table Builder, complete the following steps:

  1 In Table Builder—Data, in Fields, select Summarize to create a summary table. Current Measure Selections is highlighted.

2 In Fields, in Available Data:

1 Press Ctrl and select each dimension and attribute data field to insert in the report design. Then choose the right arrow.

2 Press Ctrl and select the measure fields to insert in the report design. Then choose the right arrow.

The selected data fields, Order Date, and Product Name, appear in Current Column Selections. The selected measure fields, Order Total, and Quantity Ordered appear in Current Measure Selections, as shown in Figure 7-4. Because Product Name is an attribute of the Product Line dimension, the data field is listed as a subcategory of the Product Line dimension in Current Column Selections.

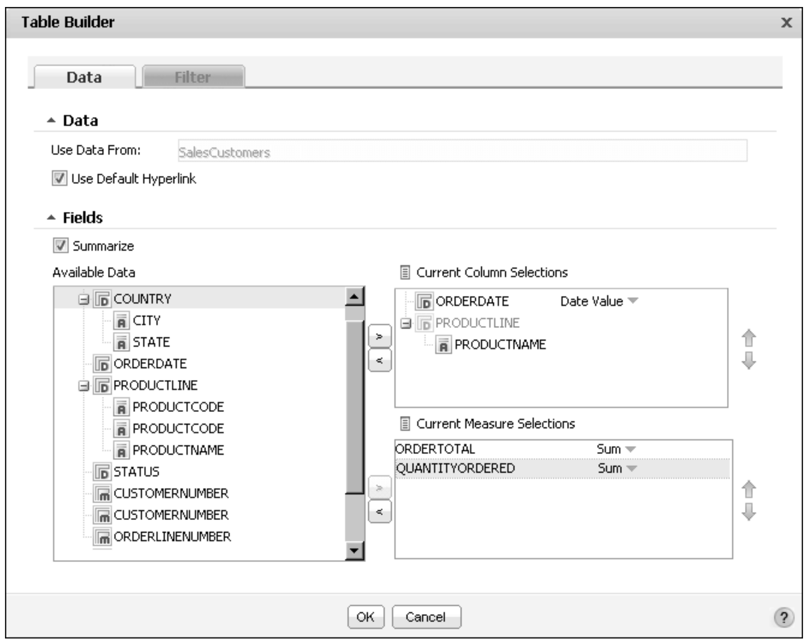


Figure 7-4 Table Builder displaying the selected data fields

3 Use the up and down arrows to modify the order in which you want the selected fields to appear in the report design.

4 For each selected dimension field containing date-and-time data, you can specify intervals by which to group the data. Select an available interval from the Date Value drop-down list. Figure 7-5 shows the Order Date field grouped in Quarters.

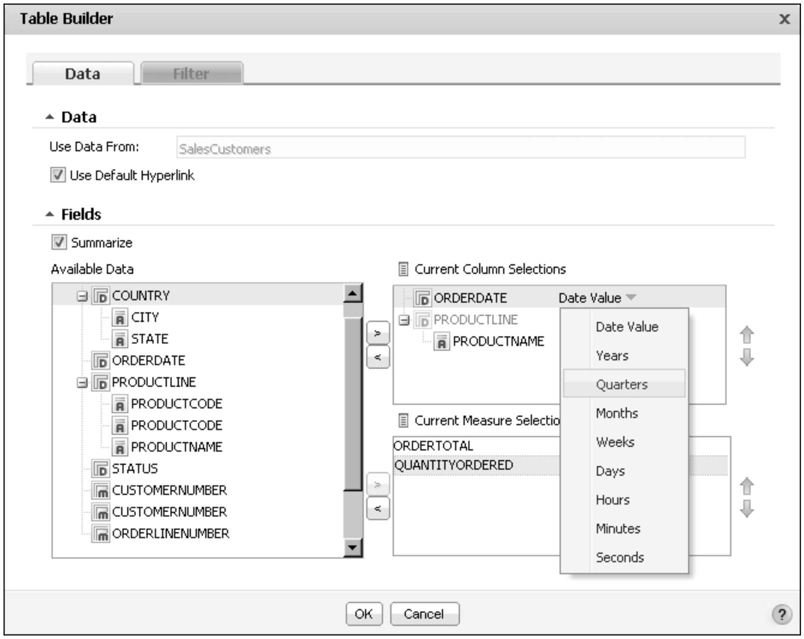


Figure 7-5 Grouping date dimension column in quarterly intervals

5 For every measure field containing numeric, or string data, select an aggregate function to apply to the data from the drop-down list. For columns containing numeric data, the default aggregation function selected is Sum. To aggregate a measure field by an additional function, do the following:

1 Select the field once again in Available Data. Then select the right-arrow so that the measure field appears once again in Current Measure Selections.

2 Use the menu to choose a new aggregate function to apply to the field. Figure 7-6 shows the average function applied to the Quantity Ordered field.

6 To limit the data displayed in the report, you can create filter conditions at the data-set level for dimension and attribute fields in a summary table. For more information see “How to limit the data displayed in a report,” in Chapter 1, “Getting started.”

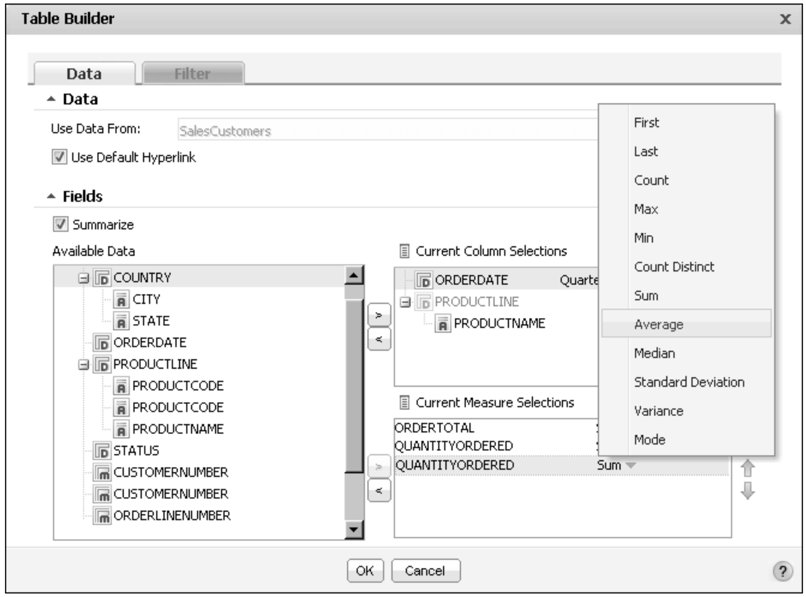


Figure 7-6 Aggregating measure columns

Choose OK. BIRT Studio displays the selected data fields in a summary table, as shown in Figure 7-7. You are now ready to modify the summary table.

ORDERDATE	PRODUCTNAME	SUM (ORDERTOTAL)	SUM (QUANTITYORDERED)	AVE (QUANTITYORDERED)
Q2, 2003				
	1965 Aston Martin DB5	$30,311	350	35.00
	1996 Moto Guzzi 1100i	$21,815	212	35.33
	P-51-D Mustang	$16,927	192	32.00
	1917 Grand Touring Sedan	$20,929	332	36.89
Sub Total (Q2, 2003)		*$89,981*	*1086*	*34.81*
Q3, 2004				
	1993 Mazda RX-7	$33,348	318	31.80
	1996 Moto Guzzi 1100i	$6,067	71	35.50
Sub Total (Q3, 2004)		*$39,415*	*389*	*33.65*
Q4, 2004				
	1949 Jaguar XK 120	$9,631	99	24.75
	1938 Cadillac V-16 Presidential Limousine	$6,236	107	35.67
Sub Total (Q4, 2004)		*$15,867*	*206*	*30.21*
Grand Total		*$145,263*	*1681*	*33.37*

Figure 7-7 Summary table report design

Modifying a summary table

Using BIRT Studio, you can work with summary tables to format data, organize data in groups, sort data, create filters, and insert charts as you would for a detail table. This section describes the actions you can perform on dimension columns, attribute columns, measure columns, and each column header. Table 7-1 lists the actions you can perform on dimension columns, measure columns, attribute columns, column headers, and the entire report table.

Table 7-1 Modifying data in a summary table

Action	Dimension column	Attribute column	Measure column	Column header	Report table
Advanced filtering					✓
Advanced sorting	✓		✓		✓
Create a new computed column			✓		
Create aggregate data			✓		
Create borders				✓	✓
Create data-set filters					✓
Create filters	✓	✓	✓		✓
Create groups	✓				
Create page breaks before or after a group	✓				
Create sections					
Delete a group	✓				
Delete the column	✓	✓	✓		
Disable default hyperlinks					✓
Edit the column header text				✓	
Filter top/bottom N			✓		
Format data	✓	✓	✓		
Group date-and-time data in intervals	✓				
Hide column					

Table 7-1 Modifying data in a summary table

Action	Dimension column	Attribute column	Measure column	Column header	Report table
Hide details for a group	✓				
Insert a chart		✓			✓
Reorder columns	✓	✓	✓		
Sort column data	✓	✓	✓		✓
Specify alignment properties	✓	✓	✓	✓	✓
Specify column width	✓	✓	✓		
Specify conditional formatting rules			✓		
Specify font properties	✓	✓	✓	✓	✓

Modifying the report table

You can perform the following actions at the report-table level in a summary table:

- Insert a chart at the report table level.
- Hide a table, and display the corresponding chart.
- Sort data.
- Specify font properties.
- Specify alignment properties for column data.
- Format data types.
- Add a border for the table.
- Add data fields from the data set.
- Reorder columns.
- Create a bookmark to the summary table.

To add a table border, add data fields, insert a chart, reorder columns, or create a bookmark to the summary table, choose an option from the context menu of the report table, shown in Figure 7-8.

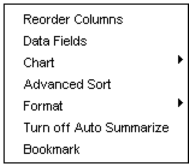


Figure 7-8 Modifying the report table

Modifying labels

You can perform the following actions on a column header, or an aggregate label, in a summary table:

- Edit the header or label text.
- Modify the alignment of label text.
- Format font properties.
- Create a border for the label.

To modify a column header, or an aggregation label select the column header, and choose an option from the context menu. The context menu that appears looks like the one shown in Figure 7-9.

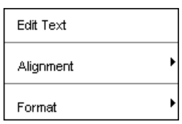


Figure 7-9 Modifying a column header

Working with a dimension column

If a report has one or more dimension columns, the order in which you select and insert the columns determines the group level. For example, if you select and insert Product Line before Country, Product Line is a higher level, or outer group. Country is the lower level, or inner group.

A grouped dimension column displays subtotals for the data in each group. When you group a dimension column, BIRT Studio eliminates duplicate data values when calculating aggregate data. Figure 7-10 displays a summary table containing three dimension columns. The Order date column is grouped in quarterly intervals, and is located in the innermost position. When a summary table contains more than one dimension column, data in the outer dimension columns can be grouped, as shown in Figure 7-10, and subtotals are displayed for each value in the grouped column.

COUNTRY	PRODUCTLINE	ORDERDATE	SUM (QUANTITYORDERED)	SUM (ORDERTOTAL)
Australia				
	Motorcycles			
		Q2, 2003	212	$21,815
	Sub Total (Motorcycles)		*212*	*$21,815*
	Planes			
		Q2, 2003	192	$16,927
	Sub Total (Planes)		*192*	*$16,927*
	Vintage Cars			
		Q2, 2003	78	$4,026
	Sub Total (Vintage Cars)		*78*	*$4,026*
Sub Total (Australia)			*482*	*$42,768*

Innermost grouped date dimension

Figure 7-10 Grouping the innermost date-and-time dimension column

When working with a dimension column that contains date-and-time data, you can group the data by Intervals such as day, week, month, quarter, or year. When placed in the innermost position, dimension columns that contain other types of data cannot be grouped, as shown in Figure 7-11. If a dimension column contains date-and-time data, you can always group data in the column by intervals.

COUNTRY	ORDERDATE	PRODUCTLINE	SUM (QUANTITYORDERED)	SUM (ORDERTOTAL)
Australia				
	Q2, 2003			
		Motorcycles	212	$21,815
		Planes	192	$16,927
		Vintage Cars	78	$4,026
	Sub Total (Q2, 2003)		*482*	*$42,768*
Sub Total (Australia)			*482*	*$42,768*

Innermost ungrouped dimension

Figure 7-11 Grouping dimension columns

The innermost dimension column does not display subtotals, irrespective of the type of data in the column, as shown in Figure 7-10 and Figure 7-11. You also cannot create filters for the data in the innermost dimension column.

For a summary table containing a single dimension column, and one or more associated attribute columns, you cannot group data in the dimension column. No subtotals appear. The summary table displays a grand total value for all Product Lines, as shown in Figure 7-12.

PRODUCTLINE	PRODUCTNAME	SUM (ORDERTOTAL)
Classic Cars	1965 Aston Martin DB5	$73,290
Motorcycles	1996 Moto Guzzi 1100i	$27,881
Planes	P-51-D Mustang	$16,927
Vintage Cars	1938 Cadillac V-16 Presidential Limousine	$27,165
Grand Total		*$145,263*

Figure 7-12 A summary table displaying a grand total

The preceding summary table results from the data object shown in Figure 7-13, which contains the Product Line dimension and two associated attribute fields, Product Code and Product Name.

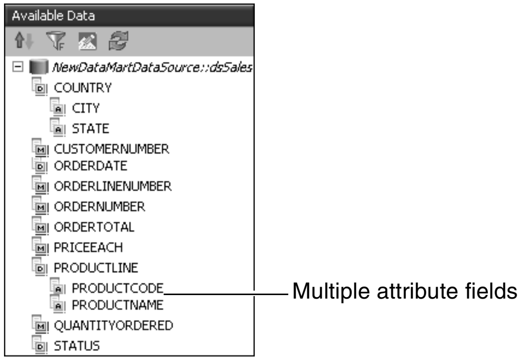


Figure 7-13 Examining a data object containing a dimension with multiple attribute fields

Modifying a dimension column

You can perform the following actions on a dimension column:

- Delete a group.
- Hide detail rows for a grouped column.
- Specify page breaks before or after a group.

To modify a dimension column, select the column, and choose an option from the context menu. An example context menu for a dimension column, shown in Figure 7-14, displays the available options.

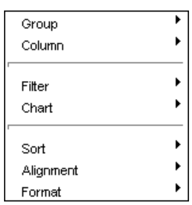


Figure 7-14 Modifying a dimension column

Moving a dimension column

When you move a dimension column in a summary table, you can move it only among other dimension columns. Dimension columns and associated attribute columns always appear preceding the measure columns in a summary table.

Dimension columns can also contain associated data fields, called attributes. You cannot move a dimension column independently of its associated attribute columns.

Deleting a dimension column

To delete a dimension column, select the column, and choose Delete. When you delete a dimension column containing associated attributes from a summary table, the attribute fields are not deleted.

Working with an attribute column

Attributes represent subcategories of a broader category, or dimension. For example, Product Code, and Product Name, can be attributes of a dimension, Product Line. For an attribute to be useful in a summary table, it must be associated with a dimension field in the data set. The developer typically sets this property at the time of designing the information object or data object.

Multiple attribute columns associated with a single dimension field are called peer attribute columns. In a summary table that contains one or more dimension columns, and associated attributes, data in the attribute columns cannot be grouped and does not display subtotals. In a summary table that contains one or more peer attribute columns, and no dimension columns, data in the attribute columns cannot be grouped and does not display subtotals. In a summary table that contains one or more non-peer attribute columns and no dimension columns, data in the outermost attribute column is grouped and displays subtotal aggregate values.

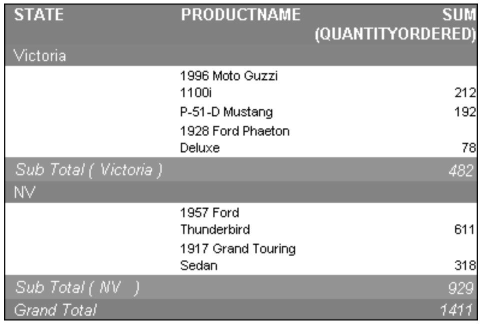
Inserting an attribute column

Attribute columns appear following the associated dimension column in a summary table. When you insert an attribute column in a summary table containing the associated dimension column, the attribute appears after the dimension column. The example in Figure 7-15 shows a dimension column, Country, an associated attribute column, State, and a measure column Quantity Ordered. There are no grouped columns in the summary tables and BIRT Studio displays aggregate values for the entire table.

COUNTRY	STATE	SUM (QUANTITYORDERED)
Australia	Victoria	482
USA	NV	929
Grand Total		*1411*

Figure 7-15 Dimension, associated attribute and measure columns

When you insert an attribute column in a summary table that does not contain an associated dimension column, or other attribute columns, the attribute column is inserted to the left in the report table, but cannot be grouped if it is the only attribute column in the summary table. When you insert one or more non-peer attribute columns in a summary table with no dimension columns, you can group data in the outermost attribute column, and subtotal values are displayed for this grouped column. The example in Figure 7-16 shows two non-peer attribute columns, State and Product Code and one measure column. Data in the State column is grouped, and aggregate data values are displayed for each group.



STATE	PRODUCTNAME	SUM (QUANTITYORDERED)
Victoria		
	1996 Moto Guzzi 1100i	212
	P-51-D Mustang	192
	1928 Ford Phaeton Deluxe	78
Sub Total (Victoria)		*482*
NV		
	1957 Ford Thunderbird	611
	1917 Grand Touring Sedan	318
Sub Total (NV)		*929*
Grand Total		*1411*

Figure 7-16 Attribute and measure columns

When you insert an attribute column in a summary table containing a dimension column that is not associated with it, or the other attribute columns in the summary table, the new column appears in the position you select, as shown in Figure 7-17.

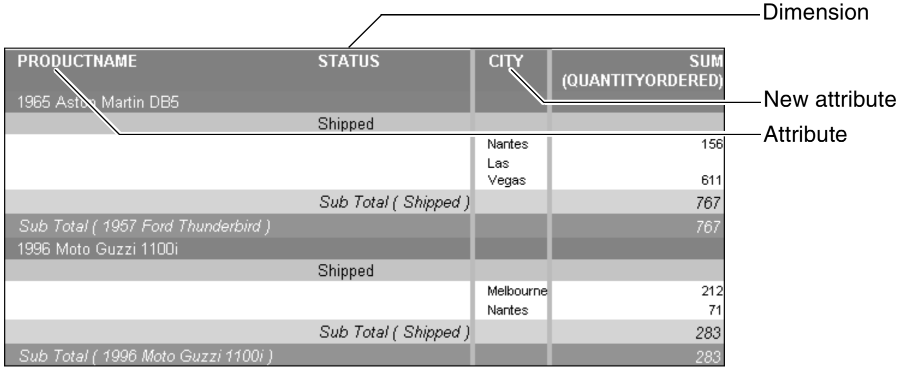


Figure 7-17 Inserting a non-associated attribute field in a summary table

Modifying an attribute column

You can modify an attribute column to change column width, reorder columns, create filters, sort data, modify alignment, format data, and change font properties. To modify an attribute column, select the column, and choose an option from the context menu, shown in Figure 7-18.

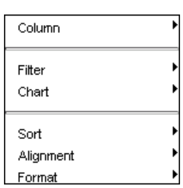


Figure 7-18 Modifying an attribute column

Moving an attribute column

In a summary table containing multiple dimensions and associated attribute columns, you can reorder attribute columns only among peer attribute columns. To move an attribute column, select the column, and choose Reorder Columns from the context menu. When you move a dimension column with its associated attribute columns in a summary table, the columns are placed in their new position as a single unit. You cannot move the dimension column without moving the associated attribute columns.

Deleting an attribute column

To delete an attribute column, select the column, and choose Delete. When a dimension column containing associated attributes is deleted from a summary table, the attributes remain in the summary table. To delete each attribute column select each column, and choose Delete.

Working with a measure column

Every summary table must contain at least one measure column to display aggregate data. The developer typically sets default aggregation functions for each measure column when creating the information object or data object. You can modify the default aggregate function and add additional aggregate functions to data in measure columns.

Modifying a measure column

You can perform the following actions on a measure column:

- Create a new computed column. For more information about creating a computed column see “Creating a computed column,” in Chapter 3, “Inserting calculated data.”
- Specify conditional formatting rules. For more information about conditional formatting rules, see “Formatting data based on conditions,” in Chapter 2, “Editing and formatting report content.”
- Use the following numeric and non-numeric subtotal functions to modify aggregate data:
  - First
  - Last
  - Count value
  - Max
  - Min
  - Sum

- Average
- Median
- Standard deviation
- Variance
- Mode

For more information about aggregate functions, see "Functions used in aggregate calculations," in Chapter 9, "Functions and operators."

To modify a measure column, select the column, and choose an option from the context menu, shown in Figure 7-19.

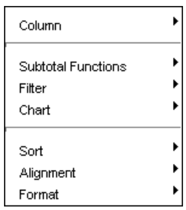


Figure 7-19 Modifying a measure column

Adding an aggregate calculation

You can apply multiple aggregate functions to a single measure column in a summary table. To add an aggregate function, complete the following steps:

1. In Available Data, select the measure column, and choose Insert.
2. In Select Subtotal Function, as shown in Figure 7-20, select an aggregate function from the drop-down list.

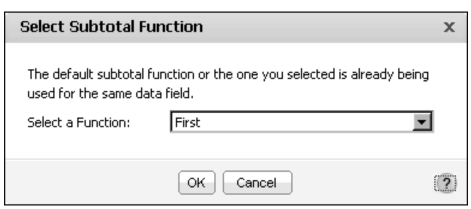


Figure 7-20 Adding an aggregate function

3. Choose OK. The measure column you selected appears in the report design displaying aggregate values for the new aggregate function you defined.

Moving a measure column

When you move a measure column in a summary table, you can move it among other measure columns. Measure columns always appear to the right of dimension and attribute columns in a report design.

Deleting a measure column

To delete a measure column, select the column, and choose Delete.

Formatting aggregate values

You can format aggregate data values displayed as subtotals, or grand totals as follows:

- Specify the alignment properties.
- Specify a font type, color, or style.
- Specify an available or custom data format to the numeric data values.

To modify alignment options for subtotals or grand total values, select the value, and choose Alignment➛Align Left, Align Right, or Align Center from the context menu.

To modify font properties for subtotals or grand total values, select the value, and choose Format➛Font from the context menu. In Font, use the drop-down lists to specify the font type, color, and style.

To apply a number format to the subtotals or grand total values, select the value, and choose Format➛Format Data from the context menu. In Number Column Format, select a format from the list of available formats.

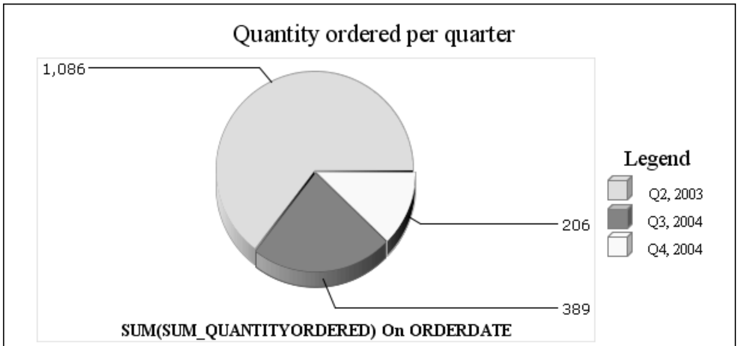
Figure 7-21 displays an example of the context menu for an aggregate data cell.

Figure 7-21 Choosing formatting options from the context menu

Inserting a chart in a summary table

You can insert standard charts at the table level in a summary table. The types of charts you can use in a summary table are the same as those available for use in a detail table. BIRT Studio assigns the outermost grouped column as the category or *x*-axis, by default. You can select the value series on the *y*-axis from the available measure columns in the summary table. In addition, you can specify a Tooltip, and format the chart by specifying a title, font properties, size, dimension, and legend properties, as you do for standard charts.

The example in Figure 7-22 displays a pie chart showing the total product quantities ordered each quarter. The outermost grouped column is Order date, which BIRT Studio sets as the category axis. You can choose either of the two measure columns as the *y*-axis, or both to display multiple value series.



ORDERDATE	COUNTRY	STATUS	SUM (QUANTITYORDERED)	SUM (ORDERTOTAL)
Q2, 2003				
	Australia			
		Shipped	482	$42,768
	Sub Total (Australia)		*482*	*$42,768*
	France			
		Shipped	156	$14,571
	Sub Total (France)		*156*	*$14,571*
	USA			
		Shipped	448	$32,642
	Sub Total (USA)		*448*	*$32,642*
Sub Total (Q2, 2003)			*1086*	*$89,981*

Figure 7-22 A summary table displaying a pie chart

For more information about charts, see Chapter 8, "Presenting data in a chart."

Changing table types

In BIRT Studio, you can convert a summary table to a detail table, or a detail table to a summary table. When you change table types, the data in the original report table is lost, and must be recreated by reinserting the selected data fields.

How to convert a summary table to a detail table

1 In BIRT Studio, choose Turn off Auto Summarize.

A message appears and informs you that if you choose to continue, all data in the current table will be lost, as shown in Figure 7-23. Choose OK.

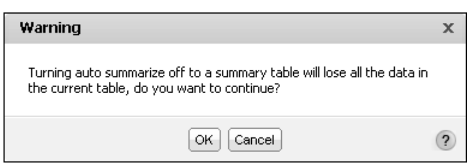


Figure 7-23 Warning dialog

2 Choose Insert to reinsert the selected fields to create a new report design. BIRT Studio displays the chosen data fields in a detail table.

How to convert a detail table to a summary table

1 In BIRT Studio, choose Turn on Auto Summarize.

A message appears and informs you that if you choose to continue, all data in the current table will be lost, as shown in Figure 7-24. Choose OK.

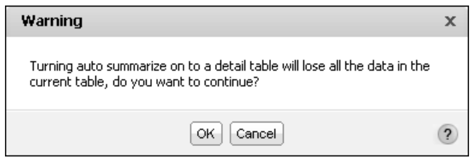


Figure 7-24 Warning dialog

2 Choose Insert to reinsert the selected fields to create a new report design. BIRT Studio displays the chosen data fields in a summary table.

Chapter

8

Presenting data in a chart

This chapter contains the following topics:

- About charts
- Creating a chart
- Inserting a chart
- Choosing a chart type
- Selecting data for a chart
- Formatting a chart
- Working with Flash charts
- Creating a Flash chart
- Selecting data for a Flash chart
- Formatting a Flash chart
- Deleting a Flash chart
- Limitations
- Displaying a chart without the table data

About charts

A chart is a graphical representation of data. Charts are particularly useful for summarizing numeric data and showing the relationship between sets of values, called series. For example, a chart can show sales by region, average temperatures by month, or the price of a stock over three months.

Because a chart presents a picture, it reveals trends that are not as apparent in a table. Figure 8-1 shows an example of a table and a bar chart that display sales data. The chart shows instantly the sales trend for the year.

Month	Sales
Jan	29000
Feb	25000
Mar	42700
Apr	33250
May	40500
Jun	45000
Jul	44000
Aug	35000
Sep	42000
Oct	37000
Nov	35000
Dec	22000

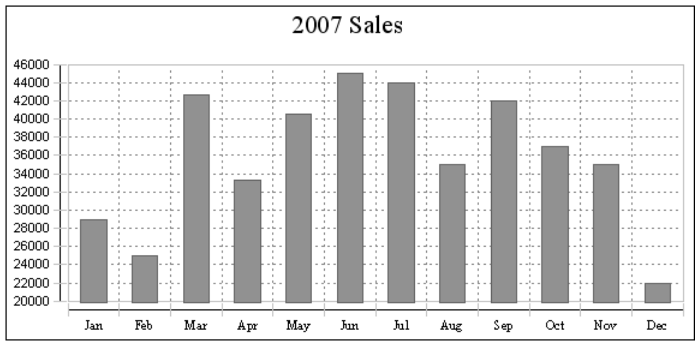


Figure 8-1 Presenting sales data in a table and in a chart

Figure 8-2 shows an example of a pie chart that displays profits by product line. The table in Figure 8-3 provides the data for the pie chart. Typically reports display detail data in a table and summary data in a chart, especially if the detail data spans multiple pages.

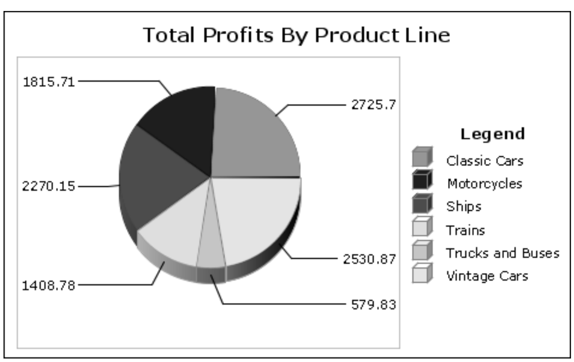


Figure 8-2 Displaying profits by product line in a pie chart

Sales Office	San Francisco	
Product Line	Classic Cars	
Product Name	Total	Profit
1961 Chevrolet Impala	$1,766.40	$1,022.81
1970 Plymouth Hemi Cuda	$1,742.88	$976.80
1971 Alpine Renault 1600s	$1,405.35	$363.69
1982 Lamborghini Diablo	$687.20	$362.40
	$5,601.83	*$2,725.70*
Product Line	Motorcycles	
Product Name	Total	Profit
1957 Vespa GS150	$2,238.30	$755.55
1982 Ducati 900 Monster	$1,948.22	$582.32
1982 Ducati 996 R	$1,274.46	$477.84
	$5,460.98	*$1,815.71*
Product Line	Ships	
Product Name	Total	Profit
Pont Yacht	$1,831.44	$532.74
The Schooner Bluenose	$2,268.94	$874.94
The Titanic	$1,935.36	$862.47
	$6,035.74	*$2,270.15*
Product Line	Trains	
Product Name	Total	Profit
1962 City of Detroit Streetcar	$2,362.08	$787.50
Collectable Wooden Train	$2,107.60	$621.28
	$4,469.68	*$1,408.78*
Product Line	Trucks and Buses	
Product Name	Total	Profit
1996 Peterbilt 379 Stake Bed with Outrigger	$1,352.86	$579.83
	$1,352.86	*$579.83*
Product Line	Vintage Cars	
Product Name	Total	Profit
1937 Lincoln Berline	$2,080.50	$565.00
1938 Cadillac V-16 Presidential Limousine	$1,235.97	$597.06
1939 Cadillac Limousine	$2,163.50	$1,006.50
1939 Chevrolet Deluxe Coupe	$1,242.54	$362.31
	$6,722.51	*$2,530.87*
	$29,643.60	*$11,331.04*
	$29,643.60	*$11,331.04*

Figure 8-3 The table that provides the data for the pie chart

Creating a chart

The data for a chart comes from the data in a table. You must create the table before you create a chart. Users typically design reports using this sequence. To display only a chart in your report, hide the table after you create the chart.

When you create a chart, you perform the following tasks:

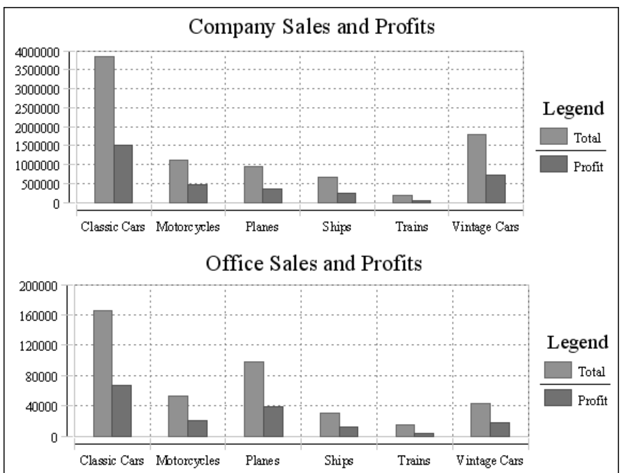
1. Insert a chart.
2. Choose a chart type.
3. Specify the data to present in the chart.
4. Format the chart.

Inserting a chart

The location in which you insert a chart determines what data the chart displays. You can insert a chart in the following locations:

- A table. A chart at the table level summarizes data for the entire table, and the chart appears at the top of every page.
- A section header, if the report organizes data in sections. A chart at the section-header level summarizes data for the section, and the chart appears at the top of each section. If a section's data spans multiple pages, the chart appears on every page.

If your report design organizes data in groups instead of sections, you can create only one chart per table. Figure 8-4 shows one page of a 76-page report that displays a chart at the table level and a chart at the level of the sales office section.



Sales Office		Tokyo	
Product Line		Trains	
Order Number	Product Name	Total	Profit
10117			
	1962 City of Detroit Streetcar	$1,168.65	$381.36
	Collectable Wooden Train	$1,715.28	$296.52
10165			
	1950's Chicago Surface Lines Streetcar	$2,433.20	$1,257.52
	1962 City of Detroit Streetcar	$1,869.98	$445.36
	Collectable Wooden Train	$4,235.50	$857.50
10372			
	1962 City of Detroit Streetcar	$1,363.68	$463.92
	Collectable Wooden Train	$2,294.00	$605.00
Sum	*Trains*	*$15,080.29*	*$4,307.18*

Figure 8-4 Placing charts in different locations in a report design

The upper chart, inserted at the table level, displays sales and profit totals for the company. The lower chart, inserted at the sales office section level, displays sales and profit totals for the Tokyo office.

How to insert a chart

To insert a chart in the report, complete the following tasks:

- To insert a chart at the table level, select the table, then choose Chart.
- To insert a chart at the table-header level, select a column header in the table header row, then choose Chart. If your report design contains sections, selecting a column header does not enable Chart on the toolbar. You must first delete the existing section or sections, select a column header and insert the chart, then recreate the section or sections.
- To insert a chart at a section level, select the section heading, then choose Chart.

After you insert a chart, BIRT Studio displays the chart editor, shown in Figure 8-5. You use the chart editor to select a chart type, select data for the chart, and format the chart.

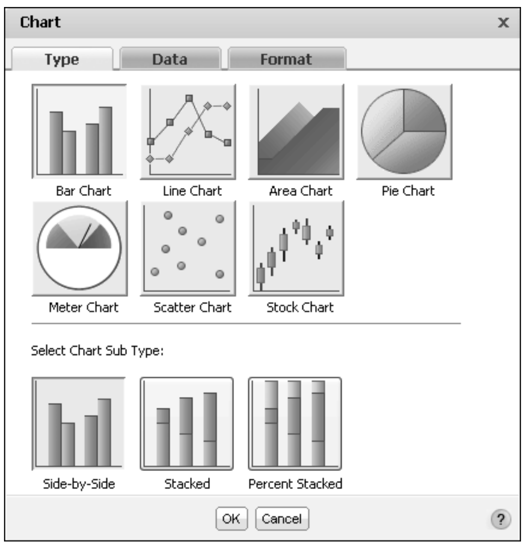


Figure 8-5 The chart editor displaying the Chart—Type page

Choosing a chart type

BIRT Studio provides a wide array of chart types that you can use in a report design. After you insert a chart, the first step is to choose the most suitable chart type for the data. You can use any chart type to display most data, but the data is easier to read when you choose a chart type that has the best visual characteristics for a particular set of data.

For example, to show what percentage each product line contributes to a company's total sales, use a pie chart, which is ideal for showing how parts relate to a whole. To compare the sales of each product line in the current year and the previous year, use a bar chart, which supports side-by-side comparisons. To show how each product line has been selling over the course of five years, a line chart is appropriate for displaying trends in linear data.

Another item to consider when choosing a chart type is the number of data values to display. Some charts, such as scatter charts, reveal trends more clearly when there are more data values. Other charts, such as pie charts, are more effective when there are fewer data values. For example, the pie chart in Figure 8-6 is difficult to read, because it displays too many data values.

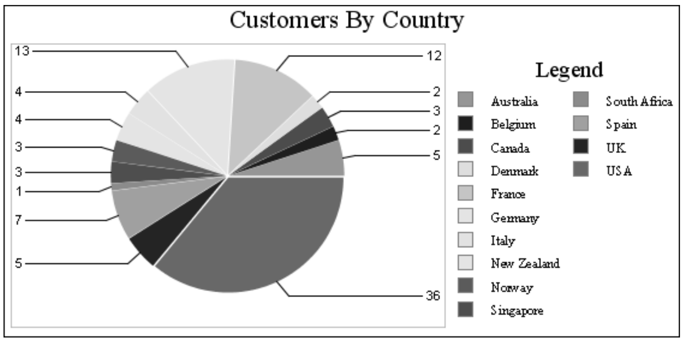


Figure 8-6 A pie chart that displays too many data values

The following sections describe the chart types BIRT Studio supports. Several of the chart types include subtypes.

About area charts

An area chart displays data values as a set of points, connected by a line, with the area below the line filled. You typically use an area chart to present data that occurs over a continuous period of time. There are three types of area charts, described in the following sections.

Stacked area chart

In a stacked area chart, multiple series are stacked vertically, as shown in Figure 8-7. The example shows that the stacked area chart is suitable for the data, because the chart displays totals for all series as well as the proportion that each series contributes to the total. The height of the top line shows the total value for each quarter. Each shaded area represents the sales amount for a specific region.

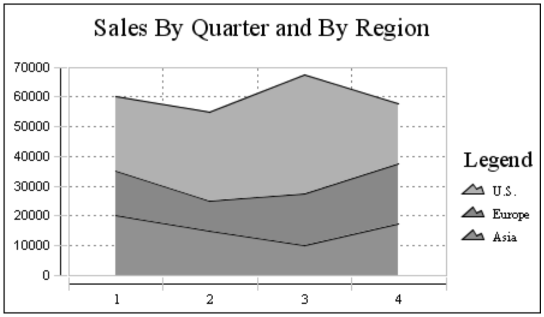


Figure 8-7 A stacked area chart

Overlay area chart

In an overlay area chart, the areas of multiple series overlap, as shown in Figure 8-8. As the example shows, this chart subtype is not suitable for showing multiple series if the data values overlap. In the example, the data for the U.S. obscures the data for Europe and Asia, because the U.S. numbers are the highest for every quarter. Use the overlay area chart if you want to show only one series, for example, only sales for Asia.

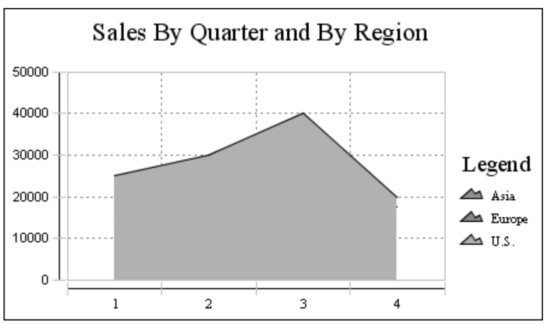


Figure 8-8 An overlay area chart

Percent stacked area chart

In a percent stacked area chart, multiple series are stacked vertically, and the values appear as a percentage of the whole. As Figure 8-9 shows, the sales values are displayed in percentages, instead of the actual numbers, as shown in the previous area charts.

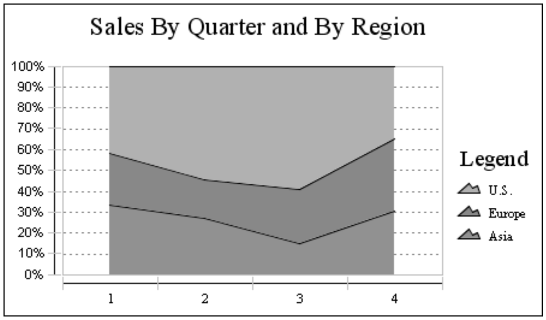


Figure 8-9 A percent stacked area chart displaying three series

The percent stacked area chart is meaningful only when displaying and comparing multiple series. If displaying only one series, the percent stacked area chart looks like the example in Figure 8-10. The sales percentage of one region, compared to the whole, is 100%.

Figure 8-10 A percent stacked area chart displaying one series

About bar charts

A bar chart, by default, displays data values as a set of vertical bars, but you can transpose the axes to display horizontal bars. A bar chart is useful for displaying data side-by-side for easy comparison. There are three subtypes of bar charts. Two of the subtypes, stacked bar chart and percent stacked bar chart, are functionally similar to the stacked area chart and percent stacked area chart subtypes.

Side-by-side bar chart

In a side-by-side bar chart, multiple series appear as side-by-side bars, as shown in Figure 8-11. This bar chart uses the same data as the area charts shown in the earlier sections.

Figure 8-11 A side-by-side bar chart

Stacked bar chart

In a stacked bar chart, multiple series are stacked vertically, as shown in Figure 8-12. The stacked bar chart shows totals for each category, each quarter in this example, as well as the proportion that each series contributes to the total.

Figure 8-12 A stacked bar chart

Percent stacked bar chart

In a percent stacked bar chart, multiple series are stacked vertically, and the values are shown as a percentage of the whole. As you can see in Figure 8-13, the sales values are shown in percentages instead of the actual numbers as shown in the previous bar charts.

Like the percent stacked area chart, the percent stacked bar chart is meaningful only when displaying and comparing multiple series. Do not use this chart subtype if you are displaying only one series, for example, only sales for Asia. Figure 8-14 shows how a percent stacked bar chart looks when it displays only one region series. All the bars show a value of 100%, because that is the sales percentage of one region when it is compared to itself.

Figure 8-13 A percent stacked bar chart displaying three series

Figure 8-14 A percent stacked bar chart displaying one series

Horizontal bar chart

For all the bar chart subtypes, you can transpose, or flip, the axes to display data as horizontal bars. Figure 8-15 shows an example of a side-by-side bar chart with flipped axes. This feature is useful for showing data that contains many categories. For example, if you want to show sales numbers for five regions for 12 months, displaying the data requires 60 bars, which is crowded for a side-by-side bar chart that displays data as vertical bars.

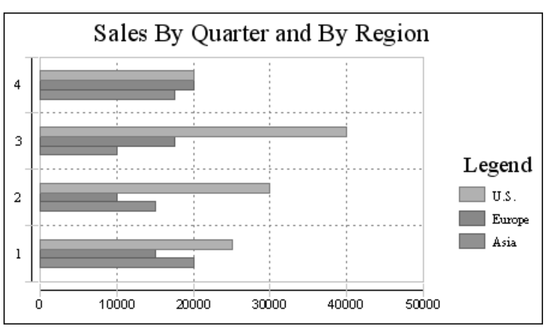


Figure 8-15 A horizontal bar chart

About line charts

A line chart displays data values as a set of points that are connected by a line. You typically use line charts to present large amounts of data that occur over a continuous period of time. A line chart is the most basic type of chart in finance. Figure 8-16 shows an example of a line chart that shows the value of the Euro against the US Dollar over 10 days.

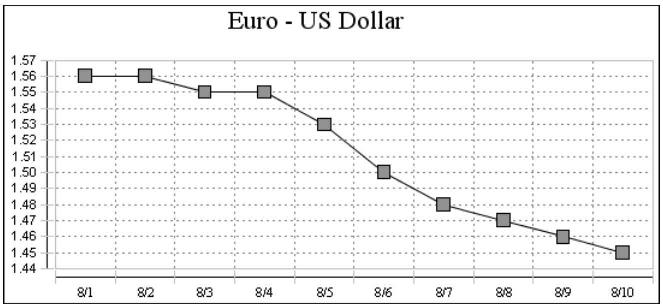


Figure 8-16 A line chart

A line chart is similar to an area chart, except that the line chart does not fill in the area below the line, and it uses a square marker for each data value.

There are three subtypes of line charts, which are functionally similar to the area chart and bar chart subtypes. The line chart subtypes in the following sections use the same data as the area charts and bar charts in earlier sections. You can compare how different chart types present the same set of data.

Overlay line chart

In an overlay line chart, multiple series appear as overlapping lines, as shown in Figure 8-17. A square marker indicates each data value.

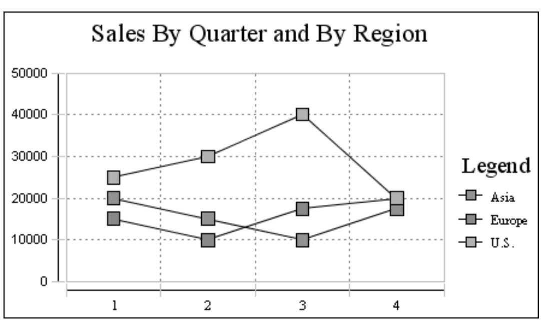


Figure 8-17 An overlay line chart

Stacked line chart

In a stacked line chart, multiple series are stacked vertically, as shown in Figure 8-18. The stacked line chart shows totals for each series, as well as the proportion that each series contributes to the grand total. In the example, the top line shows the total sales amounts for each quarter. The empty area between each line represents the sales amount for a region.

The stacked line chart is not as effective as the stacked area chart for comparing the amount that each region series contributes to the total, because the areas are not filled. In a stacked area chart, the filled-in areas provide a clear visual cue that each part is compared to the whole.

In addition, as the example shows, a user can easily misinterpret the data in a stacked line chart. There is no obvious indication that the top line shows the total sales amount for each quarter, and the middle line shows the difference in the sales amount between Europe and Asia. A user looking at this chart can mistakenly think that the top line represents the sales data for the U.S., the middle line represents the sales data for Europe, and the bottom line represents the sales data for Asia.

Figure 8-18 A stacked line chart

Percent stacked line chart

In a percent stacked line chart, multiple series are stacked vertically and the values are shown as a percentage of the whole. As shown in Figure 8-19, the sales values appear in percentages instead of numbers. Like the percent stacked area chart, the percent stacked line chart makes sense only when displaying and comparing multiple series. Do not use this chart subtype if you are displaying only one series, for example, only sales for Europe.

Like the stacked line chart, the percent stacked line chart is not as effective as its counterpart, the percent stacked area chart, for the reasons cited in the previous section.

Figure 8-19 A percent stacked line chart

About meter charts

A meter chart displays a value as a needle pointer on a semicircle, called a dial. As Figure 8-20 shows, a meter chart resembles a speedometer, with tick marks and numbers showing a range of values. In this example, the meter chart displays two pointers that represent two values: projected sales total and actual sales total. Use a meter chart to display a small set of values.

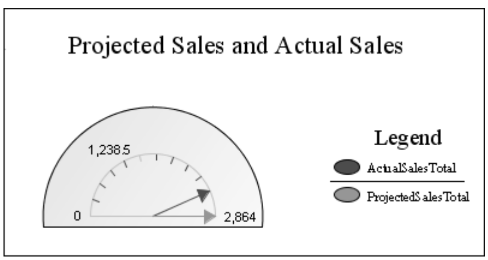


Figure 8-20 A meter chart

Superimposed meter chart

A superimposed meter chart displays multiple values in a single dial, as shown in Figure 8-20. Use the superimposed meter chart type when there are few values to display and when each value is distinct. Duplicate values result in overlapping needles.

Standard meter chart

A standard meter chart displays multiple values in multiple dials, where each dial displays a single value. The meter chart in Figure 8-21 displays the same data as the chart in Figure 8-20, using two dials instead of one.

The standard meter chart typically is used to create a dashboard effect, which can be visually compelling.

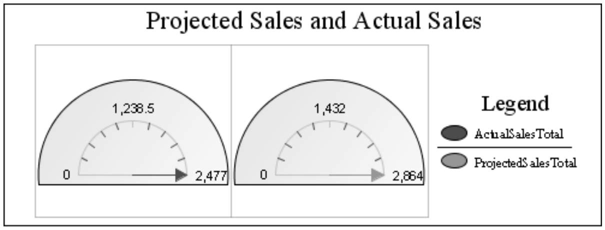


Figure 8-21 A standard meter chart displaying values in two dials

About pie charts

A pie chart is a circular chart that is divided into sectors or slices. Each sector represents a value that is proportional to the sum of the values. Use a pie chart to show the relationship of parts to the whole, for example, the amount each product line contributes to a company's total sales, as shown in Figure 8-22.

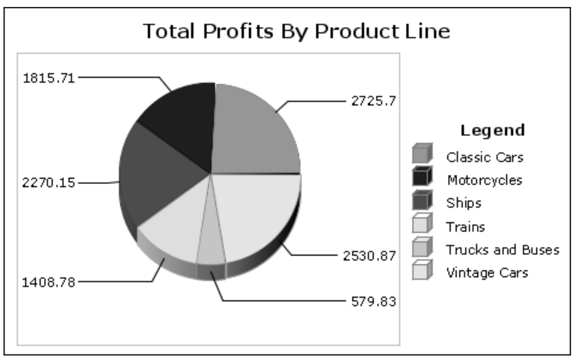


Figure 8-22 A pie chart

About scatter charts

A scatter chart presents data as *x-y* coordinates by combining two sets of numeric values into single data points. A scatter chart typically is used to display scientific and statistical data, because it shows if there is a relationship between two sets of measurements. Use a scatter chart to compare, for example, salaries and years of experience, weight and body fat, rainfall amounts and pollen levels, or test scores and hours of study. The more data values you include in a scatter chart, the clearer the trends the data reveals.

The scatter chart in Figure 8-23 shows the relationship between salary and years of experience. Each pair of values, salary and years of experience, is plotted as a single *x-y* value. The chart reveals a positive, or direct, relationship between salary and years of experience. As the number of years increases, the salary also increases. The chart also displays the salaries earned by men and women. In this example, the chart reveals that men consistently earn more than women.

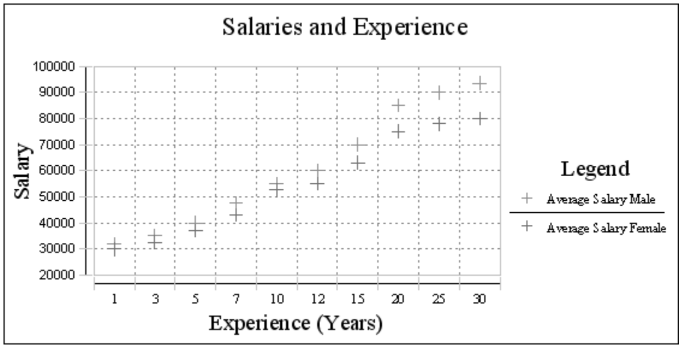


Figure 8-23 A scatter chart

About stock charts

A stock chart displays a stock's open, close, high, and low values for a set of trading dates. A stock chart can show the data for one stock or for multiple stocks. Although a stock chart is typically used to display stock data, you can also use it to chart other values that can be set up in a similar fashion. For example, you can use a stock chart to show four daily temperature values for a set of dates: high, low, sunrise, sunset.

Candlestick stock chart

A candlestick stock chart consists of a series of boxes with lines extending up and down from the ends, as shown in Figure 8-24. The top and bottom points of each line indicate the high and low values, respectively. The top and bottom of each box indicate the open and close values. If the close value is higher than the open value, the box is white. If the open value is higher than the close value, the box is shaded. This style enables you to see immediately whether a stock posted a gain or a loss for a given day. The chart in Figure 8-24 shows that the stock posted a gain every day in the trading week except the fourth day.

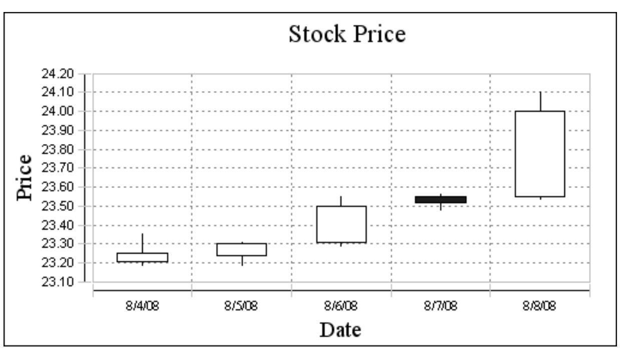


Figure 8-24 A candlestick stock chart

Bar stick stock chart

A bar stick stock chart consists of a series of vertical bars with horizontal tick marks, as shown in Figure 8-25. The top and bottom points of each bar indicate the high and low values, respectively. The horizontal tick marks indicate the open and close values. The tick mark on the left of the bar is the open value. The tick mark on the right of the bar is the close value.

The chart in Figure 8-25 shows the same data as the chart in Figure 8-24. Observe in Figure 8-25 that for every day, except the fourth, the tick mark on the left of the bar is lower than the tick mark on the right. On the fourth day, the tick mark on the left is higher than the tick mark on the right. This difference indicates a stock's gain or loss for a given day. As you can see, the candlestick stock chart shows the gain or loss pattern more clearly than the bar stick stock chart.

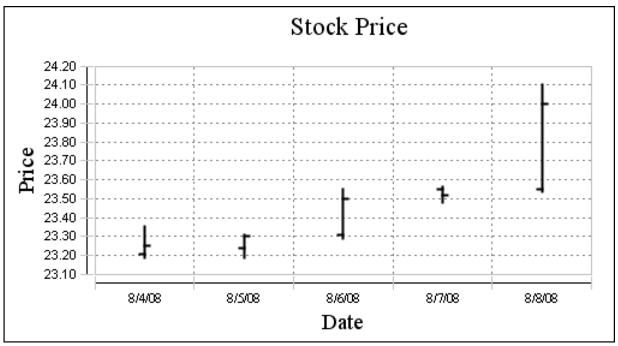


Figure 8-25 A bar stick stock chart

Selecting data for a chart

A chart shows the relationship between sets of values called series. There are two types of series: value and category. A value series contains numeric values, such as prices, sales totals, and salaries. These values determine, for example, the height of a bar in a bar chart or the size of a sector in a pie chart. For charts that use axes to display data, value series values appear on the *y*-axis.

A category series determines how to group the values, for example, by year, by country, or by product line. For charts that use axes, category series values appear on the *x*-axis.

For all chart types, you must select one column as the category series and at least one column as the value series. You can select two columns to display two value series. The bar chart in Figure 8-26 displays two value series, total and profit.

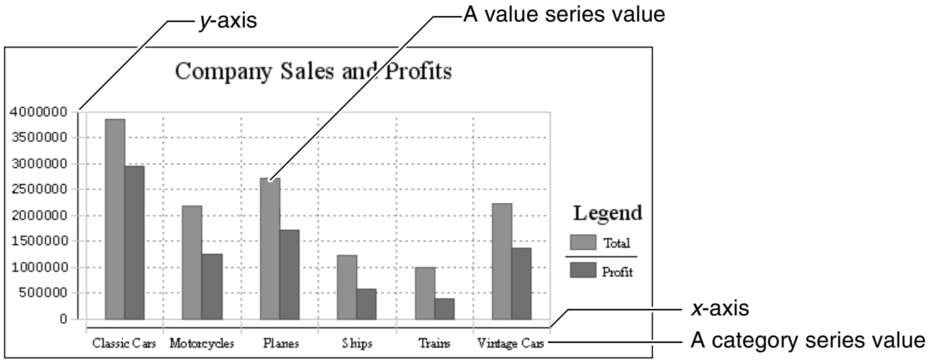


Figure 8-26 Parts of a chart

Then, depending on the table data that a chart uses, you might need to group and aggregate the values, so that the chart can display the data properly. Suppose the table displays the data in Figure 8-27. The table shows sales details by product, product line, and sales office. In Figure 8-27, the table shows only 10 rows of sample data, but the generated report contains 75 pages of data.

Sales Office	San Francisco	
Product Line	Classic Cars	
Product Name	Total	Profit
1970 Plymouth Hemi Cuda	$1,742.88	$976.80
1971 Alpine Renault 1600s	$1,405.35	$363.69
1976 Ford Gran Torino	$5,864.88	$2,778.30
Sum Classic Cars	*$9,700.31*	*$4,481.19*
Product Line	Motorcycles	
Product Name	Total	Profit
1957 Vespa GS150	$2,238.30	$755.55
1982 Ducati 900 Monster	$1,948.22	$582.32
Sum Motorcycles	*$4,186.52*	*$1,337.87*
Product Line	Trucks and Buses	
Product Name	Total	Profit
1940 Ford Pickup Truck	$4,973.50	$2,115.33
1958 Setra Bus	$2,554.44	$918.54
1996 Peterbilt 379 Stake Bed	$1,352.86	$579.83
Sum Trucks and Bus	*$8,880.80*	*$3,613.70*
Product Line	Vintage Cars	
Product Name	Total	Profit
1939 Cadillac Limousine	$2,163.50	$1,006.50
Sum Vintage Cars	*$2,163.50*	*$1,006.50*
Sum San Francisco	*$24,931.13*	*$10,439.26*
Sum	*$24,931.13*	*$10,439.26*

Figure 8-27 A table organizing detailed sales data in sections

Suppose you create a bar chart based on this table data. You want the chart to show the overall sales total and profit for each product line, just like the chart in Figure 8-26. You select the product line column as the category series and the total and profit columns as the value series, as shown in Figure 8-28.

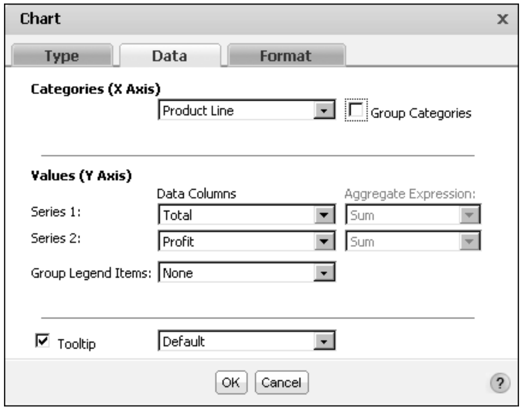


Figure 8-28 Chart—Data showing columns selected for the category and values series

Based on these selections, the bar chart looks like the one in Figure 8-29. Because you did not group or aggregate the data for the chart, the chart plots every total value and every profit value.

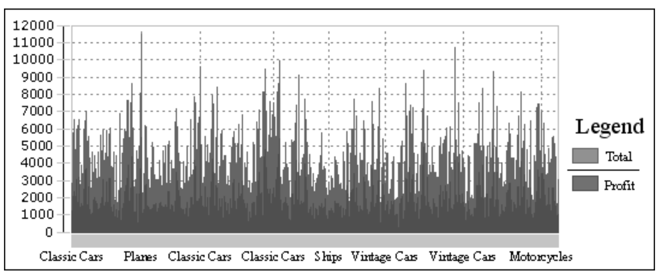


Figure 8-29 A chart that does not group or aggregate data displaying too many data points

To create the chart in Figure 8-26, which is based on the table data in Figure 8-27, group the chart data by product line and use the Sum function to calculate the grand total of the sales amounts and profits. Figure 8-30 shows the correct way to select and organize data for the chart.

The following sections expand on the concepts in this section and provide examples for selecting and organizing data, for different types of table data and for the different types of charts.

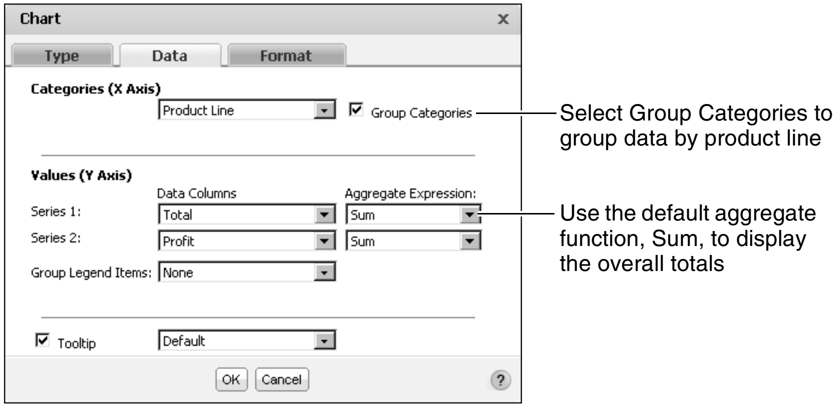


Figure 8-30 Grouping and aggregating data for a chart

Selecting data for an area, bar, or line chart

Area charts, bar charts, and line charts can often be used interchangeably to display the same types of data. This section shows examples of selecting and organizing data for these chart types. Each example shows the following items:

- The area, bar, and line charts
- The table data on which the charts are based
- The selections on Chart—Data

Example 1

The charts in Figure 8-31 show sales numbers by month.

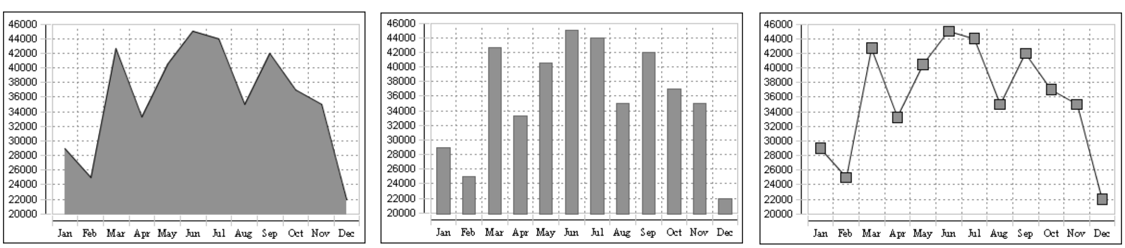


Figure 8-31 An area chart, a bar chart, and a line chart displaying sales numbers by month

Figure 8-32 shows the table data that the charts use. The data in the table is already aggregated. The table shows the total sales per month. Each value is plotted in each of the charts.

Month	Sales
Jan	29000
Feb	25000
Mar	42700
Apr	33250
May	40500
Jun	45000
Jul	44000
Aug	35000
Sep	42000
Oct	37000
Nov	35000
Dec	22000

Figure 8-32 The table data used by the charts in Figure 8-31

Figure 8-33 shows how the data is selected for the chart. You do not need to group or aggregate the data. As Figure 8-33 shows, all you do is select the Month column as the category series, and the Sales column as the value series.

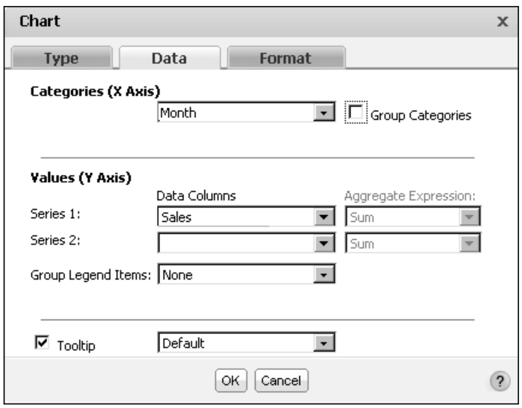


Figure 8-33 The data selected for the charts in Figure 8-31

Example 2

The charts in Figure 8-34 show sales totals by quarter and by region.

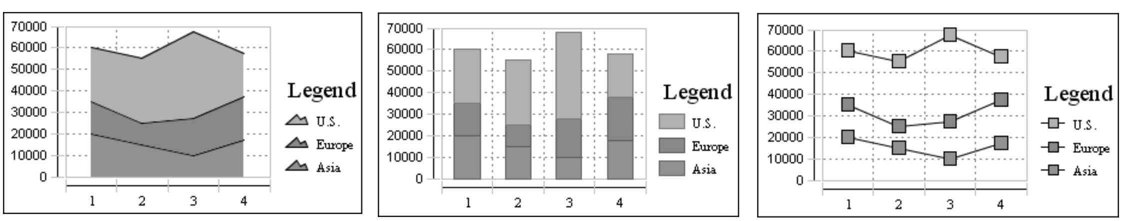


Figure 8-34 An area chart, a bar chart, and a line chart displaying sales by quarter and region

Figure 8-35 shows the table data that the charts use.

Region	Quarter	Sales
Asia	1	20000
Asia	2	15000
Asia	3	10000
Asia	4	17500
Europe	1	15000
Europe	2	10000
Europe	3	17500
Europe	4	20000
U.S.	1	25000
U.S.	2	30000
U.S.	3	40000
U.S.	4	20000

Figure 8-35 The table data used by the charts in Figure 8-34

Figure 8-36 shows how the data is selected for the chart. The Quarter column is the category series, and the Sales column is the value series. In addition, Region is the column selected for the Group Legend Items option. Without this option selected, the chart plots every value in the Quarter and Sales columns, as shown in Figure 8-37.

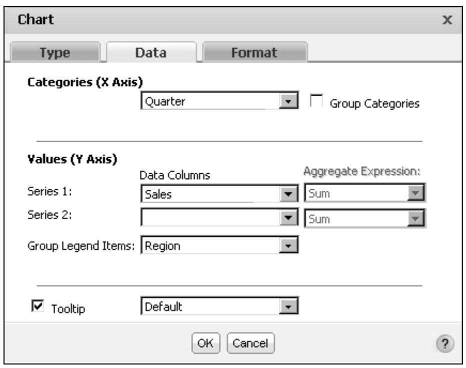


Figure 8-36 The data selected for the charts in Figure 8-34

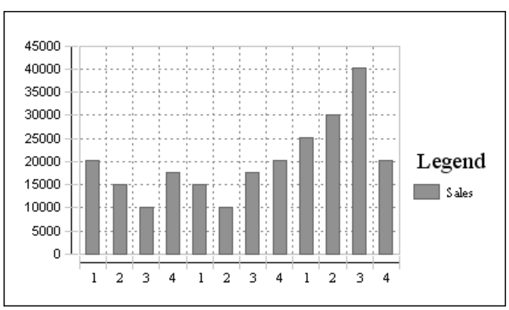


Figure 8-37 The resulting bar chart when data is not grouped by region

Selecting data for a pie chart

Unlike other chart types, a pie chart can display only one value series. For example, a pie chart can show sales totals by product line, but it cannot show sales totals and profit totals by product line. This section shows examples of selecting and organizing data for pie charts. Each example shows the following items:

- The pie chart
- The table data on which the chart is based
- The selections made in Chart—Data

Example 1

The pie chart in Figure 8-38 shows sales by quarter.

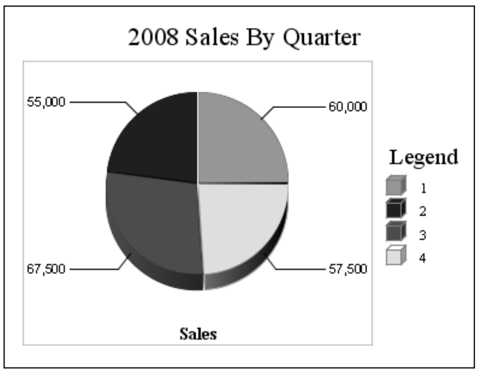


Figure 8-38 A pie chart showing sales by quarter

Figure 8-39 shows the table data that the pie chart uses.

Region	Quarter	Sales
Asia	1	20000
Asia	2	15000
Asia	3	10000
Asia	4	17500
Europe	1	15000
Europe	2	10000
Europe	3	17500
Europe	4	20000
U.S.	1	25000
U.S.	2	30000
U.S.	3	40000
U.S.	4	20000

Figure 8-39 The table data used by the chart in Figure 8-38

Figure 8-40 shows how the data is selected for the chart. The Quarter column is the category series, and the Sales column is the value series. In addition, the data is grouped by quarter, and the Sum function calculates the grand total of the sales, across regions, for each quarter.

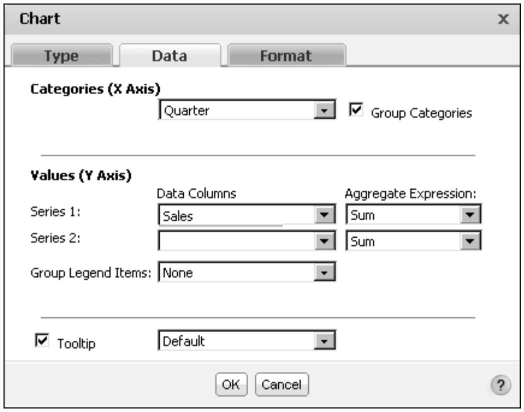


Figure 8-40 The data selected for the chart in Figure 8-38

Example 2

Figure 8-41 shows a series of pie charts. Each pie chart shows the sales by quarter for a particular region.

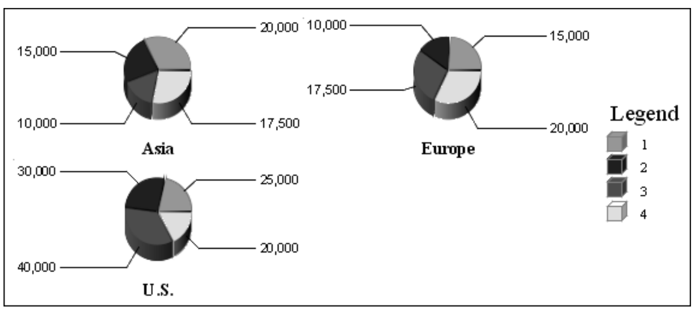


Figure 8-41 A series of pie charts showing sales by quarter and by regions

Figure 8-42 shows the table data that the series of pie charts use. It is the same set of data used in the previous pie chart example.

Region	Quarter	Sales
Asia	1	20000
Asia	2	15000
Asia	3	10000
Asia	4	17500
Europe	1	15000
Europe	2	10000
Europe	3	17500
Europe	4	20000
U.S.	1	25000
U.S.	2	30000
U.S.	3	40000
U.S.	4	20000

Figure 8-42 The table data used by the charts in Figure 8-41

Figure 8-43 shows how the data is selected for the chart. The Quarter column is the category series and the Sales column is the value series. Unlike the previous pie chart example, the category series (quarter) values are not grouped. Instead, the value series data is grouped by the region column.

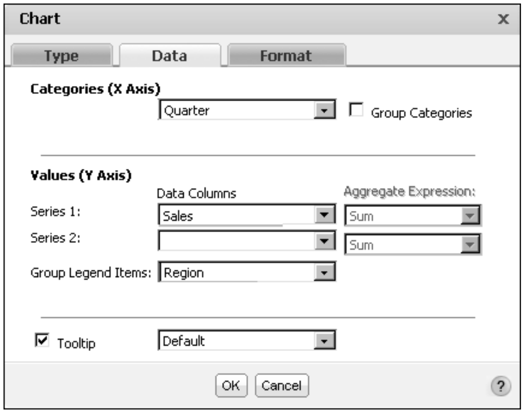


Figure 8-43 The data selected for the charts in Figure 8-41

Example 3

The pie chart in Figure 8-44 shows the number of products in each product line.

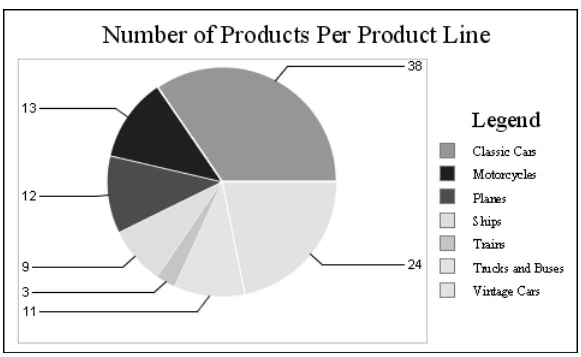


Figure 8-44 A pie chart showing the number of products in each product line

Figure 8-45 shows a sample of the table data that the pie chart uses. The actual data spans two pages.

Product Line	Product	Stock	Buy Price	Inventory Value
Classic Cars				
	1952 Alpine Renault 1300	7305	$98.58	$720,126.90
	1972 Alfa Romeo GTA	3252	$85.68	$278,631.36
	1962 LanciaA Delta 16V	6791	$103.42	$702,325.22
	1968 Ford Mustang	68	$95.34	$6,483.12
	2001 Ferrari Enzo	3619	$95.59	$345,940.21
	1969 Corvair Monza	6906	$89.14	$615,600.84
	1968 Dodge Charger	9123	$75.16	$685,684.68
	1969 Ford Falcon	1049	$83.05	$87,119.45
	1970 Plymouth Hemi Cuda	5663	$31.92	$180,762.96
	1969 Dodge Charger	7323	$58.73	$430,079.79
	1993 Mazda RX-7	3975	$83.51	$331,952.25
Motorcycles				
	1969 Harley Davidson Ultimate Chopper	7933	$48.81	$387,209.73
	1996 Moto Guzzi 1100i	6625	$68.99	$457,058.75
	2003 Harley-Davidson Eagle Drag Bike	5582	$91.02	$508,073.64
	2002 Suzuki XREO	9997	$66.27	$662,501.19
Trucks and Buses				
	1958 Setra Bus	1579	$77.90	$123,004.10
	1957 Chevy Pickup	6125	$55.70	$341,162.50
	1940 Ford Pickup Truck	2613	$58.33	$152,416.29
Vintage Cars				
	1937 Lincoln Berline	8693	$60.62	$526,969.66
	1936 Mercedes-Benz 500K Special Roadster	8635	$24.26	$209,485.10

Figure 8-45 The table data used by the chart in Figure 8-44

Figure 8-46 shows how the data is selected for the chart. The PRODUCTLINE column is the category series, and the PRODUCTNAME column is the value series. In addition, the data is grouped by product line, and the Count function counts the number of products in each product line.

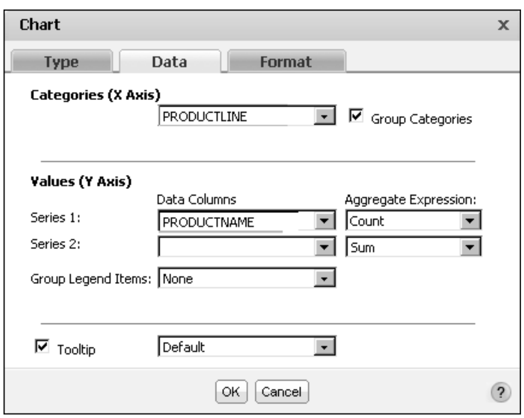


Figure 8-46 The data selected for the chart in Figure 8-44

Selecting data for a meter chart

Unlike most chart types, a meter chart does not display data on a *y*- or *x*-axis. Instead, a meter chart displays each data value as a needle on a dial. This section shows examples of selecting and organizing data for meter charts. Each example shows the following items:

- The meter chart
- The table data on which the chart is based
- The selections made in Chart—Data

Example 1

The superimposed meter chart in Figure 8-47 shows two values, actual sales total and projected sales total.

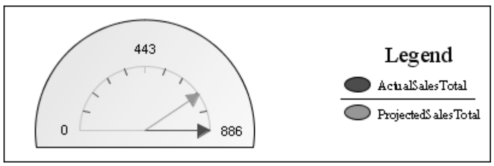


Figure 8-47 A meter chart showing actual sales and projected sales

Figure 8-48 shows the table data that the chart uses. The meter chart displays the values in the first and second columns of the table.

Actual Sales Total	Projected Sales Total	Category	Year	Projected Sales	Actual Sales
886					
	720				
		CPU			
			2005	80	78
			2006	134	120
		Sum CPU		***214***	***198***
		Keyboards			
			2005	68	164
			2006	70	80
		Sum Keyboards		***138***	***244***
		Monitors			
			2005	89	130
			2006	39	60
		Sum Monitors		***128***	***190***
		Printers			
			2005	130	164
			2006	110	90
		Sum Printers		***240***	***254***

Figure 8-48 The table data used by the chart in Figure 8-47

Figure 8-49 shows how the data is selected for the chart. A category series is not applicable to this meter chart, but a value is required, so the Year column is selected arbitrarily. The ActualSalesTotal column and the ProjectedSalesTotal column are the value series.

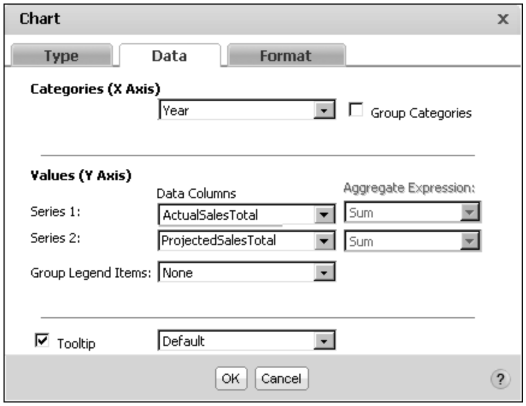


Figure 8-49 The data selected for the chart in Figure 8-47

Example 2

The standard meter chart in Figure 8-50 shows multiple dials. Each dial in the top row shows the projected sales total for an item category. Each dial in the bottom row shows the actual sales total for an item category.

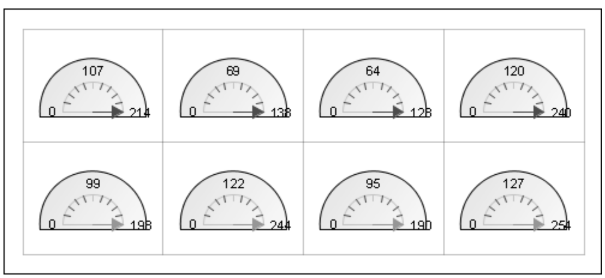


Figure 8-50 Sales by item category displayed in multiple dials in a meter chart

Figure 8-51 shows the table data that the chart uses. It is the same table the previous meter chart example uses. In this example, however, the meter chart uses data from the Projected Sales and Actual Sales columns. Notice that the aggregate (Sum) row for each item corresponds to a needle value in each dial.

Figure 8-52 shows how the data is selected for the chart. The Category column is the category series, and the Group Categories option is selected. The ProjectedSales column and the ActualSales column are the value series. The values in these columns are aggregated using the Sum function, and the values are grouped by category.

Actual Sales Total	Projected Sales Total	Category	Year	Projected Sales	Actual Sales
886					
	720				
		CPU			
			2005	80	78
			2006	134	120
		Sum CPU		***214***	***198***
		Keyboards			
			2005	68	164
			2006	70	80
		Sum Keyboards		***138***	***244***
		Monitors			
			2005	89	130
			2006	39	60
		Sum Monitors		***128***	***190***
		Printers			
			2005	130	164
			2006	110	90
		Sum Printers		***240***	***254***

Figure 8-51 The table data used by the chart in Figure 8-50

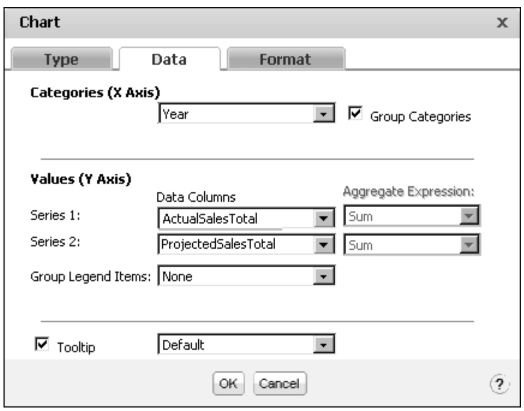


Figure 8-52 The data selected for the chart in Figure 8-50

Selecting data for a scatter chart

A scatter chart displays data as *x-y* coordinates. It combines each pair of numeric values into single data points. You select columns that contain numeric values for both the category and the value series. This section shows examples of selecting and organizing data for scatter charts. Each example shows the following items:

- The scatter chart
- The table data on which the chart is based
- The selections made in Chart—Data

Example 1

The scatter chart in Figure 8-53 shows the relationship between salaries and years of experience.

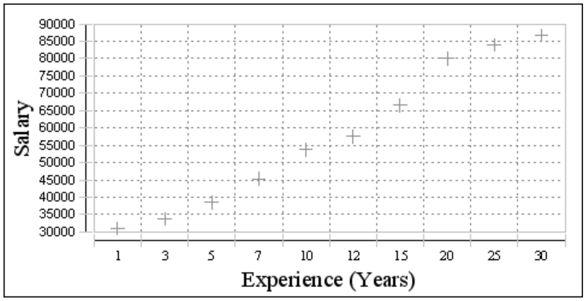


Figure 8-53 A scatter chart showing the relationship between salaries and years of experience

Figure 8-54 shows the table that contains the data which the scatter chart uses. The chart uses data from the Years_Experience and Average Salary columns.

Years_Experience	Average Salary	Average Salary (Male)	Average Salary (Female)
1.0	31000	32000.0	30000
3.0	33750	35000.0	32500
5.0	38500	40000.0	37000
7.0	45250	47500.0	43000
10.0	53750	55000.0	52500
12.0	57500	60000.0	55000
15.0	66500	70000.0	63000
20.0	80000	85000.0	75000
25.0	84000	90000.0	78000
30.0	86750	93500.0	80000

Figure 8-54 The table data used by the chart in Figure 8-53

Figure 8-55 shows how the data is selected for the chart.

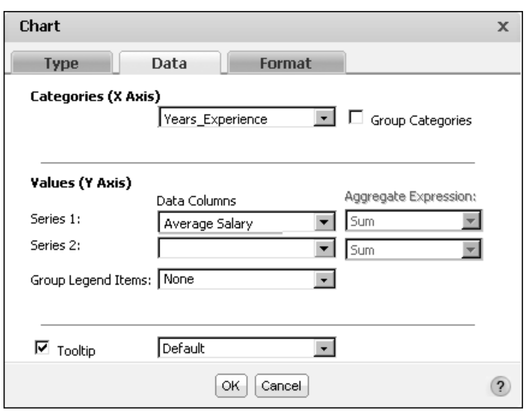


Figure 8-55 The data selected for the chart in Figure 8-53

The Years_Experience column is the category series, and the Average Salary column is the value series. The data is not grouped nor aggregated, because the scatter chart plots every value in the Years_Experience and Average Salary columns.

Example 2

Like the scatter chart in the previous example, the scatter chart in Figure 8-56 shows the relationship between salaries and years of experience. In this example, however, the chart displays two value series: the average salaries for men and the average salaries for women.

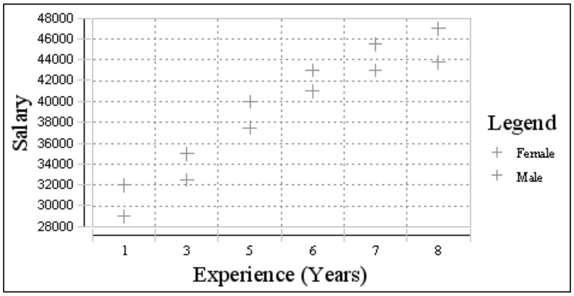


Figure 8-56 A scatter chart showing the relationship between salaries and years of experience, by gender

Figure 8-57 shows the table containing the data that the scatter chart uses. Unlike the table in the previous scatter chart example, this table does not show the average salaries by years of experience or by gender. Instead, this table shows salary and experience data for each employee. To display the average salaries by years of experience and by gender, the data must be grouped and aggregated.

Employee	Gender	Years_Experience	Salary
Alex Chang	Male	5.0	39000
Barbara Billings	Female	8.0	43500
Bob Matthews	Male	7.0	46000
David Taylor	Male	7.0	45000
George Smith	Male	3.0	35500
James Lee	Male	6.0	43000
Joe Bennett	Male	5.0	41000
Joe Manetta	Male	3.0	34500
John Smithers	Male	8.0	47000
Jose Vargas	Male	3.0	35000
Kim Smith	Female	5.0	38000
Maria Jose	Female	1.0	28000
Mary Anderson	Female	8.0	44000
Maya Salvatore	Female	3.0	32000
Richard Wade	Male	1.0	31000
Robert Miller	Male	1.0	33000
Sally Firth	Female	1.0	30000
Sandra Au	Female	5.0	37000
Sarah Jones	Female	6.0	41000
So-Young Kim	Female	3.0	33000
Susan Volt	Female	7.0	43000
Tyler Oliver	Male	5.0	40000

Figure 8-57 The table data used by the chart in Figure 8-56

Figure 8-58 shows how the data is selected, grouped, and aggregated for the chart. The Years_Experience column is the category series and the Salary column is the value series. The values are grouped by years of experience and gender. The Average aggregate function is selected, so that the chart calculates and displays the average salary for each group.

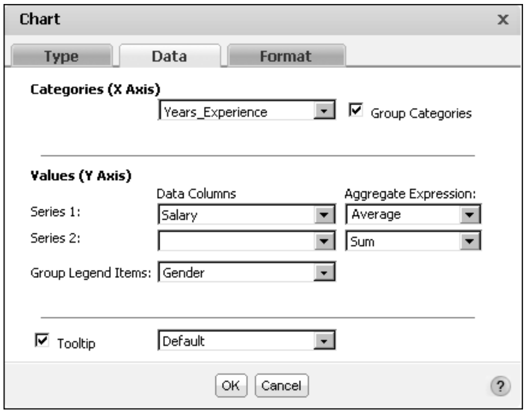


Figure 8-58 The data selected for the chart in Figure 8-56

Selecting data for a stock chart

A stock chart requires the following data:

- Date values as the categories series
- High values, low values, open values, and close values as the value series

Example

The stock chart in Figure 8-59 shows a stock's high, low, open, and close values for five days.

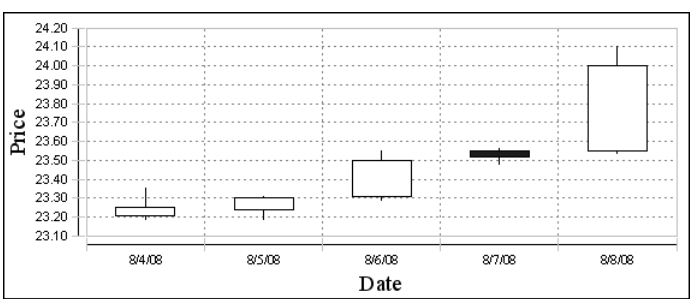


Figure 8-59 A stock chart showing high, low, open, and close values

Figure 8-60 shows the table data that the stock chart uses.

Company	Date	Open	Close	Low	High
MYCO					
	8/4/08	23.21	23.25	23.19	23.35
	8/5/08	23.24	23.3	23.19	23.31
	8/6/08	23.31	23.5	23.29	23.55
	8/7/08	23.55	23.52	23.48	23.56
	8/8/08	23.55	24.0	23.54	24.1

Figure 8-60 The table data used by the chart in Figure 8-59

Figure 8-61 shows how the data is selected for the chart. The Date column is the category series. The High, Low, Open, and Close columns are the value series.

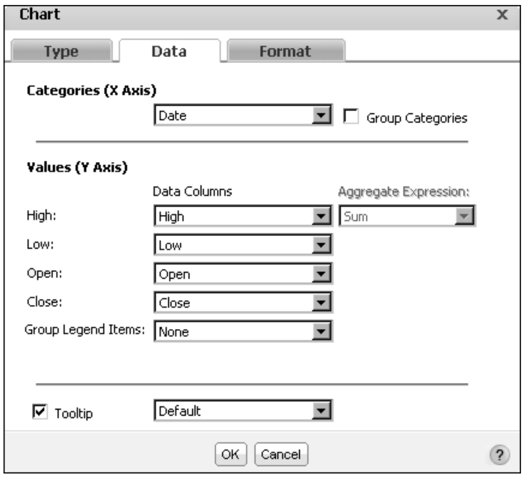


Figure 8-61 The data selected for the chart in Figure 8-59

Displaying additional series information

By now, you may have noticed the Tooltip option on the Chart—Data page of the chart editor. The Tooltip option is selected by default, as shown in Figure 8-61. A Tooltip displays the value series value when a user hovers the mouse pointer over a bar in a bar chart, a sector in a pie chart, or a data-point marker in a line chart. Figure 8-62 shows a Tooltip displaying 40,000 when the user hovers the mouse pointer over a bar in a bar chart.

You can disable the Tooltip by deselecting the Tooltip option. You also can display a different Tooltip value. For the bar chart in Figure 8-62, instead of displaying the sales value, you can display the region or the quarter. To display a different Tooltip value, select a different column from the drop-down list, as shown in Figure 8-63.

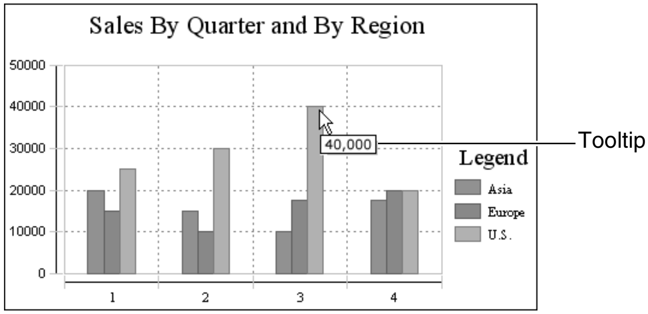


Figure 8-62 A Tooltip displaying a value series value

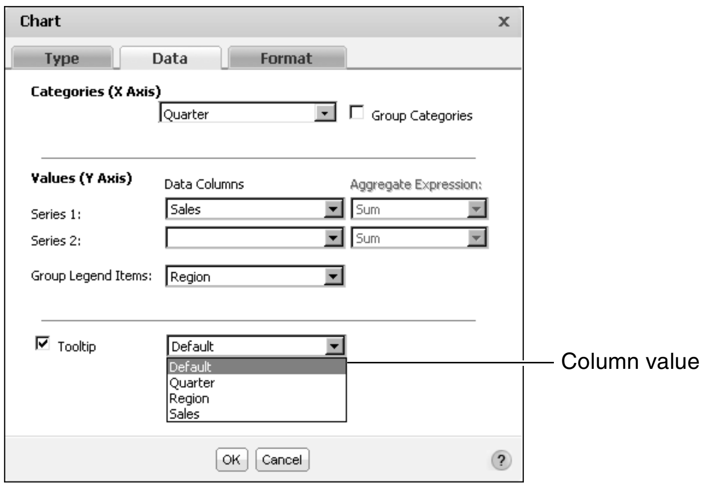


Figure 8-63 Selecting a value to display as a Tooltip

Formatting a chart

When you create a chart and use the default formatting options, the chart type and the data you select determines the basic look of the chart. The template or the software determines styles, such as fonts and colors. Figure 8-64 shows an example of a bar chart that uses default formats. You can improve the chart by editing the placeholder text for the chart title, *y*-axis title, and *x*-axis title, and by changing the size or position of the *x*-axis labels to display all the city labels.

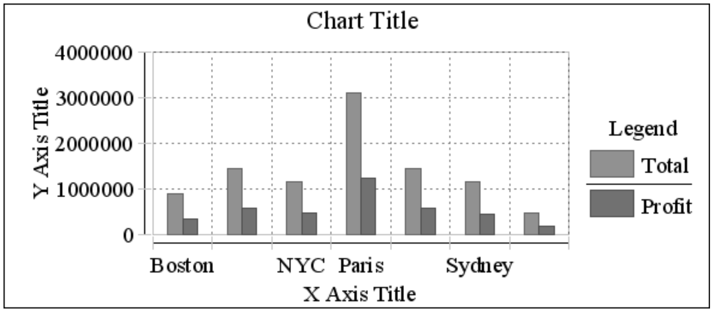


Figure 8-64 A bar chart using default formats

To change the appearance of a chart, change the option settings on Chart—Format. Figure 8-65 shows Chart—Format displaying the default option settings for a bar chart. The options that appear differ depending on the chart type. For example, a pie chart does not have axes, so the Category (X) Axis Title and Value (Y) Axis Title options do not appear on Chart—Format for a pie chart.

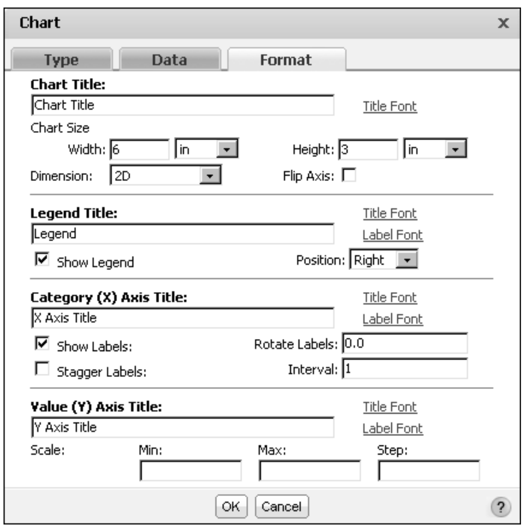


Figure 8-65 Chart—Format displaying the default options for a bar chart

Changing the size of a chart

You can change a chart's width and height. If a chart looks too big, you can reduce its width, or height, or both. For example, a superimposed meter chart that displays data on one dial can look oversized. Conversely, increase the size of a chart if items in the chart look too crowded. For example, if a bar chart displays many bars, and many of the *x*-axis labels do not appear, you can start by increasing the width of the chart to see if all the items fit.

Setting a chart's width

To set the width of a chart, on Chart—Format, in Chart Size, type a number in Width. A chart's width is limited by the width of the table if you select Fixed Width for the Layout Preference option in Page Setup. If you specify a chart width that exceeds the width of the table, the chart appears truncated. Figure 8-66 shows a chart where the legend appears truncated. If Layout Preference is set to Auto Expand Width, the table expands to accommodate the chart width you set. In either case, if you increase a chart's width, confirm that the chart still fits the page. Always check the output in the viewer and in PDF format.

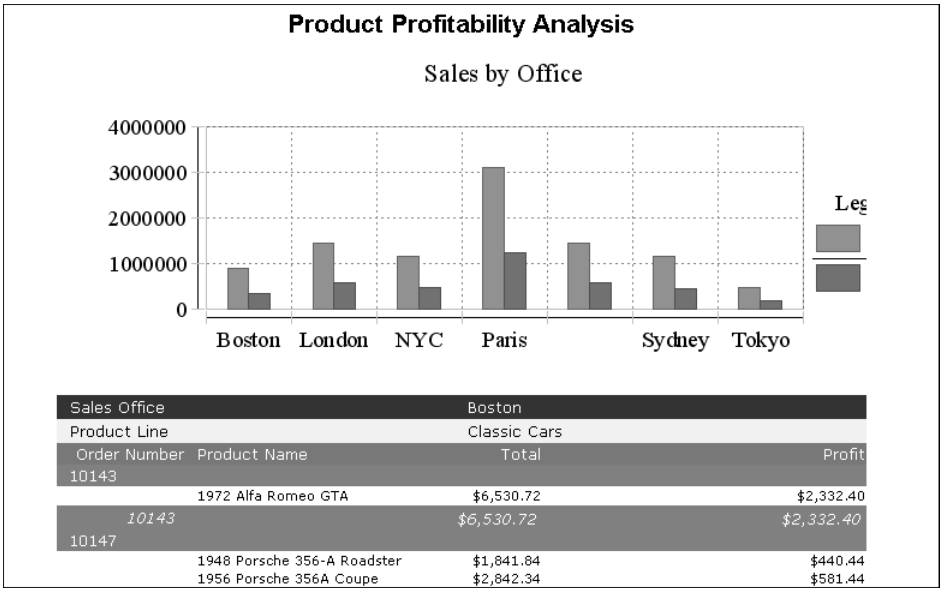


Figure 8-66 A report where the chart width exceeds the table width

For more information about the Layout Preference option, see "Changing the layout type, page size, and orientation" in Chapter 2, "Editing and formatting report content."

Setting a chart's height

To set the height of a chart, in Chart—Format, in Chart Size, type a number in Height. When you increase or decrease a chart's height, you increase or decrease the visual contrast between data values. Compare the charts in Figure 8-67 and Figure 8-68. Both present the same data. By changing the chart height in Figure 8-67, each chart presents a different impression of the data. The taller chart shows more contrast between the data values. If you look at the image alone you think there is a greater disparity between salaries earned by men and women than in the chart in Figure 8-67.

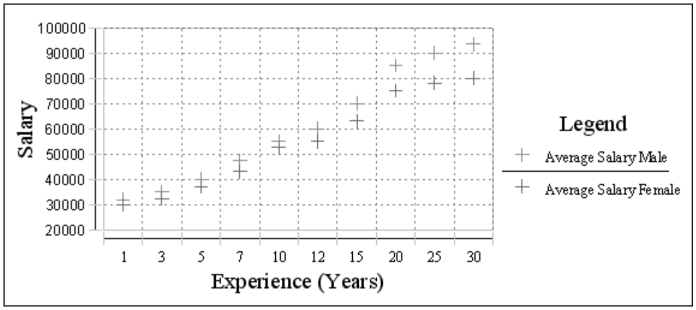


Figure 8-67 A scatter chart using the default chart size

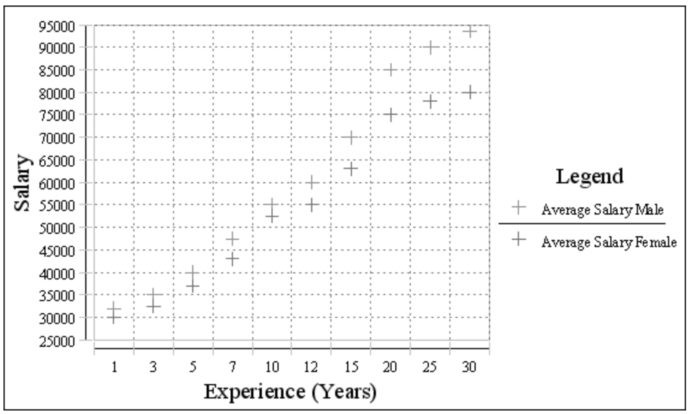


Figure 8-68 A scatter chart with its height increased

Editing and formatting titles and labels

By default, every chart displays Chart Title as its title. If a chart has axes, the chart also displays X Axis Title and Y Axis Title. To edit these titles, in Chart—Format, type new titles in the text boxes in Chart Title, Category (X) Axis Title, and Value (Y) Axis Title, respectively. If you do not want to display any titles, delete the text in the text boxes. You also can change the font attributes of every title and label in the chart. To format a title, choose Title Font. To format a label, choose Label Font. Figure 8-69 shows the font attributes you can change.

You change the font attributes of a title or labels for aesthetic reasons or for practical reasons. For example, if the *x*-axis does not display all the labels, as shown in Figure 8-66, reducing the font size of the Category (X) Axis labels is one way to solve the problem. Figure 8-70 shows the effect of reducing the size of the labels on the *x*-axis. All the city names appear.

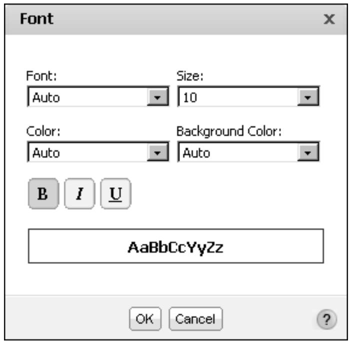


Figure 8-69 Changing attributes on Font

The chart in Figure 8-70 also displays the *y*-axis and legend labels in a smaller size than the same labels in the chart in Figure 8-66. Typically, when you change the font attributes of labels in one area, you apply the same font attributes to labels in other areas, so that the labels in the chart have a consistent appearance.

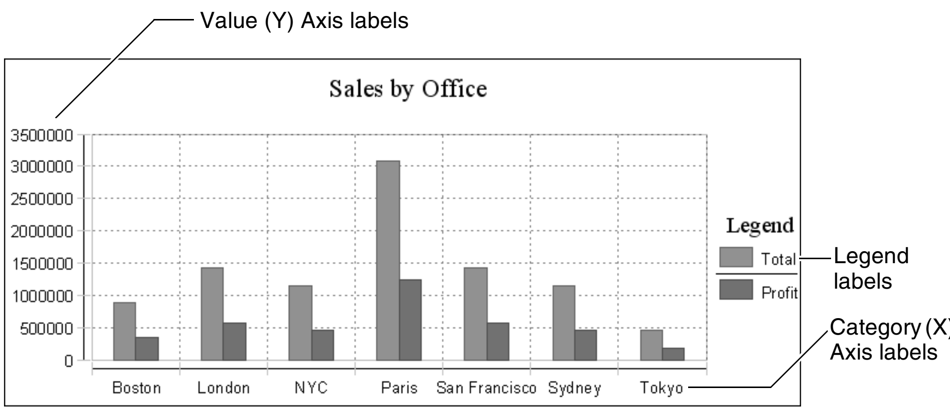


Figure 8-70 Labels in a bar chart set to the same font attributes

Making a chart look three-dimensional

A chart that uses the default formats appears as a two-dimensional chart. You can make a chart appear three-dimensional by selecting 2D With Depth in Dimension, as shown in Figure 8-71. For bar, line and area charts, you can also create three-dimensional charts, by selecting the 3D setting. If you choose the 3D option for a chart, you cannot flip the axes of the chart.

Figure 8-72 shows a bar chart that uses the 2D With Depth setting. Not all chart types support the 2D With Depth setting. For example, this setting is not available for meter charts or line charts.

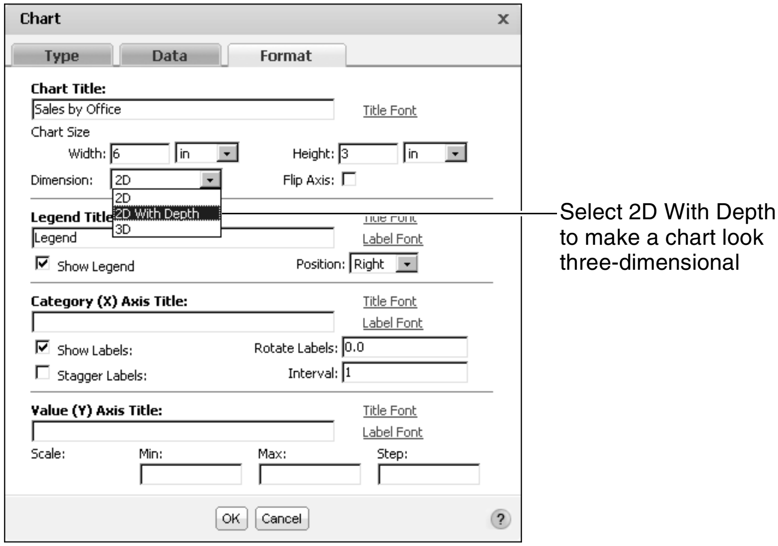


Figure 8-71 Setting the Dimension option

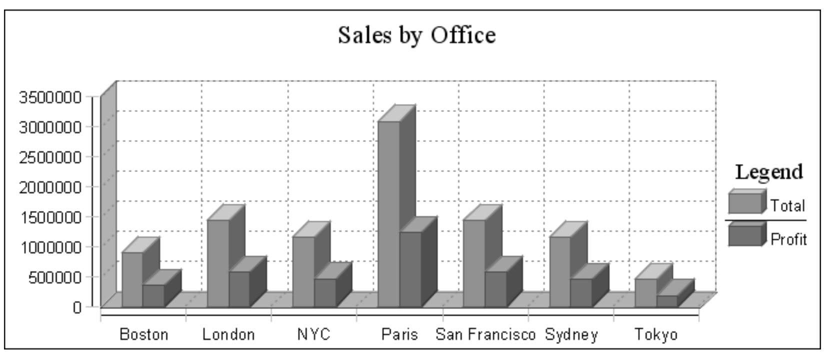


Figure 8-72 A bar chart using the 2D With Depth setting

Changing the position of the legend

By default, a legend appears on the right side of a chart. You can change the position of the legend to appear above, below, or on the left side of the chart. If a chart is too wide, positioning the legend above or below the chart saves space horizontally. To reposition a legend, select Above, Below, Left, Right, or Inside as shown in Figure 8-73.

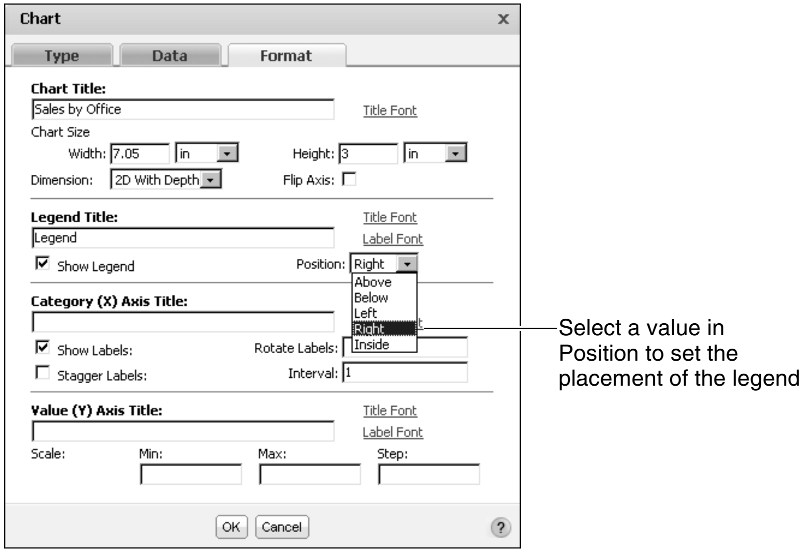


Figure 8-73 Setting the position of a legend

You can also choose to remove a legend by deselecting Show Legend. For example, it is not necessary to display a legend if the chart shows only one value series. The bar chart in Figure 8-74, for example, does not need a legend.

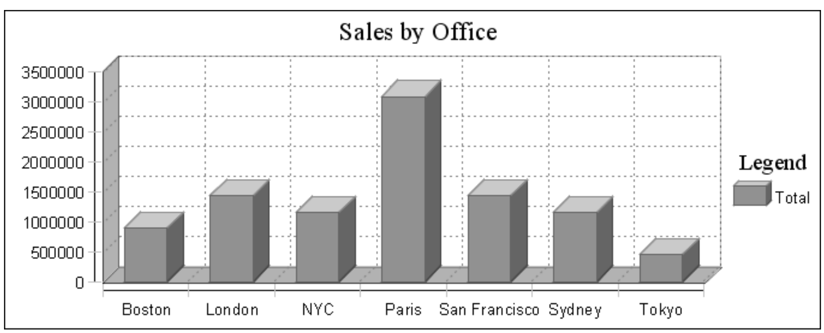


Figure 8-74 A bar chart showing an unnecessary legend

Formatting labels to fit on the *x*-axis

A chart that displays numerous category values sometimes cannot fit all the values on the *x*-axis. As discussed previously, one way to format *x*-axis labels so that they all appear is to reduce the font size. Other techniques include rotating the labels, staggering the labels, and displaying the labels at set intervals. You can also use a combination of these techniques.

Rotating *x*-axis labels

You can rotate labels by a specified number of degrees. On Chart—Format, in Rotate Labels, type the number of degrees to indicate the amount of rotation. To rotate labels in a clockwise direction, use a negative number. To rotate labels in a counter-clockwise direction, use a positive number. Figure 8-75 shows four charts, each with Rotate Labels set to a different value.

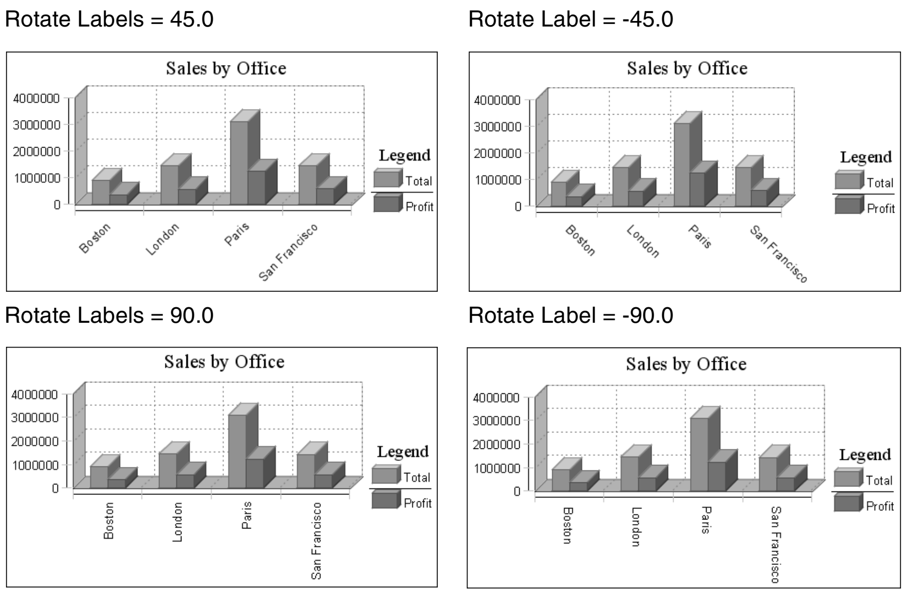


Figure 8-75 Charts using different values for the Rotate Labels option

Staggering *x*-axis labels

On Chart—Format, you can select the Stagger Labels option to place the labels in a zigzag arrangement, as shown in Figure 8-76. The first chart shows the results of staggering, but not rotating, the *x*-axis labels. The second chart shows the results of staggering and rotating the labels 30 degrees.

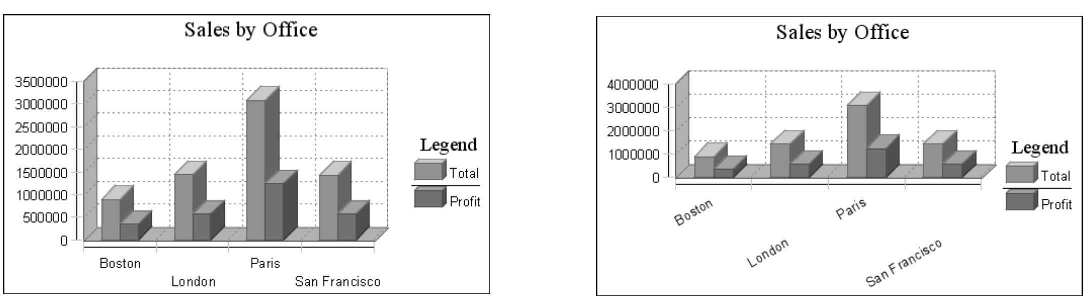


Figure 8-76 Charts using different values for the Stagger Labels and Rotate Labels options

Displaying *x*-axis labels by interval

By default, a chart displays every category value on the *x*-axis, unless the label does not fit in the space allocated to each category value. As described in the previous sections, to make all the labels fit, you can reduce the font size, rotate the labels, or stagger the labels. These techniques, however, can make the *x*-axis look crowded. If it is not essential to display every category value, you can specify that the *x*-axis display alternate values, or every third value, and so on.

On Chart—Format, the Interval option is set to 1 by default, which means every value appears. Set Interval to a different number to specify the interval at which values appear. For example, to display alternate values, set Interval to 2. Figure 8-77 shows a stock chart that displays open, close, high, and low prices for every trading day in August. Although Interval is set to 1, there is not enough space to display every date as a label on the *x*-axis, so the *x*-axis displays alternate values. Figure 8-78 shows the same stock chart, but this time, Interval is set to 5 to display the first trading date per week.

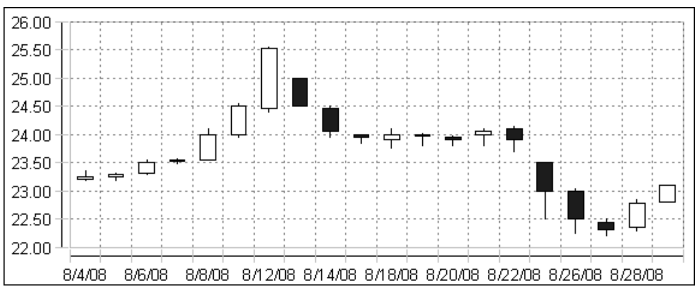


Figure 8-77 A stock chart displaying every trading date value

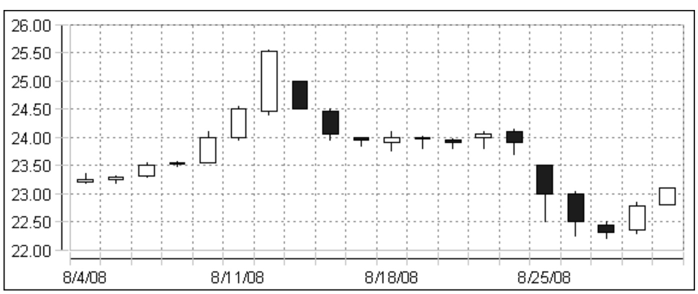


Figure 8-78 A stock chart displaying the first trading date per week

As the chart in Figure 8-78 shows, it makes sense to skip category values only if the values are consecutive numbers or dates, where the user can intuitively fill in the missing values.

Changing the range of values on the *y*-axis

The number and range of values on the *y*-axis depends on several factors, including the smallest value series value, the largest value series value, the height of the chart, and the amount of vertical space the chart items require.

Compare the charts in Figure 8-79. Both charts present the same data, but the *y*-axis values are different. In the chart on the left, the *y*-axis displays eight values, ranging from 0 to 3500000. In the chart on the right, the *y*-axis displays five values, ranging from 0 to 4000000. Both charts have the same height value, so in these examples, the different *y*-axis values are a result of the different amounts of space the *x*-axis labels require.

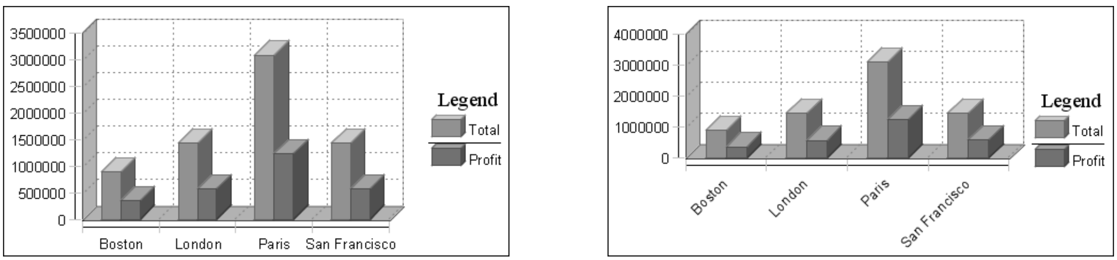


Figure 8-79 Charts displaying different ranges of values on the *y*-axis

You can control the range of values that appear on the *y*-axis. On Chart—Format, in Scale, you can set the following options:

- Min. Type a number that represents the lowest value to display on the *y*-axis. This number appears at the bottom of the *y*-axis.
- Max. Type a number that represents the highest value to display on the *y*-axis. This number appears at the top of the *y*-axis.
- Step. Type a number that represents the increment between each value.

Figure 8-80 shows the Min, Max, and Step options set to 0, 3500000, and 500000, respectively. These settings change the *y*-axis values of the second chart in Figure 8-79 to the *y*-axis values in Figure 8-81.

Notice that the *y*-axis values in Figure 8-81 are closer together. Because the chart height is the same, the chart has to display more values in the same space. When you change the scale of *y*-axis values, you typically adjust the height of the chart.

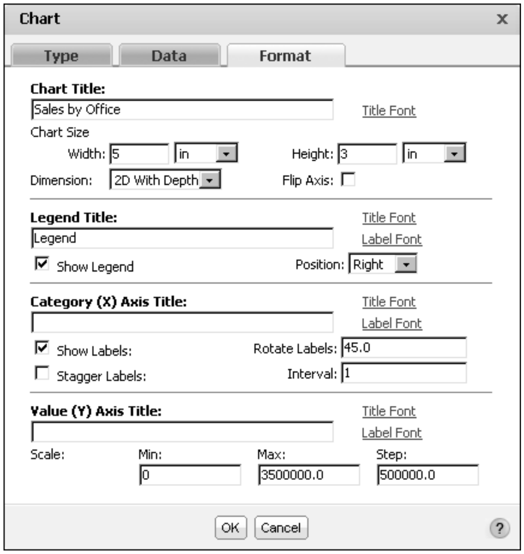


Figure 8-80 Chart—Format showing modified scale values

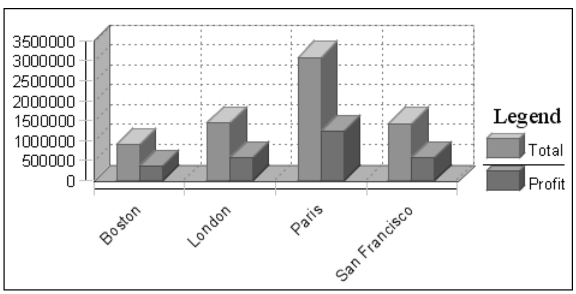


Figure 8-81 The chart using the modified scale values

Working with Flash charts

BIRT Studio enables you to create standard charts and Flash charts. Flash charts are similar to standard charts, and are designed to be used in a similar manner. Standard charts display as static images. Flash charts add animation and other visual elements that enable report developers and users to display data in a format that is easily understood. Flash charts provide additional Rich Internet Application features in a BIRT report. For example, an animated Flash column chart can progressively draw its columns from bottom to top. The examples shown in Figure 8-82 display a standard doughnut chart, and a segmented Flash doughnut chart, viewed in the Actuate BIRT Viewer. In the animated Flash doughnut chart as you select a segment, it slices away from the rest of the chart.

Figure 8-82 Standard doughnut chart, and segmented Flash doughnut chart

In BIRT Studio you can insert a Flash chart at the table level, or at the section level, if your report organizes data in sections. You can choose from the following types of Flash charts:

- Column
- Bar
- Line
- Pie
- Doughnut

Creating a Flash chart

In BIRT Studio, you can insert a Flash chart at the table level, or section level, just as for regular charts. The procedure to create a Flash chart, is similar to the one to create a standard chart. To create a Flash chart, perform the following steps:

1 On BIRT Studio, select the report table, or report section, and choose Chart➛Insert Flash Chart, as shown in Figure 8-83.

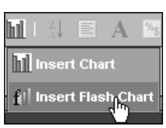


Figure 8-83 Inserting a Flash chart

Flash Chart appears.

2 In Flash Chart—Type, select a chart type. Figure 8-84 shows the types of Flash charts available in BIRT Studio.

3 In Flash Chart—Data, specify the data to present in the chart, in the same way you did for a standard chart.

4 On Flash Chart—Format, specify the formatting properties for the chart. The available formatting options depend on the chart type, and are similar to the formatting options available for standard charts.

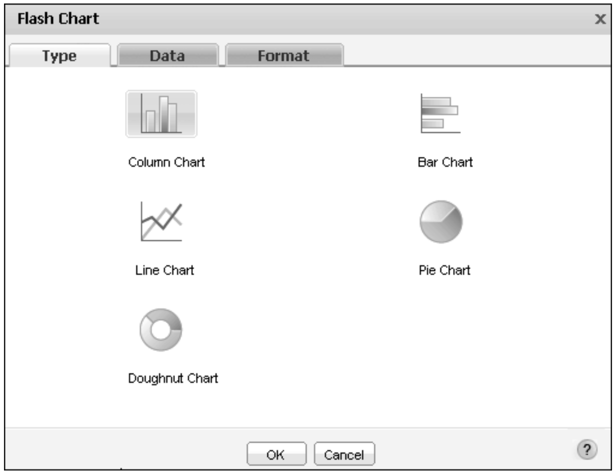


Figure 8-84 Selecting the type of Flash chart

The following sections describe how to select data to display in a Flash chart, and how to format a Flash chart.

Selecting data for a Flash chart

The process to select data for a Flash chart is similar to the process to do so for a standard chart. When you select data for a column, line, or bar Flash chart, you specify the value series and the category series, if applicable. You then set the grouping options and select a value to display for the Tooltip from the drop-down list. When you select data for a pie or doughnut Flash chart, you specify the values to display as sectors of the pie or doughnut. You then specify the grouping options and select a value to display for the Tooltip from the drop-down list.

Flash Chart—Data shown in Figure 8-85 displays an option to Use Default Hyperlinks. This option is highlighted when you work with a chart that retrieves data from a data object containing predefined hyperlinks. When this field appears highlighted, you can specify whether to display or suppress default hyperlinks in the chart.

How to select data for a column, line, or bar Flash chart

In Flash Chart—Data, complete the following steps:

1. In Categories (X Axis) select a column from the drop-down list, to set as the *x*-axis.
2. Select Group Categories, to group the values in this column. When selected, only the groups are displayed on the *x*-axis in the chart.

3. In Values (Y Axis) complete the following steps:
    1. In Series 1, select a column from the first drop-down list. If necessary, specify an aggregate function to aggregate the data in this column.
    2. In Series 2, select a column from the second drop-down list. If necessary, specify an aggregate function to aggregate the data in this column.
    3. In Group Legend Items, select a column from the drop-down list. Alternately, accept the default selection of None.
4. In Tooltip, select Tooltip, then select a column from the drop-down list, whose values are displayed as the Tooltip. Deselect Tooltip if you do not want to display a tooltip for values in the chart.

    When selected, the value series values are displayed when the user hovers a mouse pointer over a column, bar, or line in the chart.

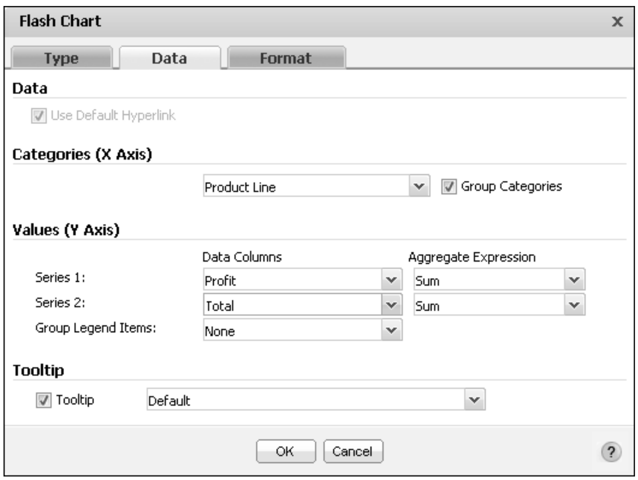


Figure 8-85 Selecting data for a Flash chart

How to select data for a pie Flash chart

In Flash Chart—Data, complete the following steps:

1. In Slice, in Category, select a column from the drop-down list to display as sectors of the pie or doughnut. Select Group Categories to display groups as sectors.
2. In Value, select a column from the drop-down list. Select an aggregate function to aggregate the values in this column.
3. In Tooltip, select Tooltip, then select a column from the drop-down list, whose values are displayed as the Tooltip. Deselect Tooltip if you do not want to display a tooltip for values in the chart.

Formatting a Flash chart

The following section describes the formatting options available for column, line, and bar charts, and pie and doughnut charts.

Formatting a column, line, or bar chart

When you format a column, line, or bar chart, you specify the properties in Flash Chart—Format in a similar way that you do for a standard chart. In Flash Chart—Format you can specify the title, chart size, dimension, and font properties, and legend properties. In addition, you can format the *x*-axis title, and label properties. You can also format the *y*-axis title, minimum and maximum values, as well as step value, as you would for a regular chart. The following section describes how you can work with each option in Flash Chart—Format.

Specifying chart properties

You can specify a chart title, width, height, as well as font properties for Flash charts using BIRT Studio. Figure 8-86 shows an example of a line chart that uses the default font properties. When using a line chart, you cannot specify a chart subtype, or dimension, as you can for bar or column charts.

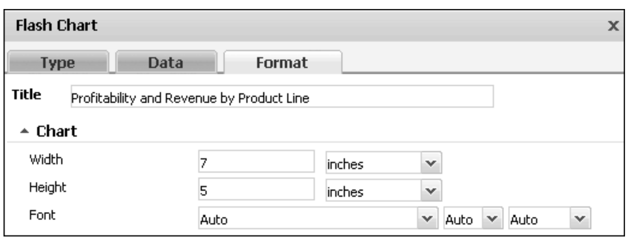


Figure 8-86 Specifying chart properties

Figure 8-87 shows an example of a column chart that uses the default font properties. In addition to specifying the size for bar and column charts, you can select a subtype, dimension, and specify font properties, as you do for standard charts.

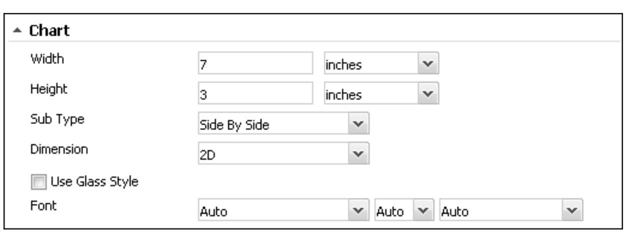


Figure 8-87 Specifying chart properties for a bar or column chart

When working with two-dimensional column or bar charts, you can also specify whether to use glass style to display the chart. Select Use Glass Style to display the bars or columns in the chart with smooth, rounded edges. Deselect Use Glass Style to display angular, sharp edges. Figure 8-88 shows two charts that use the same data. The chart on the left does not use glass style, while the chart on the right uses the glass style option. The Glass Style option is only available for a two-dimensional chart.

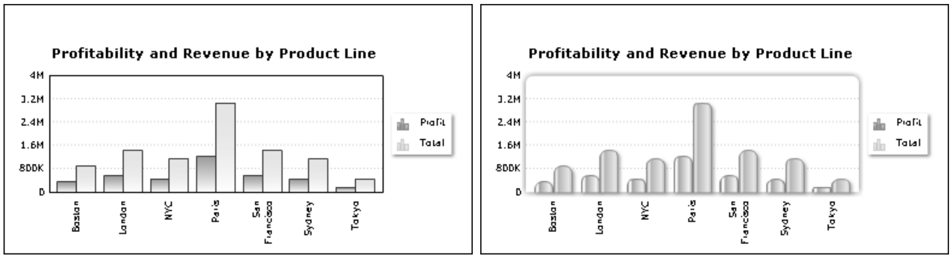


Figure 8-88 Deselecting and selecting glass style in a two-dimensional column chart

Formatting the legend

Using BIRT Studio, you can specify whether to display the legend for a Flash chart. If you choose to display the legend, you can specify a title, as well as position for the legend to the right of the chart, or below the chart, as shown in Figure 8-89.

Figure 8-89 Formatting the legend

Formatting the category (x) axis

You can specify a title for the category axis and specify whether to display labels for *x*-axis values. If you choose to display labels, you can further specify whether to stagger them at an interval, or display them at an angle, as shown in Figure 8-90.

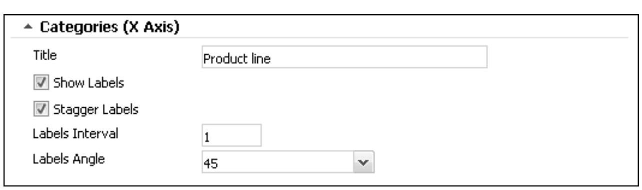


Figure 8-90 Specifying category (x) axis properties

Formatting the value (y) axis

You can specify a title for the value axis, as well as the minimum and maximum value to display. In addition, you can choose Auto Step to enable BIRT Studio to determine the step interval as shown in Figure 8-91, or use the menu to specify a step interval for the displayed values.

Figure 8-91 Specifying value (y) axis properties

Formatting a pie or doughnut chart

When you format a pie or a doughnut chart, you specify a chart title, size, dimension, angle of rotation, radius of the pie or width of the doughnut ring, font, and legend properties.

When you format a pie chart, in addition to specifying the title, size, dimension, and legend, complete the following steps:

1 Specify the rotation angle in degrees. Figure 8-92 displays an angle of rotation set to 180 degrees.

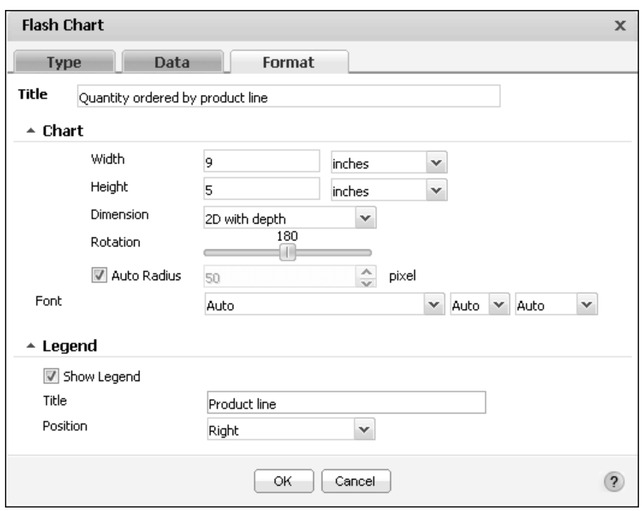


Figure 8-92 Formatting a pie Flash chart

2 Specify the size of the pie:

- Select Auto Radius to enable BIRT Studio to set a radius for the pie, as shown in Figure 8-92.

- Deselect Auto Radius, then specify the size of the pie in pixels.

3 Specify font properties using the drop-down lists to set font type, size, and color.

When you format a doughnut Flash chart, you can specify the width of the doughnut ring as follows:

- Accept the default selection of Auto Outer Radius, and Auto Inner Radius to enable BIRT Studio to determine the width of the doughnut, as shown in Figure 8-93.
- To specify a different value, deselect Auto Outer Radius, and Auto Inner Radius, and use the menus to set the width of the doughnut.

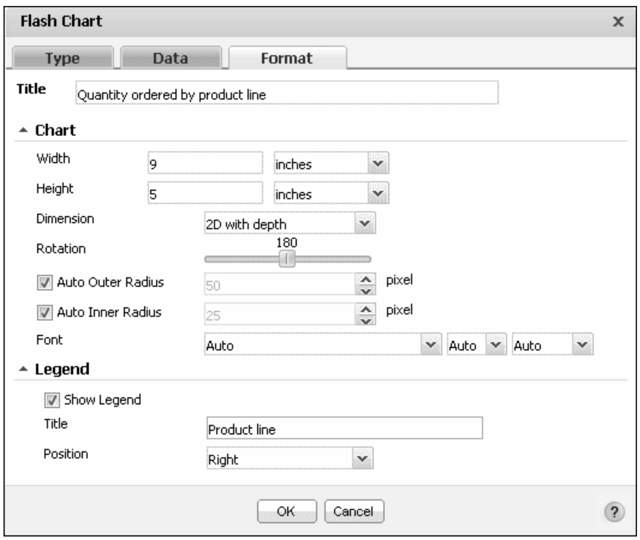


Figure 8-93 Formatting the width of a doughnut Flash chart

Deleting a Flash chart

To delete a Flash chart from a report design, select the chart at the report table or section level, and choose Delete Chart from the context menu, as shown in Figure 8-94.

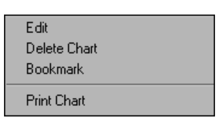


Figure 8-94 Deleting a Flash chart

Limitations

You can export the Flash charts created in BIRT Studio to several formats using the Actuate BIRT Viewers. To preserve the animation properties of a Flash chart, export it to PDF format. When exporting a report containing a Flash chart to other formats, such as Post Script, Excel, Word, or XHTML, you can choose to export the chart as a static image. When the chart is viewed in any of these formats, it does not appear animated.

Displaying a chart without the table data

A report design that contains a lot of information in a series of tables with charts can often be more effective if it displays only the charts. To display only a chart, right-click anywhere in the table, and choose Hide Table. Choose Show All to re-display the table data. You can hide the table for report designs that contain regular charts and Flash charts.

Chapter

9

Functions and operators

This chapter contains the following topics:

- Functions
- Operators

Functions

This section is a complete reference to all of the functions in BIRT Studio. This reference organizes the functions into two sections based on those used in expressions when creating a computed column and functions that are used to perform aggregate calculations.

Functions used in computed column expressions

The following list of functions appear when you create expressions to compute column data. Each function entry includes a general description of the function, its syntax, the arguments to the function, the result the function returns, and an example that shows typical usage. Use this reference to find information about a function that you want to use when you insert a computed column to display calculated data in a report design.

ABS( )

Returns the absolute value of a number without regard to its sign. For example, 6 is the absolute value of 6 and -6.

Syntax ABS(num)

Argument **num**
The number, or numeric expression that specifies the number, for which you want to find the absolute value.

Returns A number that represents the absolute value of num.

Example The following example returns the absolute value for each number in the TemperatureCelsius data field:

```
ABS([TemperatureCelsius])
```

ADD_DAY( )

Adds a specified number of days to a date value.

Syntax ADD_DAY(date, daysToAdd)

Arguments **date**
The date or date expression that represents the start date.

daysToAdd
The number of days to add to the start date. If you specify a negative number, the result appears to subtract the number from the start date.

Returns The date value that results from adding the specified number of days to the start date.

Example The following example adds 15 days to each date value in the InvoiceDate data field:

```
ADD_DAY([InvoiceDate], 15)
```

ADD_HOUR( )

Adds a specified number of hours to a date value.

Syntax ADD_HOUR(date, hoursToAdd)

Arguments

date
The date or date expression that represents the start date. If a start date does not have a time value, the function assumes the time is midnight, 12:00 AM.

hoursToAdd
The number of hours to add to the start date. If you specify a negative number, the result appears to subtract the number from the start date.

Returns The date-and-time value that results from adding the specified number of hours to the start date.

Example The following example adds eight hours to each date value in the ShipDate data field:

```
ADD_HOUR([ShipDate], 8)
```

ADD_MINUTE( )

Adds a specified number of minutes to a date value.

Syntax ADD_MINUTE(date, minutesToAdd)

Arguments

date
The date or date expression that represents the start date. If a start date does not have a time value, the function assumes the time is midnight, 12:00 AM.

minutesToAdd
The number of minutes to add to the start date. If you specify a negative number, the result appears to subtract the number from the start date.

Returns The date-and-time value that results from adding the specified number of minutes to the start date.

Example The following example subtracts 30 minutes from each date in the StartTime data field:

```
ADD_MINUTE([StartTime], -30)
```

ADD_MONTH( )

Adds a specified number of months to a date value.

Syntax ADD_MONTH(date, monthsToAdd)

Arguments

date
The date or date expression that represents the start date.

monthsToAdd
The number of months to add to the start date. If you specify a negative number, the result appears to subtract the number from the start date.

Returns The date value that results from adding the specified number of months to the start date. This function always returns a valid date. If necessary, the day part of the resulting date is adjusted downward to the last day of the resulting month in the resulting year. For example, if you add one month to 1/31/08, ADD_MONTH( ) returns 2/29/08, not 2/31/08 or 2/28/08, because 2008 is a leap year.

Example The following example adds two months to each date value in the InitialRelease data field:

```
ADD_MONTH([InitialRelease], 2)
```

ADD_QUARTER( )

Adds a specified number of quarters to a date value.

Syntax ADD_QUARTER(date, quartersToAdd)

Arguments

date
The date or date expression that represents the start date.

quartersToAdd
The number of quarters to add to the start date. If you specify a negative number, the result appears to subtract the number from the start date.

Returns The date value that results from adding the specified number of quarters to the start date. A quarter is equal to three months. For example, if you add two quarters to 9/22/08, ADD_QUARTER( ) returns 3/22/09.

Example The following example adds two quarters to each date value in the ForecastClosing data field:

```
ADD_QUARTER([ForecastClosing], 2)
```

ADD_SECOND( )

Adds a specified number of seconds to a date value.

Syntax ADD_SECOND(date, secondsToAdd)

Arguments **date**
The date or date expression that represents the start date. If a start date does not have a time value, the function assumes the time is midnight, 12:00 AM.

secondsToAdd
The number of seconds to add to the start date. If you specify a negative number, the result appears to subtract the number from the start date.

Returns The date-and-time value that results from adding the specified number of seconds to the start date.

Example The following example adds 30 seconds to each date value in the StartTime data field:

```
ADD_SECOND([StartTime], 30)
```

ADD_WEEK( )

Adds a specified number of weeks to a date value.

Syntax ADD_WEEK(date, weeksToAdd)

Arguments **date**
The date or date expression that represents the start date.

weeksToAdd
The number of weeks to add to the start date. If you specify a negative number, the result appears to subtract the number from the start date.

Returns The date value that results from adding the number of weeks to the start date.

Example The following example adds two weeks to each date value in the OrderDate data field:

```
ADD_WEEK([OrderDate], 2)
```

ADD_YEAR( )

Adds a specified number of years to a date value.

Syntax ADD_YEAR(date, yearsToAdd)

Arguments **date**
The date or date expression that represents the start date.

yearsToAdd
The number of years to add to the start date. If you specify a negative number, the result appears to subtract the number from the start date.

Returns The date value that results from adding the number of years to the start date.

Example The following example adds five years to each date value in the HireDate data field:

```
ADD_YEAR([HireDate], 5)
```

BETWEEN( )

Tests if a value is between two specified values.

Syntax BETWEEN(value, upperBound, lowerBound)

Arguments **value**
The value to test. The value can be a string, numeric, or date value.

upperBound
The first value in the range of values to which to compare. String and date values must be enclosed in double quotation marks (" ").

lowerBound
The second value in the range of values to which to compare. String and date values must be enclosed in double quotation marks (" ").

Returns True if value is between upperBound and lowerBound, or equal to upperBound or lowerBound; returns false otherwise.

Examples The following example tests each value in the SalesTotal data field to see if the value is between 10000 and 20000:

```
BETWEEN([SalesTotal], 10000, 20000)
```

The following example tests each value in the CustomerName data field to see if the value is between A and M:

```
BETWEEN([CustomerName], "A", "M")
```

The following example tests each value in the ReceiptDate data field to see if the value is between 10/01/07 and 12/31/07:

```
BETWEEN([ReceiptDate], "10/01/07 12:00 AM", "12/31/07 12:00 AM")
```

The following example uses BETWEEN( ) in conjunction with the IF( ) and ADD_DAY( ) functions to calculate a shipment date. If an orderDate value is in December 2007 (between 12/1/07 and 12/31/07), add five days to the orderDate value. If an orderDate value is in a month other than December, add three days to the orderDate value.

```
IF((BETWEEN([orderDate], "12/1/07 12:00 AM", "12/31/07 12:00
   AM")), (ADD_DAY([orderDate], 5)), (ADD_DAY([orderDate], 3)))
```

CEILING( )

Rounds a number up to the nearest specified multiple.

Syntax CEILING(num, significance)

Arguments **num**
The numeric value to round up.

significance
The multiple up to which to round num.

Returns The number that results from the rounding. If num is an exact multiple of significance, no rounding occurs.

Examples CEILING( ) is commonly used to round up prices. For example, to avoid dealing with pennies, if the Price value is 20.52, CEILING( ) returns 20.55. You can round prices in a Price data field up to the nearest nickel with the following expression:

```
CEILING([Price], 0.05)
```

If the Price value is 20.52, CEILING( ) returns 20.60. If the Price value is 20.40, CEILING( ) returns 20.40. No rounding occurs because 20.40 is already a multiple of 0.1. The following example rounds prices up to the nearest dime:

```
CEILING([Price], 0.1)
```

The following example rounds prices up to the nearest dollar. If the Price value is 20.30, CEILING( ) returns 21.0.

```
CEILING([Price], 1)
```

DAY( )

Returns a number from 1 to 31 that represents the day of the month.

Syntax DAY(date)

Argument **date**
The date or date expression from which you want to extract the day.

Returns The number of the day of the month for the specified date value.

Example The following example gets the number of the day for each date value in the ShipDate data field:

```
DAY([ShipDate])
```

DIFF_DAY( )

Calculates the number of days between two date values.

Syntax DIFF_DAY(date1, date2)

Arguments **date1**
The first date or date expression to use in the calculation.

date2
The second date or date expression to use in the calculation.

Returns The number of days between date1 and date2.

Example The following example calculates the number of days between each value in the invoiceDate data field and each value in the paymentDate data field:

```
DIFF_DAY([invoiceDate],[paymentDate])
```

The results show how long it takes to pay invoices.

The following example calculates the number of days from an order date to Christmas:

```
DIFF_DAY([orderDate], "12/25/08")
```

The following example calculates the number of days from the current date to Christmas. TODAY( ) is a function that returns the current date:

```
DIFF_DAY(TODAY(), "12/25/08")
```

DIFF_HOUR( )

Calculates the number of hours between two date values.

Syntax DIFF_HOUR(date1, date2)

Arguments **date1**
The first date or date expression to use in the calculation. If the date does not have a time value, the function assumes the time is midnight, 12:00 AM.

date2
The second date or date expression to use in the calculation. If the date does not have a time value, the function assumes the time is midnight, 12:00 AM.

Returns The number of hours between date1 and date2.

Example The following example calculates the number of hours between each value in the startTime data field and each value in the finishTime data field:

```
DIFF_HOUR([startTime],[finishTime])
```

The following example calculates the number of hours from the current date to Christmas. NOW( ) is a function that returns the current date and time. If you supply a literal date as an argument, you must include the time value, as shown in the following example:

```
DIFF_HOUR(NOW(), "12/25/08 12:00 AM")
```

DIFF_MINUTE( )

Calculates the number of minutes between two date values.

Syntax DIFF_MINUTE(date1, date2)

Arguments **date1**
The first date or date expression to use in the calculation. If the date does not have a time value, the function assumes the time is midnight, 12:00 AM.

date2
The second date or date expression to use in the calculation. If the date does not have a time value, the function assumes the time is midnight, 12:00 AM.

Returns The number of minutes between date1 and date2.

Example The following example calculates the number of minutes between each value in the startTime data field and each value in the finishTime data field:

```
DIFF_MINUTE([startTime],[finishTime])
```

The following example calculates the number of minutes from the current date to Christmas. NOW( ) is a function that returns the current date and time. If you supply a literal date as an argument, you must include the time value, as shown in the following example:

```
DIFF_MINUTE(NOW(), "12/25/08 12:00 AM")
```

DIFF_MONTH( )

Calculates the number of months between two date values.

Syntax DIFF_MONTH(date1,date2)

Arguments **date1**
The first date or date expression to use in the calculation.

date2
The second date or date expression to use in the calculation.

Returns The number of months between date1 and date2. The function calculates the difference by subtracting the month number of date1 from the month number of date2. For example, if date1 is 8/1/08 and date2 is 8/31/08, DIFF_MONTH( ) returns 0. If date1 is 8/25/08 and date2 is 9/5/08, DIFF_MONTH( ) returns 1.

Example The following example calculates the number of months between each value in the askByDate data field and each value in the ShipByDate data field:

```
DIFF_MONTH([askByDate],[shipByDate])
```

The following example calculates the number of months from each value in the hireDate data field to the end of the year:

```
DIFF_MONTH([hireDate], "1/1/09")
```

DIFF_QUARTER( )

Calculates the number of quarters between two date values.

Syntax DIFF_QUARTER(date1, date2)

Arguments **date1**
The first date or date expression to use in the calculation.

date2
The second date or date expression to use in the calculation.

Returns The number of quarters between date1 and date2. DIFF_QUARTER calculates the difference by subtracting the month number of date1 from the month number of date2. A difference of three months is equal to one quarter. For example, if date1 is 8/1/08 and date2 is 10/31/08, DIFF_QUARTER( ) returns 0. If date1 is 8/25/08 and date2 is 11/5/08, DIFF_QUARTER( ) returns 1.

Example The following example calculates the number of quarters between each value in the PlanClosing data field and each value in the ActualClosing data field:

```
DIFF_QUARTER([PlanClosing],[ActualClosing])
```

The following example calculates the number of quarters from each value in the orderDate data field to the end of the year:

```
DIFF_QUARTER([orderDate], "1/1/09")
```

DIFF_SECOND( )

Calculates the number of seconds between two date values.

Syntax DIFF_SECOND(date1, date2)

Arguments

date1
The first date or date expression to use in the calculation. If the date does not have a time value, the function assumes the time is midnight, 12:00 AM.

date2
The second date or date expression to use in the calculation. If the date does not have a time value, the function assumes the time is midnight, 12:00 AM.

Returns The number of seconds between date1 and date2.

Example The following example calculates the number of seconds between each value in the startTime data field and each value in the finishTime data field:

```
DIFF_SECOND([startTime],[finishTime])
```

The following example calculates the number of seconds from the current date to Christmas. NOW( ) is a function that returns the current date and time. If you supply a literal date as an argument, you must include the time value, as shown in the following example:

```
DIFF_SECOND(NOW(), "12/24/08 12:00 AM")
```

DIFF_WEEK( )

Calculates the number of weeks between two date values.

Syntax DIFF_WEEK(date1, date2)

Arguments

date1
The first date or date expression to use in the calculation.

date2
The second date or date expression to use in the calculation.

Returns The number of weeks between date1 and date2. The function calculates the difference by subtracting the week number of date1 from the week number of date2. For example, if date1 is 1/3/08 (week 1 of the year), and date2 is 1/7/08 (week 2 of the year), DIFF_WEEK( ) returns 1.

Example The following example calculates the number of weeks between each value in the askByDate data field and each value in the shipByDate data field:

```
DIFF_WEEK([askByDate],[shipByDate])
```

The following example calculates the number of weeks from each value in the orderDate data field to the end of the year:

```
DIFF_WEEK([orderDate], "1/1/09")
```

DIFF_YEAR( )

Calculates the number of years between two date values.

Syntax DIFF_YEAR(date1, date2)

Arguments

date1
The first date or date expression to use in the calculation.

date2
The second date or date expression to use in the calculation.

Returns The number of years between date1 and date2. The function calculates the difference by subtracting the year number of date1 from the year number of date2. For example, if date1 is 1/1/08 and date2 is 12/31/08, DIFF_YEAR( ) returns 0. If date1 is 11/25/08 and date2 is 1/5/09, DIFF_YEAR( ) returns 1.

Example The following example calculates the number of years between each value in the HireDate data field and each value in the TerminationDate data field:

```
DIFF_YEAR([HireDate],[TerminationDate])
```

The following example calculates the number of years from each value in the HireDate data field to the current date. TODAY( ) is a function that returns the current date:

```
DIFF_YEAR([HireDate], TODAY())
```

FIND( )

Finds the location of a substring in a string.

Syntax FIND(strToFind, str)

FIND(strToFind, str, startPosition)

Arguments

strToFind
The substring to search for. The search is case-sensitive.

str
The string in which to search.

startPosition
The position in str where the search starts.

Returns The numerical position of the substring in the string. The first character of a string starts at 1. If the substring is not found, FIND( ) returns 0.

Example The following example searches for the substring, Ford, in each ProductName value:

```
FIND("Ford", [ProductName])
```

If the product name is 1969 Ford Falcon, FIND( ) returns 6.

The following example searches for the first hyphen (-) in each product code:

```
FIND("-", [ProductCode])
```

If the product code is ModelA-1234-567, FIND( ) returns 7.

The following example uses FIND( ) in conjunction with the LEFT( ) function to display the characters that precede the hyphen in a product code. The LEFT( ) function extracts a substring of a specified length, starting from the first character. In this example, the length of the substring to display is equal to the numerical position of the hyphen character.

```
LEFT([ProductCode], FIND("-", [ProductCode]))
```

If the product code is ModelA-1234, LEFT( ) returns the following string:

```
ModelA
```

IF( )

Returns one value if a specified condition evaluates to TRUE, or another value if the condition evaluates to FALSE.

Syntax IF(condition, doIfTrue, doIfFalse)

Arguments **condition**
The condition to test.

doIfTrue
The value to return if condition evaluates to TRUE.

doIfFalse
The value to return if condition evaluates to FALSE.

Returns Returns the doIfTrue value if condition is TRUE or the doIfFalse value if condition is FALSE.

Example The following example calculates and displays different discount amounts based on the value in the Total data field. If the Total value is greater than 5000, the discount is 15%. Otherwise, the discount is 10%.

```
IF([Total]>5000, [Total]*15%, [Total]*10%)
```

The following example uses IF( ) in conjunction with the BETWEEN( ) and ADD_DAY( ) functions to calculate a shipment date. If an orderDate value is in December 2007 (between 12/1/07 and 12/31/07), add five days to the orderDate value. If a orderDate value is in a month other than December, add three days to the orderDate value.

```
IF((BETWEEN([orderDate], "12/1/07 12:00 AM", "12/31/07 12:00
   AM")), (ADD_DAY([orderDate], 5)), (ADD_DAY([orderDate], 3)))
```

The following example checks each value in the Office data field. If the value is Boston, San Francisco, or NYC, the computed column displays U.S. If the value is something other than Boston, San Francisco, or NYC, the computed column displays Europe and Asia Pacific.

```
IF([Office]="Boston" OR [Office]="San Francisco" OR
   [Office]="NYC", "U.S.", "Europe and Asia Pacific")
```

IN( )

Tests if a value is equal to a value in a list.

Syntax IN(value, check1,..., checkN)

Arguments

value
The value to test. The value can be a string, numeric, or date value.

check1, ..., checkN
The value or values to which to compare.

Returns TRUE if value is equal to one of the check values; returns FALSE otherwise.

Example The following example tests if New Haven, Baltimore, or Cooperstown are values in the city data field. If any one of the cities is in the data field, IN( ) returns TRUE.

```
IN([city], "New Haven", "Baltimore", "Cooperstown")
```

The following example tests if 9/15/08 or 9/30/08 are values in the payDate data field. If you supply a literal date as an argument, you must include the time value, as shown in the following example:

```
IN([payDate], "9/15/08 12:00 AM", "9/30/08 12:00 AM")
```

The following example uses IN( ) in conjunction with the IF( ) function to test if Ships or Trains are values in the ProductLine data field. If Ships or Trains is a value in the field, the computed column displays Discontinued Item; otherwise, the product line value is displayed as it appears in the field.

```
IF(IN([ProductLine], "Ships", "Trains"),"Discontinued Item",
   [ProductLine])
```

ISNULL( )

Tests if a value in a specified data field is a null value. A null value means that no value exists.

Syntax ISNULL(value)

Argument **value**
The data field in which to check for null values.

Returns TRUE if a value in the specified data field is a null value; returns FALSE otherwise.

Example The following example uses ISNULL( ) in conjunction with the IF( ) function to test for null values in the BirthDate data field. If there is a null value, the computed column displays No date specified; otherwise the BirthDate value is displayed.

```
IF(ISNULL([BirthDate]), "No date specified", [BirthDate])
```

LEFT( )

Extracts a substring from a string, starting from the left-most, or first, character.

Syntax LEFT(str)

LEFT(str, n)

Arguments **str**
The string from which to extract a substring.

n
The number of characters to extract, starting from the first character.

Returns A substring of a specific length:

- If you omit n, the number of characters to extract, the function returns the first character only.
- If n is zero, the function returns an empty string.
- If n is greater than the length of the string, the function returns the entire string.

Example The following example displays the first letter of each name in the CustomerName data field:

```
LEFT([CustomerName])
```

The following example uses the LEFT( ) and FIND( ) functions to display the characters that precede the hyphen in a product code:

```
LEFT([ProductCode], FIND("-", [ProductCode]))
```

If the product code is ModelA-1234, LEFT( ) returns the following string:

```
ModelA
```

LEN( )

Counts the number of characters in a string.

Syntax LEN(str)

Argument **str**
The string expression to evaluate.

Returns The number of characters in the specified string.

Example The following example returns the length of each value in the ProductCode data field:

```
LEN([ProductCode])
```

The following example uses LEN( ) in conjunction with the RIGHT( ) and FIND( ) functions to display the characters that appear after the hyphen in a product code. RIGHT( ) extracts a substring of a specified length, starting from the last character. In this example, the length of the entire string returned by LEN( ) minus the length up to the hyphen is the number of characters to display.

```
RIGHT( [PRODUCTNAME], LEN([PRODUCTNAME] ) - (FIND("-"
   ,[PRODUCTNAME] )))
```

If the product code is ModelA-Ford, RIGHT( ) returns the following string:

```
A-Ford
```

LIKE( )

Tests if a string matches a pattern.

Syntax LIKE(str, pattern)

str
The string to evaluate.

pattern
The string pattern to match. You must enclose the pattern in double quotation marks (" "). The match is case-sensitive. You can use the following special characters in a pattern:

- A percent character (%) matches zero or more characters. For example, %ace% matches any string value that contains the substring ace, such as Facebook, and MySpace. It does not match Ace Corporation because this string contains a capital A, and not the lowercase a.
- An underscore character (_) matches exactly one character. For example, t_n matches tan, ten, tin, and ton. It does not match teen or tn.

To match a literal percent (%), underscore (_), precede those characters with two backslash (\\) characters. For example, to see if a string contains M_10, specify the following pattern:

```
"%M\\_10%"
```

Returns TRUE if the string matches the pattern; returns FALSE otherwise.

Example The following example returns true for values in the customerName field that start with D:

```
LIKE([customerName], "D%")
```

The following example returns true for productCode values that contain the substring Ford:

```
LIKE([productCode], "%Ford%")
```

The following example uses two LIKE( ) expressions to look for the substrings "Ford" or "Chevy" in each ProductName value. If a product name contains either substring, the computed column displays U.S. Model; otherwise, it displays Imported Model.

```
IF(((LIKE([ProductName], "%Ford%") = TRUE) OR (LIKE([ProductName],
   "%Chevy%") = TRUE)), "U.S. model", "Imported Model")
```

LOWER( )

Converts all letters in a string to lowercase.

Syntax LOWER(str)

Argument **str**
The string to convert to lowercase.

Returns The specified string in all lowercase letters.

Example The following example displays all the string values in the productLine data field in lowercase:

```
LOWER([productLine])
```

MATCH( )

Returns a Boolean indicating whether a pattern exists within a string.

Syntax MATCH(source, pattern)

Arguments

source
The string to evaluate.

pattern
The string pattern to match. The pattern uses ECMAScript (JavaScript) syntax, as defined in Section 15.10 of Standard ECMA-262.

Returns TRUE if the pattern matches, FALSE otherwise.

Example The following example uses ECMAScript syntax in the pattern to detect any set of characters followed by the letter C, and returns TRUE:

```
MATCH("ABC",".*C")
```

The following example checks whether the string starts with the letter X, followed by any single character, and ending with C. It returns FALSE:

```
MATCH("ABC","X.C")
```

MOD( )

Returns the modulo value for a number and a divisor.

Syntax MOD(number, divisor)

Arguments

number
The number from which to derive the mod value.

divisor
The divisor for the mod function.

Returns Returns the remainder value of number divided by divisor.

Example The following example computes the remainder of PriceEstimate data field divided by 12, returning an integer. For example, if the PriceEstimate value is 27365, MOD( ) returns 5.

```
MOD([PriceEstimate], 12)
```

MONTH( )

Returns the month for a specified date value.

Syntax MONTH(date)

MONTH(date, option)

Arguments **date**
The date or date expression whose month to get.

option
A number that represents the month format to return. Use one of the following values:

- 1 to get the month as a number from 1 to 12.
- 2 to get the full month name, for example, January. The result is locale-specific.
- 3 to get the abbreviated month name, for example, Jan. The result is locale-specific.

If you omit option, MONTH( ) returns the month as a number.

Returns The month for a specified date value.

Example The following example returns the month (1 – 12) for each value in the ShipDate data field:

```
MONTH([ShipDate])
```

The following example returns the full month name for each ShipDate value:

```
MONTH([ShipDate], 2)
```

NOT( )

Negates a Boolean expression.

Syntax NOT(expression)

Argument **expression**
The Boolean value or expression to negate.

Returns TRUE if the expression evaluates to FALSE, and FALSE if the expression evaluates to TRUE.

Example The following example uses NOT( ) in conjunction with the IF( ) and LIKE( ) functions. It tests if the value in the State data field is not CA. If the value is not CA, it returns the value of the Markup data field multiplied by 10%. If the value is CA it returns the value of the Markup data field multiplied by 15%:

```
IF(NOT(LIKE([State], "CA")),[Markup]*10%,[Markup]*15%)
```

The previous IF( ) statement is semantically equivalent to the following statement:

```
IF(LIKE([State], "CA"),[Markup]*15%,[Markup]*10%)
```

NOTNULL( )

Tests if a value in a specified data field is a non-null value.

Syntax NOTNULL(value)

Argument **value**
The data field in which to check for non-null values.

Returns TRUE if a value in the specified data field is not a null value; returns FALSE otherwise.

Example The following example uses NOTNULL( ) in conjunction with the IF( ) function to test for non-null values in the BirthDate data field. If there is a non-null value, the BirthDate value is displayed; otherwise the string "No date specified" is displayed.

```
IF(NOTNULL([BirthDate]), [BirthDate], "No date specified")
```

NOW( )

Returns the current date and time.

Syntax NOW( )

Returns The current date and time. For example:

```
Sep 23, 2008 11:56 AM
```

Example The following example uses the DIFF_MINUTE( ) and NOW( ) functions to calculate the number of minutes from the current date and time to Christmas:

```
DIFF_MINUTE(NOW(), "12/25/08 12:00 AM")
```

QUARTER( )

Returns the quarter number for a specified date value.

Syntax QUARTER(date)

Arguments **date**
The date or date expression whose quarter number to get.

Returns A number from 1 to 4 that represents the quarter for a specified date value. Quarter 1 starts in January.

Example The following example displays the quarter number for each value in the CloseDate data field:

```
QUARTER([CloseDate])
```

The following example displays a string—Q1, Q2, Q3, or Q4—for each value in the CloseDate data field:

```
"Q" & QUARTER([CloseDate])
```

RIGHT( )

Extracts a substring from a string, starting from the right-most, or last, character.

Syntax RIGHT(str)

RIGHT(str, n)

Arguments **str**
The string from which to extract a substring.

n
The number of characters to extract, starting from the last character.

Returns A substring of a specific length.

- If you omit n, the number of characters to extract, the function returns the last character only.
- If n is zero, the function returns an empty string.
- If n is greater than the length of the string, the function returns the entire string.

Example The following example displays the last four characters of each value in the ProductCode data field:

```
RIGHT([ProductCode], 4)
```

The following example uses RIGHT( ) in conjunction with the LEN( ) and FIND( ) functions to display the characters that appear after the hyphen in a product code. This example assumes that the number of characters after the hyphen varies. Therefore, the length of the entire string (returned by LEN( )) minus the length up to the hyphen (returned by FIND( )) is the number of characters to display.

```
RIGHT([ProductCode], (LEN([ProductCode]) - FIND("-" ,
   [ProductCode])))
```

If the product code is ModelA-Ford, RIGHT( ) returns Ford. If the product code is ModelB-Toyota, RIGHT( ) returns Toyota.

ROUND( )

Rounds a number to a specified number of digits.

Syntax ROUND(num)

ROUND(num, dec)

Arguments

num
The number to round.

dec
The number of digits up to which to round num. If you omit dec, ROUND( ) assumes 0.

Returns A number rounded to a specified number of digits.

Example The following example rounds the numbers in the PriceEstimate data field to return an integer. For example, if the PriceEstimate value is 1545.50, ROUND( ) returns 1546. If the PriceEstimate value is 1545.25, ROUND( ) returns 1545.

```
ROUND([PriceEstimate])
```

The following example rounds the numbers in the PriceEstimate data field to one decimal place. For example, if the PriceEstimate value is 1545.56, ROUND( ) returns 1545.6. If the PriceEstimate value is 1545.23, ROUND( ) returns 1545.2.

```
ROUND([PriceEstimate], 1)
```

The following example rounds the numbers in the PriceEstimate data field to one digit to the left of the decimal point. For example, if the PriceEstimate value is 1545.56, ROUND( ) returns 1550. If the PriceEstimate value is 1338.50, ROUND( ) returns 1340.

```
ROUND([PriceEstimate], -1)
```

ROUNDDOWN( )

Rounds a number down to a specified number of digits.

Syntax ROUNDDOWN(num)

ROUNDDOWN(num, dec)

Arguments

num
The number to round down.

dec
The number of digits up to which to round num down. If you omit dec, ROUND( ) assumes 0.

Returns A number rounded down to a specified number of digits.

Example The following example rounds down the numbers in the PriceEstimate data field to return an integer. For example, if the PriceEstimate value is 1545.25, ROUNDDOWN( ) returns 1545. If the PriceEstimate value is 1545.90, ROUNDDOWN( ) returns 1545.

```
ROUNDDOWN([PriceEstimate])
```

The following example rounds down the numbers in the PriceEstimate data field to one decimal place. For example, if the PriceEstimate value is 1545.56, ROUNDDOWN( ) returns 1545.5. If the PriceEstimate value is 1545.23, ROUNDDOWN( ) returns 1545.2.

```
ROUNDDOWN([PriceEstimate], 1)
```

The following example rounds the numbers in the PriceEstimate data field down to one digit to the left of the decimal point. For example, if the PriceEstimate value is 1545.56, ROUNDDOWN( ) returns 1540. If the PriceEstimate value is 1338.50, ROUNDDOWN( ) returns 1330.

```
ROUNDDOWN([PriceEstimate], -1)
```

ROUNDUP( )

Rounds a number up to a specified number of digits.

Syntax

ROUNDUP(num)

ROUNDUP(num, dec)

Arguments

num
The number to round up.

dec
The number of digits up to which to round num up. If you omit dec, ROUND( ) assumes 0.

Returns

A number rounded up to a specified number of digits.

Example

The following example rounds up the numbers in the PriceEstimate data field to return an integer. For example, if the PriceEstimate value is 1545.25, ROUNDUP( ) returns 1546. If the PriceEstimate value is 1545.90, ROUNDUP( ) returns 1546.

```
ROUNDUP([PriceEstimate])
```

The following example rounds up the numbers in the PriceEstimate data field to one decimal place. For example, if the PriceEstimate value is 1545.56, ROUNDUP( ) returns 1545.6. If the PriceEstimate value is 1545.23, ROUNDUP( ) returns 1545.3.

```
ROUNDUP([PriceEstimate], 1)
```

The following example rounds up the numbers in the PriceEstimate data field to one digit to the left of the decimal point. For example, if the PriceEstimate value is 1545.56, ROUNDUP( ) returns 1550. If the PriceEstimate value is 1338.50, ROUNDUP( ) returns 1340.

```
ROUNDUP([PriceEstimate], -1)
```

SEARCH( )

Finds the location of a substring in a string. The substring can contain wildcard characters.

Syntax SEARCH(pattern, str)

SEARCH(pattern, str, startPosition)

Arguments **pattern**
The string pattern to search for. You must enclose the pattern in double quotation marks (" "). You can use the following special characters in a pattern:

- An asterisk (*) matches zero or more characters, including spaces. For example, t*n matches tn, tin, and teen.
- A question mark (?) matches exactly one character. For example, t?n matches tan, ten, tin, and ton. It does not match teen or tn.

str
The string in which to search.

startPosition
The position in str where the search starts.

Returns The numerical position of the string pattern in the string. The first character of a string starts at 1. If the substring is not found, SEARCH( ) returns 0.

Example The following example searches for the string pattern, S*A, in each product code. If the product name is KBS5412A, SEARCH( ) returns 3.

```
SEARCH("S*A", [ProductCode])
```

The following example uses SEARCH( ) in conjunction with the LEFT( ) function to display the characters that precede the first space character in a product name. The LEFT( ) function extracts a substring of a specified length, starting from the first character. In this example, the length of the substring to display is equal to the numerical position of the space character.

```
LEFT([ProductName], SEARCH(" ", [ProductName]))
```

If the product name is 1969 Ford Falcon, the expression returns 1969.

SQRT( )

Calculates the square root of a number.

Syntax SQRT(num)

Argument **num**
The number, or numeric expression that specifies the number, for which you want to find the square root. The number must be a positive number.

Returns A number that is the square root of num.

Example The following example calculates the square root of each value in the LotSize data field:

```
SQRT([LotSize])
```

The following example uses SQRT( ) to calculate the actual distance travelled uphill, given the base distance and elevation values. This example applies the Pythagorean theorem, which states that $a^2 + b^2 = c^2$. Using this theorem, the actual distance traveled is c, which means we want to calculate

$$c = \sqrt{a^2 + b^2}$$

which translates to the following expression:

```
SQRT((([Distance] * [Distance]) + ([Elevation] * [Elevation])))
```

TODAY( )

Returns the current date that includes a time value of midnight, 12:00 AM.

Syntax TODAY( )

Returns The current date in the following format:

```
Sep 25, 2008 12:00 AM
```

Example The following example calculates the number of days from the current date to Christmas:

```
DIFF_DAY(TODAY(), "12/25/08")
```

The following example calculates the number of years from each value in the HireDate data field to the current date:

```
DIFF_YEAR([HireDate], TODAY())
```

TRIM( )

Removes the leading and trailing blanks from a specified string. TRIM( ) does not remove blank characters between words.

Syntax TRIM(str)

Argument **str**
The string from which to remove leading and trailing blank characters.

Returns A string with all leading and trailing blank characters removed.

Example The following example uses TRIM( ) to remove all leading and trailing blank characters from values in the FirstName and LastName data fields. The expression uses the & operator to concatenate each trimmed FirstName value with a space, then with each trimmed LastName value.

```
TRIM([FirstName]) & " " & TRIM([LastName])
```

TRIMLEFT( )

Removes the leading blanks from a specified string.

Syntax TRIMLEFT(str)

Arguments **str**
The string from which to remove the leading blank characters.

Returns A string with all leading blank characters removed.

Example The following example concatenates a literal string with each value in the customerName data field. TRIMLEFT( ) removes all blank characters preceding the customerName value so that there are no extra blank characters between the literal string and the customerName value.

```
"Customer name: " & TRIMLEFT([customerName])
```

TRIMRIGHT( )

Removes the trailing blanks from a specified string.

Syntax TRIMRIGHT(str)

Argument **str**
The string from which to remove the trailing blank characters.

Returns A string with all trailing blank characters removed.

Example The following example concatenates each value in the Comment data field with a semicolon, then with a value in the Action data field. TRIMRIGHT( ) removes all blank characters after the Comment value so that there are no extra blank characters between the Comment string and the semicolon.

```
TRIMRIGHT([Comment]) & "; " & [Action]
```

UPPER( )

Converts all letters in a string to uppercase.

Syntax UPPER(str)

Argument **str**
The string to convert to uppercase.

Returns The specified string in all uppercase letters.

Example The following example displays all the string values in the customerName data field in all uppercase:

```
UPPER([customerName])
```

WEEK( )

Returns a number from 1 to 52 that represents the week of the year.

Syntax WEEK(date)

Argument **date**
The date or date expression whose week of the year to get.

Returns A number that represents the week of the year for the specified date value.

Example The following example gets the week number of the year for each date value in the ShipDate data field:

```
WEEK([ShipDate])
```

WEEKDAY( )

Returns the day of the week for a specified date value.

Syntax WEEKDAY(date, option)

Arguments **date**
The date or date expression from which you want to get the day of the week.

option
A number that represents the weekday format to return. Use one of the following values:

- 1 to get the day as a number from 1 (Sunday) to 7 (Saturday).
- 2 to get the day as a number from 1 (Monday) to 7 (Sunday).
- 3 to get the day as a number from 0 (Monday) to 6 (Sunday).
- 4 to get the full weekday name, for example, Wednesday. The result is locale-specific.
- 5 to get the abbreviated weekday name, for example Wed. The result is locale-specific.

If you omit option, WEEKDAY( ) assumes option 1.

Returns The day of the week for a specified date value.

Example The following example gets the full weekday name for each date value in the DateSold data field:

```
WEEKDAY([DateSold], 4)
```

YEAR( )

Returns the four-digit year value for a specified date value.

Syntax YEAR(date)

date
The date or date expression from which you want to extract the year part.

Returns The number that represents the four-digit year for the specified date value.

Example The following example gets the four-digit year for each date value in the ShipDate data field, and adds 15 to the four-digit year. For example, if the ShipDate value is Sep 16, 2008, YEAR( ) returns 2023.

```
(YEAR([ShipDate]) + 15)
```

Functions used in aggregate calculations

This section describes the range of functions that perform aggregate calculations. In BIRT Studio, you can perform aggregate calculations across the data rows in a group, section, or across an entire report table, as shown in Table 9-1.

Table 9-1 Aggregate functions

Aggregate function	Description
AVERAGE	Returns the average, or mean for a set of data rows. For example, if a report column contains values 5, 2, 7, and 10, AVERAGE returns 6.
COUNT	Counts the number of data rows. If a column contains values 5, 2, 7, and 10, COUNT returns 4.
COUNTVALUE	Counts the number of unique values in a set of data rows. If a report column contains values 5, 2, 5, 7, and 10, COUNTVALUE returns 4.
FIRST	Returns the first value in set of data rows. If a report column contains data rows 5, 2, 7, and 10, FIRST returns 5.

Table 9-1 Aggregate functions (continued)

Aggregate function	Description
LAST	Returns the last value in a set of data rows. If a report column contains data row values 2, 5, 7, and 10, LAST returns 10.
MAX	Returns the largest value in a set of data rows. If a report column contains data row values 5, 2, 7, and 10, MAX returns 10. For string values, MAX returns the last alphabetical value. For date values, MAX returns the latest date.
MEDIAN	Returns the median, or middle value in a set of data rows. If a report column contains values, 5, 2, 7, and 10, MEDIAN returns 6.
MIN	Returns the smallest value in a set of data rows. If a report column contains data row values 5, 2, 7, and 10, MIN returns 2. For string data, MIN returns the first alphabetical value. For date values, MIN returns the earliest date.
MODE	Returns the mode, or the value that occurs most frequently in a set of data rows. If a report column contains values, 5, 2, 5, 7, and 10, MODE returns 5.
QUARTILE	Returns the quartile value in a set of data rows, given a specified quart (0-4). A quartile can be defined as any three values that divide a set of values into four equal parts, such that each part represents 1/4 of the set of values. MIN, MEDIAN, and MAX return the same value as QUARTILE when quart is equal to 0, 2, and 4 respectively. If a set of data rows contains 50, 75, 80, 90, and 95, and you specify a quart of 2, QUARTILE returns 80.
STDDEV	Returns the standard deviation of a set of data rows. Standard deviation is a statistic that shows how widely values disperse from the mean value. If a set of data rows contains 50, 75, 80, 90 and 95, STDDEV returns 17.536.
SUM	Adds all the values in a set of data rows. If a report column contains 50, 75, 80, 90, and 95, SUM returns 390.

(continues)

Table 9-1 Aggregate functions (continued)

Aggregate function	Description
VARIANCE	Returns the variance of a set of data rows. Variance is a statistical measure expressing large the size of the differences between the values. The variance increases as the differences between the numbers increase. If a set of data rows contains 50, 75, 80, 90, and 95, VARIANCE returns 307.5. If a set of data rows contains 5, 2, 5, 7, and 10, VARIANCE returns 8.7.
WEIGHTEDAVE	Returns the weighted average value in a set of data rows, given weights specified in another set of values. In a weighted average, each number is assigned a weight or degree of importance. These weights determine the relative importance of each number on the average. Grades are often computed using a weighted average. For example, for a set of scores 50, 75, 80, 90, and 95, with respective weights, 10, 25, 15, 30, and 20, WEIGHTEDAVE returns 81.75.

Operators

This section is a complete reference to all of the operators that you can use when you create expressions. This reference organizes the operators into the following categories:

- Operators in computed column expressions
- Operators in conditional formatting and filter condition expressions

Operators in computed column expressions

Table 9-2 lists the operators you can use when you write expressions for a computed column.

Table 9-2 Operators in computed column expressions

Operator	Use to	Example
+	Add two or more numeric values.	[OrderAmount] + [SalesTax]
-	Subtract one numeric value from another.	[OrderAmount] - [Discount]
*	Multiply numeric values.	[Price] * [Quantity]
/	Divide numeric values.	[Profit]/12
^	Raise a numeric value to a power.	[Length]^2
%	Specify a percent.	[Price] * 80%
=	Test if two values are equal.	IF([ProductName] = "1919 Ford Falcon", "Discontinued Item", [ProductName])
>	Test if one value is greater than another value.	IF([Total] > 5000, [Total]*15%, [Total]*10%)
<	Test if one value is less than another value.	IF([SalePrice] < [MSRP] , "Below MSRP", "Above MSRP")
>=	Test if one value is greater than or equal to another value.	IF([Total] >= 5000, [Total]*15%, [Total]*10%)
<=	Test if one value is less than or equal to another value.	IF([SalePrice] <= [MSRP], "Below or equal to MSRP", "Above MSRP")

(continues)

Table 9-2 Operators in computed column expressions (continued)

Operator	Use to	Example
<>	Test if two values are not equal.	IF([Country] <> "USA", "Imported product", "Domestic product")
AND	Test if two or more conditions are true.	IF(([Gender] = "Male" AND [Salary] >= 150000 AND [Age] < 50), "Match found", "No match")
OR	Test if any one of multiple conditions is true.	IF(([City] = "Boston") OR ([City] = "San Francisco"), "U.S.", "Europe and Asia")
&	Concatenate string values.	[FirstName] & " " & [LastName]

Operators in conditional formatting and filter condition expressions

Table 9-3 lists the operators you can use when you create expressions for conditional formatting and filter conditions.

Table 9-3 Operators in conditional formatting and filter condition expressions

Operator	Use to	Example
Between	Test if a column value is between two specified values.	Profit Between 1000 and 2000
Bottom N	Test if a column value is among the lowest *n* values.	SalesAmount Bottom N 10
Bottom Percent	Test if a column value is in the bottom *n* percent of all values.	SalesAmount Bottom Percent 5
Equal to	Test if a column value is equal to a specified value.	Country Equal to France
Greater Than	Test if a column value is greater than a specified value.	Total Greater Than 5000
Greater Than or Equal to	Test if a column value is greater than or equal to a specified value.	Total Greater Than or Equal to 5000
In	Test if a column value is in the list of specified values. Usage similar to the Any Of operator.	Country In USA, Canada, Mexico
Is Blank	Test if a column value is blank (" "). This operator applies only to string values.	E-mail Is Blank
Is False	Test if a column value is False.	LoanApproved Is false

Table 9-3 Operators in conditional formatting and filter condition expressions

Operator	Use to	Example
Is Not Blank	Test if a column value is not blank. This operator applies only to string values.	Email Is Not Blank
Is Not Null	Test if a column value is not a null value. A null value means that no value is supplied.	CreditLimit Is Not Null
Is Null	Tests if a column value is a null value.	CreditLimit Is Null
Is True	Test if a column value is True.	LoanApproved Is true
Less Than	Test if a column value is less than a specified value.	Total Less Than 5000
Less Than or Equal to	Test if a column value is less than or equal to a specified value.	Total Less Than or Equal to 5000
Like	Test if a column value matches a string pattern.	ProductName Like %Ford%
Match	Test if a column value matches a string pattern.	ProductCode Match S20
Not Between	Test if a column value is not between two specified values.	Profit Not Between 1000 and 2000
Not Equal to	Test if a column value is not equal to a specified value.	Country Not Equal to France
Not In	Test if a column value is not in the specified list of values.	Country Not In USA, Canada, Mexico
Not Like	Test if a column value does not match a string pattern.	ProductName Not Like %Ford%
Not Match	Test if a column value does not match a string pattern.	Product Code Not Match S10
Top N	Test if a column value is among the top *n* values.	SalesAmount Top N 10
Top Percent	Test if a column value is in the top *n* percent of all values.	SalesAmount Top Percent 5

Chapter

10

Using data from multiple information objects

This chapter contains the following topics:

- Requirements for using multiple information objects
- Joining information objects
- Getting current data from an information object

Requirements for using multiple information objects

In “Selecting a data source,” in Chapter 1, “Getting started,” you learned how to select an information object, a data object, or a data set that is defined in a template to retrieve a particular set of data to use in your report. The data fields you insert in a table in a report design typically come from a single information object or a single data set.

For usability and performance reasons, a data administrator can organize data in a suite of information objects rather than in a single information object. For example, rather than include all customer data and order data in a single information object, a data administrator can organize data in two information objects; one information object retrieves customer data, and another information object retrieves order data. To create a report that displays order data for every customer, you would use data from both information objects.

You can use data from two or more information objects if the following requirements are met:

- You have been assigned an advanced user role.
- You have access to multiple information objects that can be combined.

In addition, you should have knowledge of the following:

- The principles of combining, or joining, database tables. To use data from two or more information objects, you join information objects in a similar manner.
- The structure and the relationship among the information objects whose data you want to join.

You can join information objects only. You cannot join data sets in a template, or those in a data object.

Joining information objects

Joining information objects is similar to joining tables in a database. As in a table join, you join information objects through a common field. For example, suppose one information object retrieves customer data, such as name, customer ID, address, phone, and credit limit. Another information object retrieves order data, such as order number, order date, ship date, status, and customer ID. Both information objects include a customer ID field, which you use to join the information objects.

You can retrieve data from any number of information objects as long as there is a relationship, identified by common fields, among the information objects. Figure 10-1 shows three information objects and the fields that join them. The

Customer IO is joined to the Orders IO through the customerNumber field, and the Orders IO is joined to the OrderDetails IO through the orderNumber field.

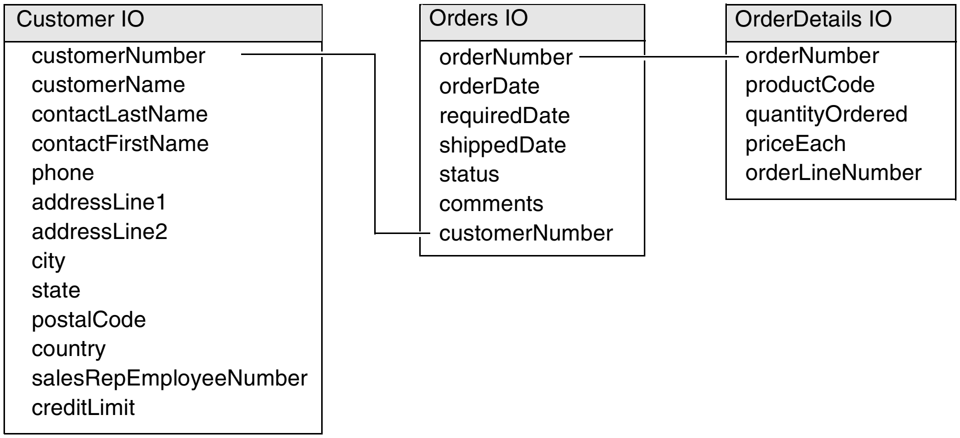


Figure 10-1 Joining information objects through common fields

How to join information objects

This procedure assumes you have created a new report design and selected an information object when prompted:

1 Under Available Data, choose Modify, as shown in Figure 10-2. This figure also shows an example of Available Data displaying the data fields in the first information object, Customers.

Figure 10-2 Choosing Modify in Available Data

Join displays the information object, Customers, under List of Information Objects, as shown in Figure 10-3.

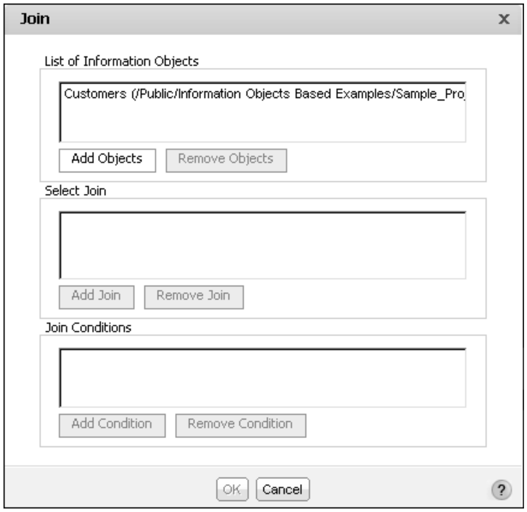


Figure 10-3 Join displaying the current information object used in the report

2 Select a second information object to join to the first information object:

1 Choose Add Objects.

2 On Select Information Object, browse through the folders to find and select an information object. Figure 10-4 shows an example of Select Information Object displaying the contents of a folder, which includes two information objects.

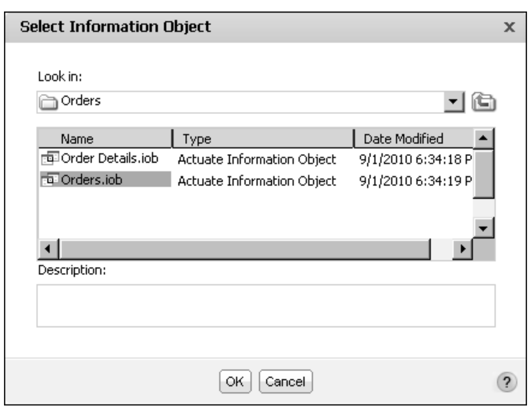


Figure 10-4 Select Information Object displaying the contents of a folder

3 Choose OK. The second information object appears in List of Information Objects on Join.

3 Join the information objects:

1 On Join, in Select Join, choose Add Join. Add Join displays two fields, Left and Right, which contain the names of the information objects.

2 In Left, select one information object. Then, in Right, select the other information object. Figure 10-5 shows an example where CustomerData and OrderData are selected in the Left and Right fields, respectively.

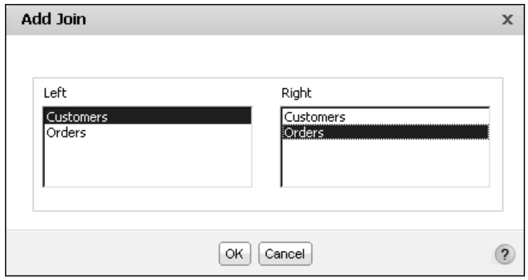


Figure 10-5 Using Join to select two information objects to join

3 Choose Add. Select Join displays the information objects selected for the join, as shown in Figure 10-6.

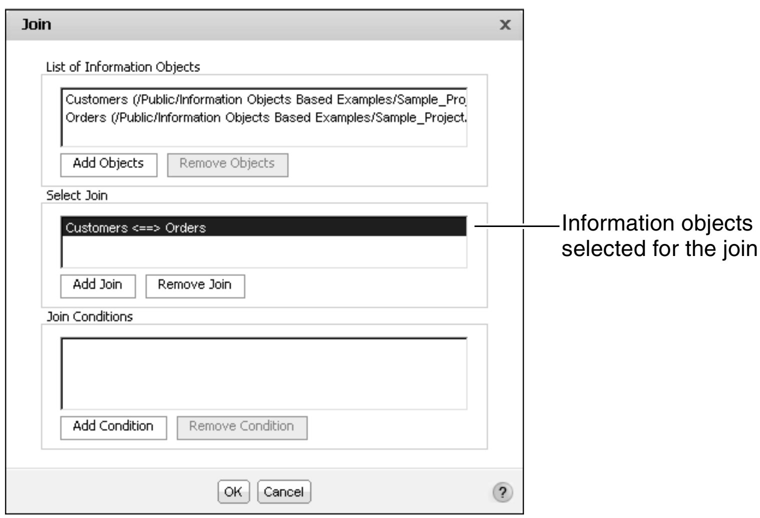


Figure 10-6 Examining the information objects selected for the join

4 Specify the join condition:

1 In Join, in Join Conditions, choose Add Condition.

Add Join Condition lists all the fields in the first and second information objects.

2 In Left, select a field. In Right, select a field that contains the same data. The two fields do not have to have identical names, but they must have the same data type.

3 In Condition, select a comparison operator. Typically, you use the default operator, equal to. Figure 10-7 shows an example of a completed join condition. The join condition joins the information objects on their respective customerNumber fields. This condition retrieves only those rows in which the customerNumber values match.

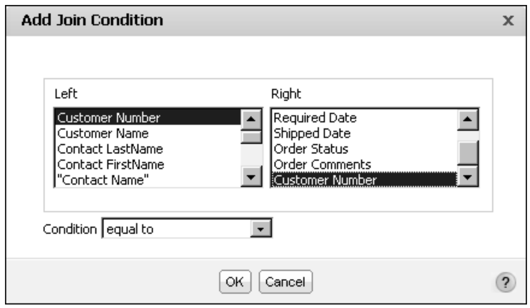


Figure 10-7 Adding a join condition

4 Choose OK. Join Conditions displays the specified join condition, as shown in Figure 10-8.

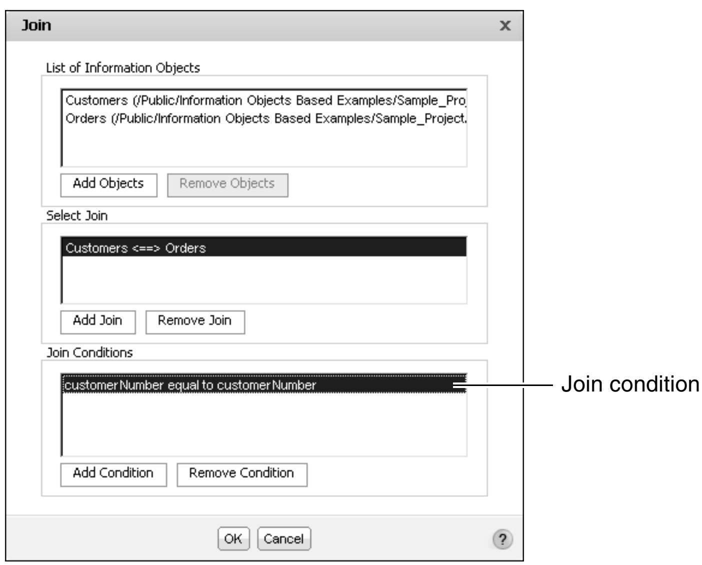


Figure 10-8 Examining the completed join

You have finished joining two information objects.

5 Repeat the previous steps to add and join additional information objects. Choose OK when you finish joining the information objects.

Available Data displays all the data fields from the joined information objects that you can insert in a table in your report design. In Figure 10-9, Available Data shows the data fields in three information objects, CustomerData, OrderData, and OrderDetailsData.

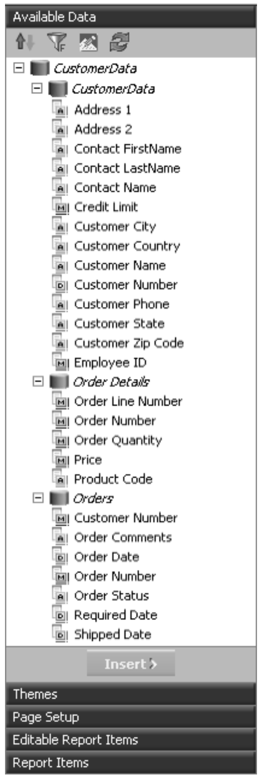


Figure 10-9 Examining the fields of three joined information objects

Getting current data from an information object

When you select an information object, BIRT Studio makes the data fields in the information object available to your report design. The information object remains on the server and can be modified after you design your report. For example, a data administrator can add or remove fields from an information object.

To ensure that your report design uses current data from an information object, you can synchronize your data set and the data set in the modified information object. To synchronize a data set, you must have an advanced user role.

How to synchronize data sets

In Available Data, choose Synchronize Data Sets, as shown in Figure 10-10. If a new field has been added to the information object, the new field appears in the list. If a field is removed from the information object, the field is removed from the list. If your report design uses a field that is no longer available, BIRT Studio displays an error message. To fix it, delete the column from the report design.

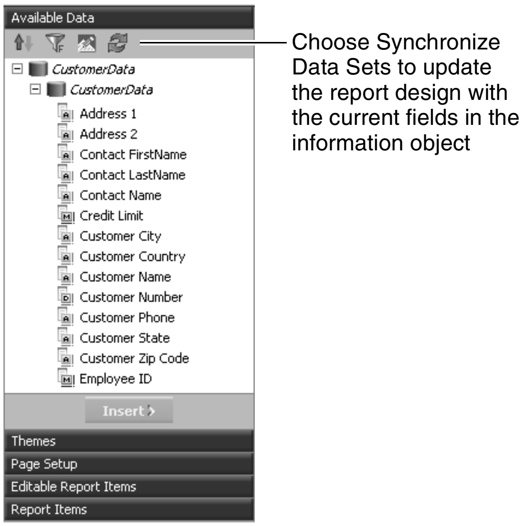


Figure 10-10 Synchronizing data sets to update the list of fields

Part Two

Customizing and Integrating BIRT Studio

Chapter

11

Introduction to BIRT Studio customization

This chapter covers the following topics:

- About BIRT Studio and the Actuate Information Console
- Overview of customization tasks
- Required Actuate software
- Understanding the context root

About BIRT Studio and the Actuate Information Console

BIRT Studio is a licensed option of BIRT iServer. A user accesses BIRT Studio from Actuate Information Console, a web-based application that provides reporting services. Actuate Information Console supports accessing, running, and viewing reports through the user interface part of the BIRT Studio, which installs with the Information Console. iServer modifies and generates BIRT Studio reports.

Figure 11-1 illustrates how the Information Console and iServer communicate in fulfilling BIRT Studio requests. The BIRT Studio is implemented as an AJAX client application, residing on the Information Console. BIRT Studio connects to the Information Console and iServer through two servlets, ERNIViewerServlet and ERNIServerServlet, using HTTP and SOAP protocols.

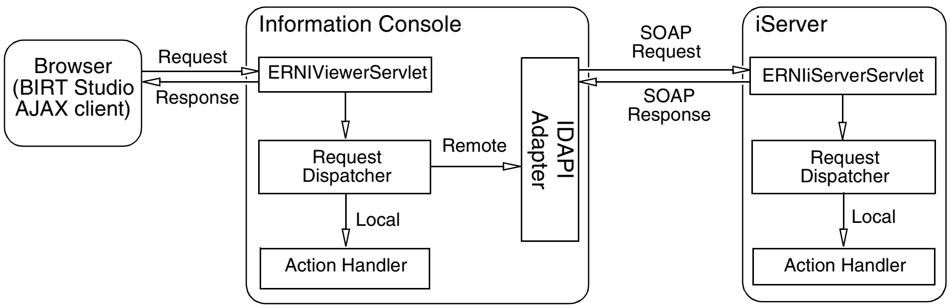


Figure 11-1 Understanding the BIRT Studio process flow

- Every time a user initiates a report creation, modification, or generation in BIRT Studio, the AJAX client forms an HTTP GET/POST request and sends the request to the BIRT Studio ERNIViewerServlet servlet in the Information Console.
- The ERNIViewerServlet receives the request and passes it to the Requester Dispatcher.
- If the request cannot be handled locally, the IDAPI Adapter constructs a SOAP request, based on the information from AJAX client request and forwards the request to iServer.
- ERNIServerServlet in iServer receives the request and passes it to Requester Dispatcher.
- The corresponding action handler processes the request and returns a SOAP response to the Information Console.
- The Information Console sends the response to the AJAX client.

You can install the Information Console separately or with iServer. The Welcome page for the Information Console appears in your web browser, as shown in Figure 11-2.

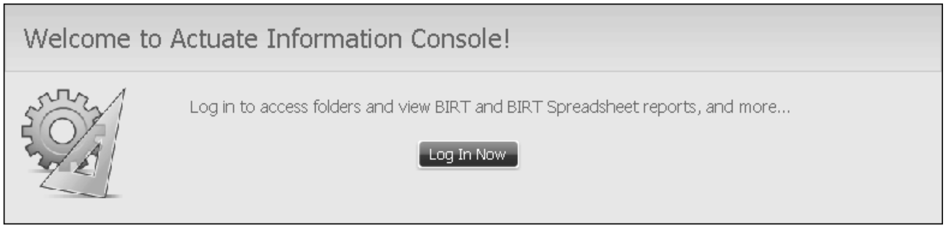


Figure 11-2 Using Actuate Information Console with iServer

Overview of customization tasks

Often, organizations customize web applications to offer users a specific suite of services. Organizations also personalize the look and feel of these applications to reflect their corporate styles. You can customize and personalize both Actuate Information Console and BIRT Studio. The following list of customization tasks are specific to BIRT Studio:

- Create a suite of report templates on which users can base their reports. BIRT Studio includes a few predefined templates. Custom templates can better suit the types of reports that specific users create.
- Provide sets of data that users can use in their reports. This task is required for BIRT Studio users to access the data they need for their reports.
- Customize the appearance and functionality of BIRT Studio. For example, if you do not want users to create computed fields in a report, you can disable that functionality.
- Deploy multiple instances of Actuate Information Console and BIRT Studio to provide different reporting services to different groups within your organization. For example, users in the marketing group can log into an Actuate Information Console instance that displays different pages and provides access to sales and marketing data only.

This section provides information about performing these customization tasks.

Required Actuate software

To customize BIRT Studio, you must have the following products:

- BIRT iServer. Actuate Information Console and Management Console install with either product.
- Actuate BIRT Designer Professional. You use this desktop application to create report templates and data objects or information objects, which provide data to reports.

Understanding the context root

BIRT Studio customization requires that you modify configuration files that install with the Information Console. The paths to these files differ depending on how you install the Information Console and iServer.

The directory in which the Information Console web service resides is called the context root. The context root contains the Java Server Pages (JSPs) to which the application server routes requests from the user's browser for Information Console web content. To locate a file, this manual refers to the following paths:

- For a typical Windows installation where Actuate Information Console is installed separately, the default context root is:

  ```
  C:\Program Files\Actuate11\iPortal\iportal
  ```

- For a typical Windows installation where Actuate Information Console is installed with BIRT iServer, the default context root is:

  ```
  C:\Program Files\Actuate11\iServer\servletcontainer\iportal
  ```

For example, to change the category of report templates that appears by default in the Report Template dialog box, you edit web.xml, which is located in:

```
<context root>\WEB-INF
```

Some files that you modify to customize BIRT Studio are files whose content is cached while the application server runs. To view the changes that you make to these files, you must restart the application server to refresh the user interface. Depending on the Actuate products in your installation, you complete one of the following tasks in Control Panel➛Administrative Tools➛Services to refresh the user interface:

- For a Windows installation where Actuate Information Console is installed separately, restart the application server. If you are using the Apache Tomcat application server that installs with Actuate Information Console, restart the Apache Tomcat for Actuate Information Console service.
- For a Windows installation where Actuate Information Console is installed with BIRT iServer, restart the Actuate BIRT iServer service.

Chapter

12

Creating and publishing report templates

This chapter covers the following topics:

- About report templates
- Design considerations
- Creating a report template
- Providing data with a report template
- Creating themes for a report template
- Publishing a template
- Setting the default template category
- Publishing resources associated with a template

About report templates

A report template defines a basic report structure on which new reports are based. The BIRT Studio user always creates a new report by selecting a template first.

BIRT Studio includes several predefined templates. Often, however, users request custom templates that better suit the data they want to present. In addition, an organization typically requires reports with a particular look and feel. You create templates using the Report Design perspective in Actuate BIRT Designer Professional, an Eclipse-based application for creating reports.

Design considerations

A template typically contains visual elements, such as tables, charts, and labels. It can, however, also contain defined data sets, which specify the data to display in a report. A template can even be a complete report that presents professionally formatted data.

Before you begin creating templates, gather the following basic information:

- What data will the BIRT Studio user use in a report?
- How does the user want to present the data?
- What does the user need as a starting point for a report?

The rest of this section provides tips for creating effective templates.

Separating or combining visual and data elements

When designing a template, one of the key decisions you make is whether to include both visual and data elements in the template or keep them separated. Good design typically dictates that templates contain only visual elements, and that data objects or information objects contain the data. You might decide, however, to create templates that include data, because some users, particularly inexperienced users, prefer to view a report without having to do anything other than select the template.

Templates that contain only visual elements are more versatile. A single template can be used for different reports that present different data. The user can mix and match data objects or information objects with templates. From a development and administrative perspective, separating presentation from data can be efficient and optimal, because template design and data retrieval can be accomplished by developers with design expertise and data-retrieval expertise, respectively. This strategy, however, requires that the template designer and the data architect coordinate to ensure that the templates and data are suitable for use together.

Designing themes

BIRT Studio provides several options for formatting the contents of a report. The standard toolbar provides the user with formatting options to modify the font, color, style, text alignment, and other properties of individual report items. The report design area on the left side of BIRT Studio displays themes, if any are provided, that the user can select to apply a set of styles to the entire report.

A theme is a collection of styles. The concept and functionality of styles are similar to styles in Microsoft Word and cascading style sheets (CSS). Designers create a theme to apply a consistent style, or look and feel, to a report. A theme, for example, can specify the colors used in a report, the font used for different text, label, or data items, the spacing or alignment of items, border styles, and so on.

BIRT Studio provides three themes with the default templates, as shown in Figure 12-1.

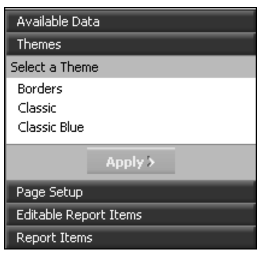


Figure 12-1 Themes provided with the default templates

When you create a template, consider creating different themes, so that the user can choose from multiple styles. While the creation of a theme is optional, it is standard practice among designers, similar to the use of cascading style sheets with web pages. Themes are stored in a BIRT library file, separate from the template file.

Defining all the styles in a theme within a library, rather than applying formats to individual report items in the template, makes it easier to maintain and update the appearance of a template. When a user requests new or modified styles to use with a particular template, all you do is update the theme in the library, then publish the latest version of the library. You do not need to modify the template file.

Improving usability

A template should be intuitive to use. The user should be able to quickly determine how to use the elements in a template, and be able to freely edit most elements. The following are some guidelines for improving the usability of a template:

- Set the general properties of the template file:

- The display name of the template
- A brief description of the template
- The image to use as the thumbnail image of the template

These properties are not required. However, if you do not specify a display name, the name of the template file is used. If you do not supply a description, the word "null" appears, and if you do not specify an image, the user sees a grey box with the x icon. Figure 12-2 shows what the user sees on the Report Template page when a template's properties are set and when the properties are not set.

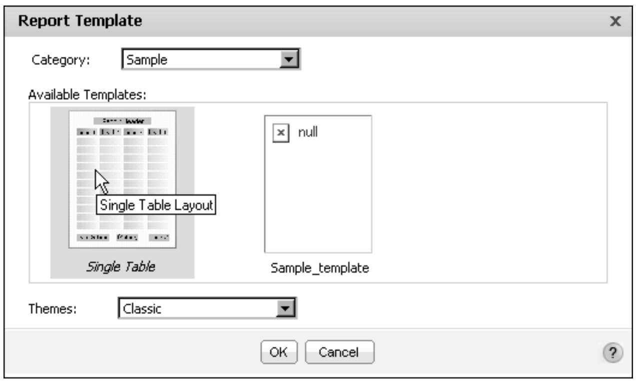


Figure 12-2 Displaying two templates, one with properties set, one without properties set

- Decide which report elements in the template are editable. Examples of editable elements include label elements for displaying report titles, section titles, or column headings, and empty tables into which users insert data. Examples of non-editable elements include company logos and standard copyright statements.
- Provide meaningful names for each report element, so the BIRT Studio user can easily identify the type and purpose of the element. If you do not specify a name, BIRT Studio displays the name of the element type, such as Text or Label. If your template contains three labels, and you do not specify a name for any of the labels, BIRT Studio displays Label three times under Report Items. Report Items lists all the elements in the template, so users can choose whether to display the elements in the report. Figure 12-3 shows one of the default templates with several elements listed under Report Items. The elements have descriptive names.
- Provide instructions for using each editable element. For example, a table can display a message, such as "To insert data in this report, drag the columns you want from Available Data and drop them in this table." Figure 12-3 shows a table with detailed instructions.

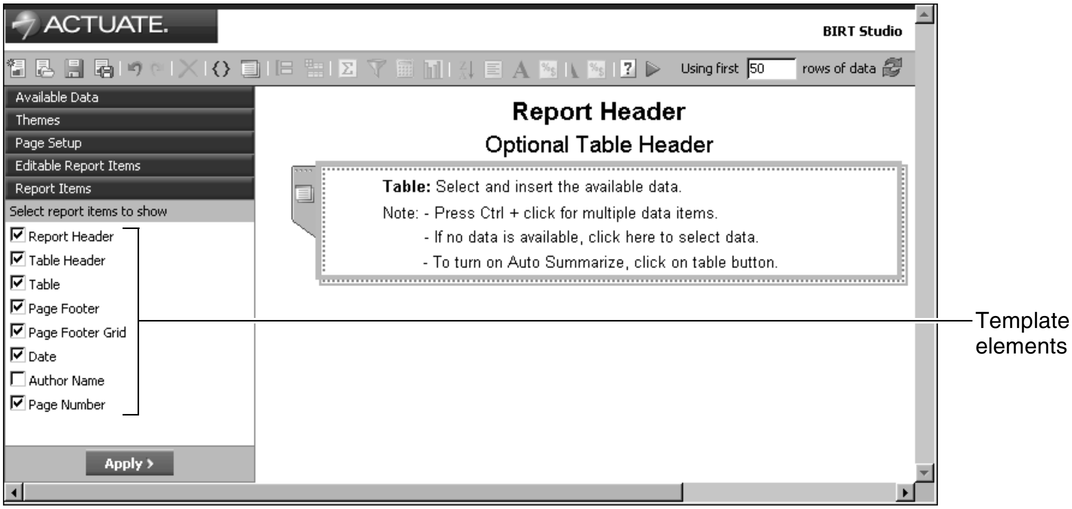


Figure 12-3 Report Items listing all the template elements

Creating a report template

You use Actuate BIRT Designer Professional to create report templates. You design a template in the same way that you design a BIRT report. In fact, you can create a report design then save it as a template. The file-name extension for a template file is .rpttemplate. If you are creating templates for users worldwide, you can localize the text in the templates the same way you localize text in a BIRT report. The templates that ship with BIRT Studio are localized, and the text in each template appears in the language specified by the locale the user chooses when logging in to Actuate Information Console.

This section describes the key steps for creating a template but does not provide information about the report elements you can use in a template. For information about designing BIRT reports and templates, see the book that accompanies open source BIRT Report Designer, *BIRT: A Field Guide.*

How to create a report template

1 In the Report Design perspective, create a new template, using the following procedure:

   1 Choose File➛New➛Other.

   2 On New, expand Business Intelligence and Reporting Tools, then select Template. Choose Next.

3 In New Template—Template, select the folder in which to create the template file, specify a file name, then choose Next.

4 In New Template—Set Template Property:

1 In Display Name, specify a display name for the template. This name identifies the template when the template is displayed on the Report Template page in BIRT Studio.

2 In Description, provide a description of the template. This description appears as a Tooltip when the user hovers the mouse pointer over the template in BIRT Studio.

3 In Template Image, browse to the thumbnail image of the template. This step assumes that you have already created the image you want to use as the thumbnail image and placed it in Shared Resources.

4 Choose Finish. A blank report page appears in the layout editor.

2 Drag elements from the palette, and drop them in the layout editor.

3 For elements that you want BIRT Studio users to edit for their report designs, identify those elements as template report items. Only labels and tables can be edited in BIRT Studio:

1 Right-click the element, then choose Create Template Report Item.

2 Specify a descriptive name for the element, so the BIRT Studio user can easily identify the purpose of the element.

3 Provide instructions for using the element.

Figure 12-4 shows an example of an element name and instructions for using the element.

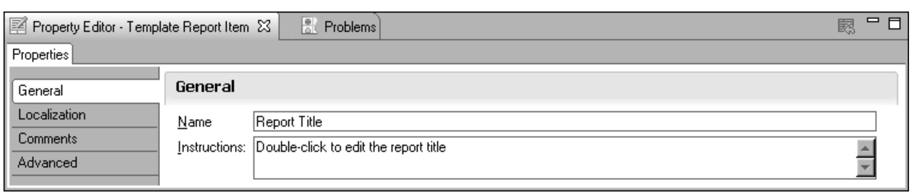


Figure 12-4 Specifying name and instructions for an editable element

4 For elements that you do not want the BIRT Studio user to edit, you should also specify a descriptive name, so that the BIRT Studio user can easily identify the purpose of the element and decide whether to include the element in the report. In Property Editor, type a name for the Name property. Figure 12-5 shows setting a label's name as Copyright.

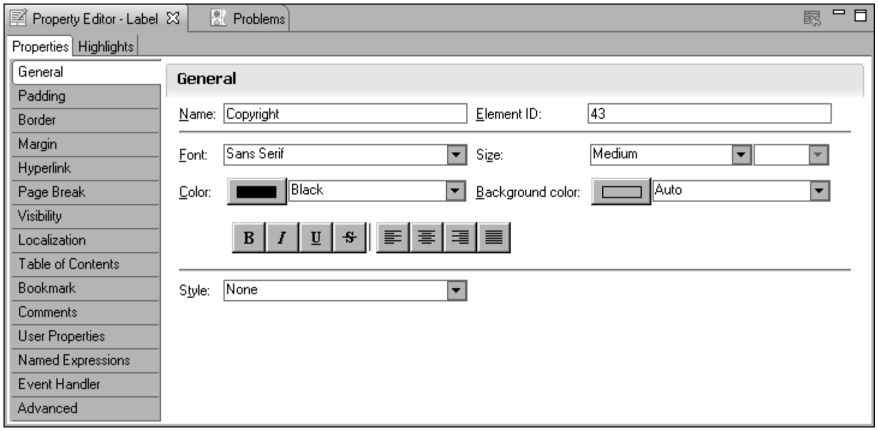


Figure 12-5 Specifying a name for a label element that users cannot edit

Providing data with a report template

If you are creating a template that you want the BIRT Studio user to use with an information object or a data object, add an editable table element to the template. When a template contains an editable table element, BIRT Studio prompts the user to select what type of data source they choose, data objects or information objects. Then the report wizard displays a list of the objects from the selected type, published on iServer and available to you. You can choose a data object or an information object to use in your report.

If you are creating a template that includes data, create a data source and data set that specifies the data that the BIRT Studio user can display in the report. You can define multiple data sources and multiple data sets in a template. When the user selects a template with multiple data sets, BIRT Studio prompts the user to select one of the data sets to use for the report. For information about defining data sources and data sets, see *BIRT: A Field Guide.* In addition to the standard data source types, described *BIRT: A Field Guide,* you can use information objects and data objects as data sources as well.

Using a CSV file as a data source

A BIRT Studio report design can use a comma-separated values (CSV) file as a data source if the CSV file is a predefined data set in a report template. To use the file as a data source, you must copy the CSV file to the appropriate directory. To determine which directory to use, download the .rptdesign file to a local directory as an XML file. In the XML code, locate the <data-sources> element, shown in the following example:

```
<data-sources>
<oda-data-source
extensionID="org.eclipse.datatools.connectivity.oda.flatfile"
name="Data Source" id="266">
<text-property name="displayName">Data Source
</text-property>
<property name="HOME">C:\</property>
<property name="CHARSET">UTF-8</property>
<property name="INCLTYPELINE">YES</property>
</oda-data-source>
</data-sources>
```

The HOME property shows the directory in which to place the CSV file.

Excluding a data set

You can exclude a data set in a template from the Select Data dialog in BIRT Studio. For example, you want to display stock quote data from a web service in the report, but you do not want the user to manipulate the data. To exclude a data set from the Select Data dialog, set the data set's UsageInBRS property to excluded in the template's XML representation. For example, the following code excludes the Orders data set:

```
<oda-data-set extensionID="org.eclipse.birt.report.data.oda.jdbc
   .JdbcSelectDataSet" name="Orders" id="8">
   <list-property name="userProperties">
      <structure>
         <property name="name">UsageInBRS</property>
         <property name="type">string</property>
         <property name="isVisible">true</property>
      </structure>
   </list-property>
   <property name="UsageInBRS">excluded</property>
```

Creating themes for a report template

As described earlier in this chapter, you create themes to provide the user with different sets of styles to apply to a report. You use Actuate BIRT Designer Professional to create the themes in a library. After you create the library, you associate the library with the template.

A library can contain any number of themes, and a theme can contain any number of styles. Actuate BIRT Designer Professional provides support for two types of styles. You are able to do the following:

- Create a custom style, and apply it to a report element. For example, you can create a style named Report Title, then apply the style to a label that displays the report title.
- Apply style properties to predefined style names, or selectors. These predefined style names correspond to the different types of report elements. For example, you can apply style properties to a predefined style named table-header, and all table headers in the report are formatted accordingly. This technique is similar to defining styles in CSS where you associate styles with HTML elements, such as <H1> or <P>.

How to create a theme

1 Create a library:

   1 Choose File➛New➛Other.

   2 On New, expand Business Intelligence and Reporting Tools, then select Library. Choose Next.

   3 In New Library, specify the folder in which to create the library, specify a file name, then choose Finish. If a message box appears, choose OK.

2 Choose the Outline view. The Outline view, shown in Figure 12-6, displays the types of report elements you can add to a library.

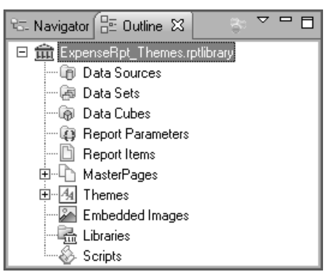


Figure 12-6 Outline view

3 In the Outline view, expand Themes.

4 Right-click defaultTheme, and choose Rename to change the name of the theme.

5 Right-click the theme, and choose New Style to create a style for the theme.

6 On New Style, select one of the following options:

   - To apply style properties to a specific type of report element, select Predefined Style, and select a style from the drop-down list.

- To create a user-named style, select Custom Style, and specify a unique descriptive name, such as Report Title or Copyright.

7 Set the desired properties for the style by selecting a property category on the left and specifying property values.

8 When you finish setting style properties, choose OK to save the style.

9 Repeat steps 5 to 8 to create additional styles for the theme.

10 To create a new theme, right-click Themes, and choose New Theme.

How to associate a library with a template

1 If the BIRT resource folder is not the current project folder, place the library in the BIRT resource folder, so that it is available to the template. To specify a folder as the resource folder:

   1 Choose Window➛Preferences.

   2 In Preferences, expand Report Design, then choose Resource, as shown in Figure 12-7.

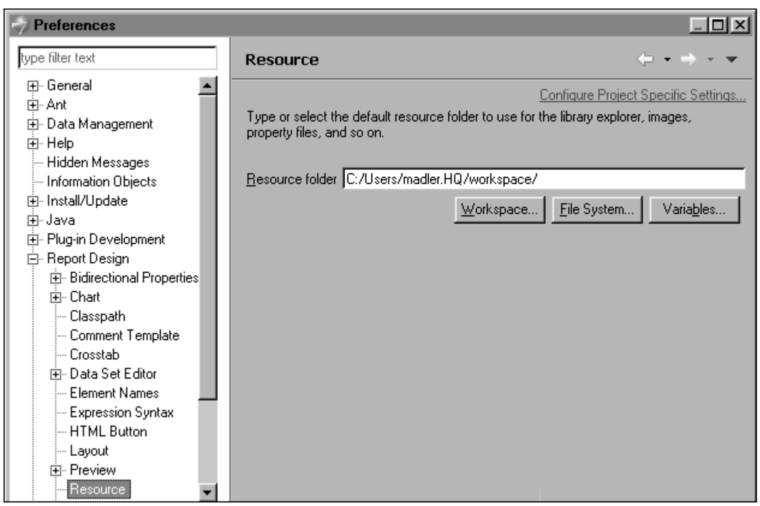


Figure 12-7 Specifying the location of the resource folder in Preferences

   3 Choose File System to select a folder to use as the resource folder.

   4 On Directory Selection, navigate to a folder on your computer or on the network, or choose Make New Folder to create a new folder.

   5 Choose OK to confirm your folder selection. Preferences displays the path to the resource folder.

   6 Choose OK to save the resource folder location information, and close the Preferences window.

7 In the layout editor or Navigator, select the library, then choose File➛Copy Library to Shared Resource Folder. Share Library displays the library name and the location of the resource folder.

8 Choose Finish to confirm placing a copy of the library in the resource folder.

2 Open the template file, and choose the Outline view.

3 In the Outline view, right-click Libraries, and choose Use Library, as shown in Figure 12-8.

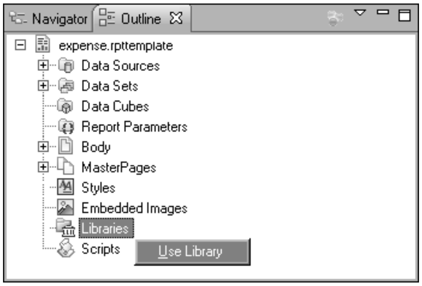


Figure 12-8 Choosing Use Library in the template file's outline view

4 In Use Library, expand Shared Resources to display the libraries in the BIRT resource folder. Figure 12-9 shows an example of Use Library.

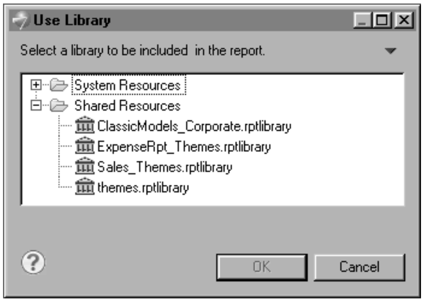


Figure 12-9 Displaying libraries in the resource folder

5 Select the library that contains the themes you want to use with the template, then choose OK.

Publishing a template

Templates must be published in specific locations. Otherwise, BIRT Studio cannot display the templates. BIRT Studio displays templates by categories. The default user interface has only one category named Standard. When you publish a template, you can create a new category or select an existing category in which to display your template.

If you create and publish a wide variety of templates, you can organize the templates into different categories. You can, for example, organize templates by report types (budget reports, expense reports, stock purchase plan reports) or by departments in your organization (Human resources, Sales, Customer Support).

Figure 12-10 shows an example of BIRT Studio customized to display three template categories by report types.

The Standard category appears at the top of the list because it is the default category supplied with BIRT Studio. All other categories that you create are listed in alphabetical order. You can designate a different template category as the default category. This task is described later in this section.

Templates must be published to the \<iServer resource folder>
\BizRDRptTemplates folder on the Actuate iServer Encyclopedia volume. The general steps are:

1 Use the default resource folder, named Resources, at the root level of the Encyclopedia volume. Alternatively, you can create a different resource folder on the Encyclopedia volume using Management Console.

2 Publish the template, using Actuate BIRT Designer Professional. The first time you publish a template, the BizRDRptTemplates folder is created within the Encyclopedia volume's resource folder. Do not change the name of the BizRDRptTemplates folder. BIRT Studio looks for templates in this folder, and the folder name is not configurable.

Each of these steps is described in detail in this section.

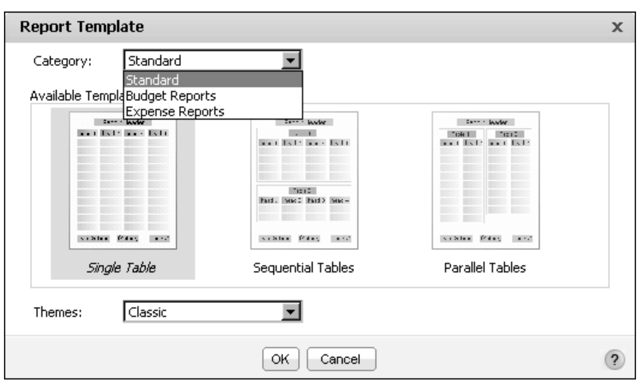


Figure 12-10 Displaying three template categories on Report Template

How to create a resource folder on an Encyclopedia volume

1 Log into Management Console as Administrator.

2 Create a new folder that you can designate as the resource folder.

3 Set the Encyclopedia volume's resource folder property:

1. Choose Volume.
2. On Volume, choose Properties.
3. On the General Properties page, in Resource folder, specify a resource folder name. The default resource folder name, Resources, appears in Figure 12-11.

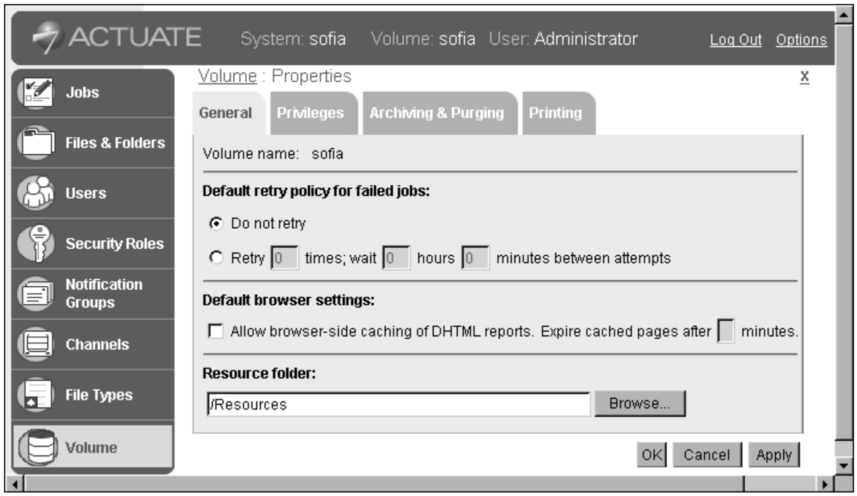


Figure 12-11 Specifying the Encyclopedia volume's resource folder

How to publish a report template

1 In Actuate BIRT Designer Professional, in Navigator, right-click the template file, then choose Publish Templates to iServer. Publish Templates appears with the template file selected, as shown in Figure 12-12.

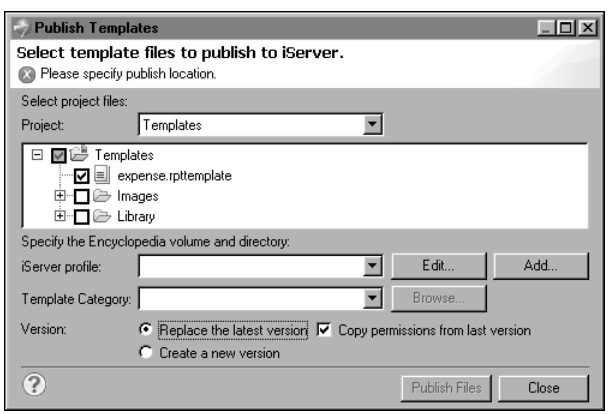


Figure 12-12 Publish Templates

2 On Publish Templates, select an iServer profile from the drop-down list. An iServer profile specifies the connection properties to connect to a specific Encyclopedia volume. If there is no appropriate profile, create a new profile:

   1 Choose Add.

   2 Specify the connection information in New iServer Profile:

      1 In Profile name, type a unique name that identifies the new profile.

      2 In iServer, type the name or IP address of the computer on which Actuate iServer is installed.

      3 In Port number, type the number of the port to access Actuate iServer.

      4 In Volume, select the Actuate iServer Encyclopedia volume.

      5 In User name, type the user name for accessing the volume.

      6 In Password, type the password for accessing the volume.

   3 Choose Finish to save the iServer profile.

3 On Publish Templates, in Template Category, select an existing category from the drop-down list in which to publish the template. Alternatively, create a new category by choosing Browse, then specifying the name of the new category.

4 In Version, select the desired versioning option. To copy the privileges from the last published version of the template, select Copy permissions from last version.

5 Choose Publish Files, then choose Close after the file is published.

The first time you publish a template to an Encyclopedia volume, you must grant users access to the appropriate template folders and files. For more information about assigning privileges for folders and files on an Actuate iServer Encyclopedia volume, see *Managing an Encyclopedia Volume.*

Setting the default template category

By default, the Report Template dialog displays the templates in the predefined category, Standard, as shown earlier in Figure 12-10. To view and select a template in another category, the user has to select a different category from the Category drop-down list. Typically, you want to make your custom templates more visible. You can configure BIRT Studio so that the Report Template dialog displays a different category of templates by default.

How to set the default template category

1 Open web.xml for editing. This file is in:

```
<context root>\WEB-INF
```

2 Change the value of the DEFAULT_REPORT_TEMPLATE_CATEGORY _NAME parameter from Standard to the name of the category whose templates you want the Report Template dialog to display by default. The following example shows the Sales category set as the default template category:

```
<param-name>DEFAULT_REPORT_TEMPLATE_CATEGORY_NAME
   </param-name>
<param-value>Sales</param-value>
```

3 Restart the appropriate Windows service for the change to take effect.

Publishing resources associated with a template

Typically, each template uses the following external resources that you must also publish to specific locations:

- An image file that provides the user with a thumbnail image of the template
- A library file that contains the themes the user can select to apply to the report

If a template contains localized text and you have created resource files that translate text into different languages, you must also publish these resource (.properties) files.

You publish resources used by a template to the resource folder on the Actuate iServer Encyclopedia volume. The procedure for creating and designating an Encyclopedia volume resource folder is described earlier in this section.

How to publish resources associated with a template

1 Using the file system, put the files in the BIRT resource folder (not to be confused with the Actuate iServer Encyclopedia volume resource folder). The location of the BIRT resource folder is specified in the Preferences page, which you access by choosing Window➛Preferences, then choosing Report Design➛Resource.

2 Choose File➛Publish Resources to iServer. Publish Resources displays the files in the BIRT resource folder.

3 Select the files to publish.

4 In iServer profile, select the profile that specifies the connection properties to the Encyclopedia volume where you want to publish the files. Publish location displays the name of the Actuate iServer Encyclopedia volume’s resource

folder. All BIRT resource files are published to the volume's resource folder. You cannot change the location.

Figure 12-13 shows an example of selections made in Publish Resources.

5 Choose Publish Files, then choose Close after the files are published.

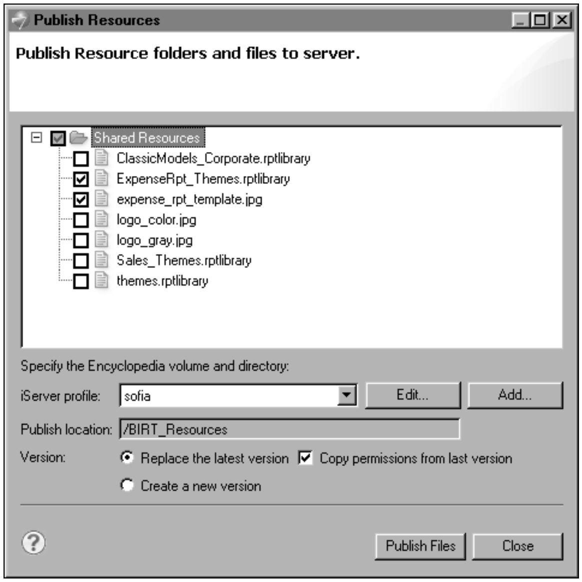


Figure 12-13 Selecting files for publishing to a specified iServer volume

Chapter

13

Providing data

This chapter covers the following topics:

- Types of data sources
- Using data objects as a data source
- Creating information objects
- Creating a custom data source

Types of data sources

To create a new report, the first step the BIRT Studio user performs is to select a template on which to base the report. The second step is selecting the data to display. You provide BIRT Studio users access to data in any of the following ways:

- Create and publish data objects.
- Create and publish information objects.
- Create and publish report templates that include defined data sets. Use this method if you want to provide templates that combine both presentation and data elements.
- Create a custom data source and integrate it with BIRT Studio. Use this method if the data is stored in a proprietary system or if you want to design a custom user interface for selecting data.

This chapter describes all of these methods except creating and publishing report templates that include defined data sets. This method is described in Chapter 12, "Creating and publishing report templates."

Using data objects as a data source

A data object can include any number of data sources, data sets, data cubes, and report parameters. The data objects can be used as predefined data sources in BIRT Studio. Business users can select already published data objects as data sources, or templates that are designed to use data objects as data sources.

The data objects created for business reports should be designed with the business report requirements in mind. They should provide the data in a structure that is appropriate for business users and for the report elements that users can add to a report.

When you design data objects for business reports always consider the following:

- Report parameters in a data object do not link to parameters created in BIRT Studio, so you typically do not include report parameters in a data object that you create for BIRT Studio users.
- In BIRT Studio, a chart uses data from a table in a report. The chart does not use data directly from a data set or a cube.
- BIRT Studio cross tab reports require specialized cross-tab templates and the data sources used in the templates must be data objects.

Deploying data objects

There are two types of data objects, and either one can be a data source:

- Data object design

  A data object design specifies a data source and a data set and data cubes. A report developer or data modeler creates the data object design in BIRT Designer Professional. A data object design has a .datadesign extension.

- Data object store

  A data object store contains the materialized data generated by a data object design. A volume administrator creates the data object store in Management Console. A data object store has a .data extension.

If the user uses a data object design as a data source, the report retrieves data from databases and other data sources when it runs. The advantage of this approach is that the report retrieves current data. The disadvantage is that it increases the load on production databases.

If the user uses a data object store as a data source, the report retrieves materialized data from the data object store when it runs. The advantage of this approach is that it does not increase the load on production databases. The disadvantage is that the report may not retrieve current data.

A volume administrator controls which users have access to data objects by assigning privileges on the .datadesign and .data files. If the volume administrator does not want to increase the load on production databases, they can give users access to .data files but not .datadesign files.

To deploy a data object, a report developer, data modeler, and volume administrator cooperate in performing the following tasks:

- A report developer or data modeler creates a data object design in BIRT Designer Professional. For more information about creating a data object design, see *Using Actuate BIRT Designer Professional*.
- The report developer or data modeler places the .datadesign file in the BIRT Resource folder.
- The report developer or data modeler publishes the .datadesign file to the iServer /Resources folder.
- A volume administrator schedules a job for the .datadesign file that generates a .data file. For more information about scheduling a job, see *Managing an Encyclopedia Volume*.
- A volume administrator assigns privileges for the appropriate users on the .datadesign and .data files. To give a user access to a .datadesign file, assign read and execute privileges. To give a user access to a .data file, assign read privilege.

Deploying cross tab templates

When a user creates a cross tab report, they must use a template that contains a cross tab element. The cross tab template that installs with iServer resides in $AC_SERVER_HOME\resources\BizRDRptTemplates\Standard \template4.rpttemplate. A template developer can modify this template or create their own.

Creating information objects

An information object is a business-friendly view of data designed for use with report design applications such as BIRT Studio. Information objects, which can extract and integrate data from a variety of sources, including relational databases, stored procedures, web services, and XML documents, enable BIRT Studio users to focus on the data to present in a report, rather than focusing on how to obtain the data. You create information objects using the IO Design perspective in Actuate BIRT Designer Professional.

About design considerations

Design information objects for the business user, not for other data experts. Take into account the experience levels of the users. The point of creating information objects is to shield the user from the complexities of data structures and relationships. Before you begin creating information objects, understand the types of reports users want to create, the data they need, and the different ways users can manipulate data in BIRT Studio. The rest of this section provides general guidelines for designing usable information objects.

Organizing data in a suite of information objects

An information object can contain any amount of data and can integrate data from any number of data sources. It is possible to create an information object that returns hundreds of columns of data from multiple database tables and even from disparate data sources. While it may be tempting to create a single information object that contains all the data that BIRT Studio users want, an information object that contains too much data is not usable. Once you determine the data the BIRT Studio users need, evaluate how best to organize the data into information objects, and how much data to include in each information object.

For usability reasons, it is good practice to limit the number of columns per information object. One way to do so is to include only related data in each information object. For example, create one information object that contains customer data and another information object that contains orders data. On the other hand, if a novice user wants to create a report that displays order totals for customers, and the relevant data is stored in an orders table and a customers table, you can create an information object that joins the data from the two tables and calculates the totals.

A single report can display data from multiple information objects, so you do not need to create a single information object that provides all the data that a particular report requires.

BIRT Studio provides two options for using data from multiple information objects:

- The user can join data in information objects through a common column. To support this operation, the advanced data-manipulation option must be enabled for the user. For more information about enabling this option see "Configuring advanced data operations" in Chapter 13, "Providing data."
- The user can use a report template that includes more than one table, and associate a different information object with each table. This strategy requires coordination of the template design with the information object design.

Consider also that BIRT Studio provides the user with easy ways to select, sort, group, filter, and aggregate data. You can create an information object that provides more data than any one report needs and let the user process the data further.

Planning folder names and structure

When the user selects a report template with a table element that contains no data, BIRT Studio prompts the user to select an information object from the Actuate iServer Encyclopedia volume. Plan the folder hierarchy and names that BIRT Studio displays to the user, so that the user can find and identify the information objects easily.

When you publish an information object project to the Actuate iServer Encyclopedia volume, the folder hierarchy and names that BIRT Studio displays are the same as the project folder hierarchy and names you use in the IO Design perspective. With this in mind, you should create a logical hierarchy of projects and information objects, and use descriptive names.

Figure 13-1 shows an example of a project in the Navigator view in the IO Design perspective. In the example, a single project named BIRT Studio Data contains two information objects that retrieve data from two different data sources.

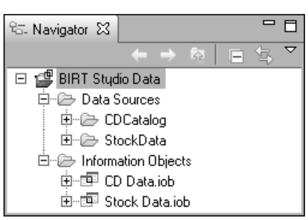


Figure 13-1 Project folder hierarchy in the Navigator view

Figure 13-2 shows the folder hierarchy that the BIRT Studio user traverses, by default, to select an information object for the report, assuming that the user can access the information object project folders.

In addition to planning the folder names and hierarchy of an information object project, you can simplify user access to information objects. For more information about this strategy see “Simplifying access to information objects,” later in this chapter.

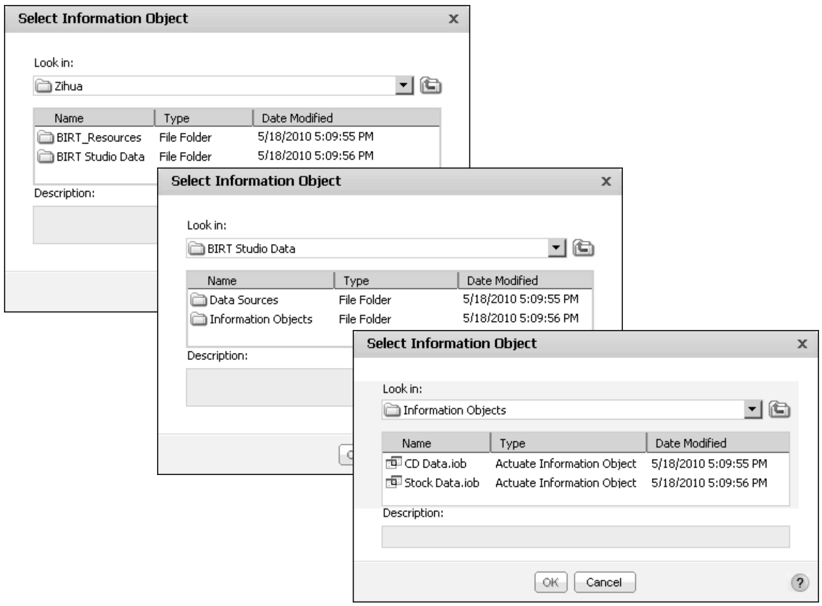


Figure 13-2 Information objects folder hierarchy in BIRT Studio

Specifying the listing order for information object columns

By default, the Available Data pane in BIRT Studio displays the columns in an information object in alphabetical order, rather than in the order in which they appear in the IO Design perspective. You can specify a different order using the DATAFIELDS_DISPLAY_ORDER parameter in web.xml. web.xml resides in the following location:

```
<context root>\WEB-INF
```

For example, if you want information object columns to appear in the same order in which they appear in the IO Design perspective, uncomment the DATAFIELDS_DISPLAY_ORDER parameter, and set it to none:

```
<context-param>
   <param-name>DATAFIELDS_DISPLAY_ORDER</param-name>
   <param-value>none</param-value>
</context-param>
```

You can also set DATAFIELDS_DISPLAY_ORDER to ascending or descending. The default value is ascending.

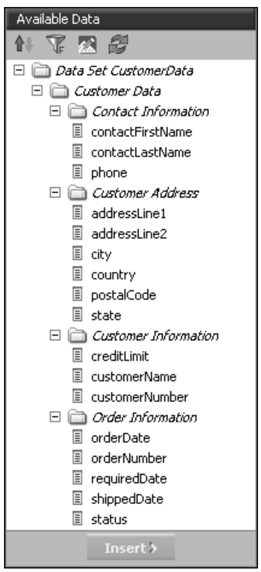
Restart the appropriate Windows service for the change to take effect.

Organizing columns into categories

If an information object contains a large number of columns, you can group the columns into logical categories. For example, organize Address, City, State, Postalcode, and Country columns under a category named Customer address. Without this categorization, the address-related columns are scattered throughout the list of columns.

Figure 13-3 shows a comparison of columns that are organized in categories with columns that are not categorized, as they appear in BIRT Studio.

Columns organized in categories



Uncategorized columns

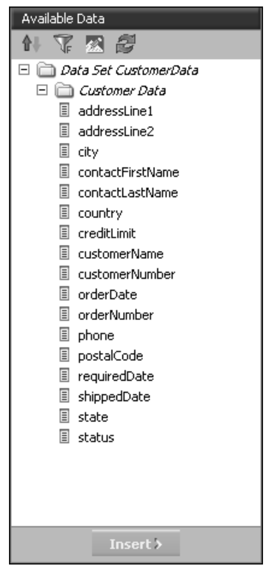


Figure 13-3 Comparing categorized and uncategorized data, as it appears in BIRT Studio

Creating information objects with the IO Design perspective

Before you can actually create an information object, you must perform a series of tasks to connect to a data source and prepare the data for use in the information object. This section describes generally all the key tasks involved in creating an information object and other required objects. For detailed information about each task, see *Designing BIRT Information Objects.*

How to create an information object

1 Create an information object project. An information object project organizes all the files required for the creation and use of information objects. You can create one project for all the information objects and their related objects, or you can organize information objects in multiple projects.

2 Create a data connection definition. A data connection definition specifies the properties to connect to a data source, as shown in Figure 13-4. The properties include the data source type, user name, and password. A data connection definition file name has a .dcd extension.

3 Create a map. A map, as shown in Figure 13-5, represents the data in a discrete entity, for example, a single database table, a result set that a stored procedure returns, or an XML file or flat file. A map file name has a .sma extension.

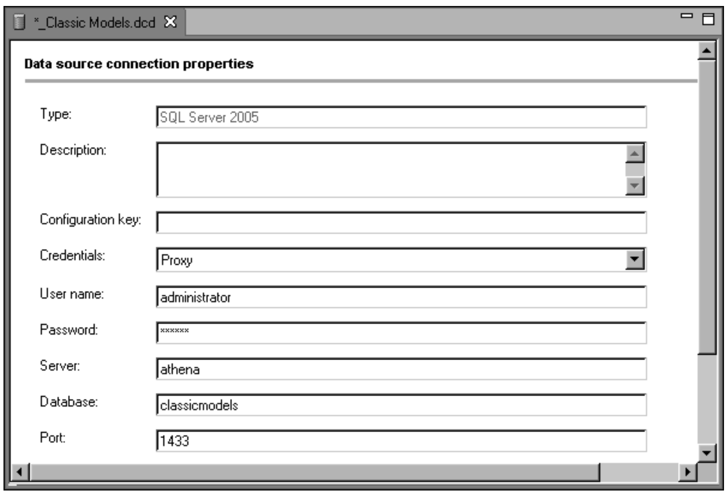


Figure 13-4 Defining a data connection

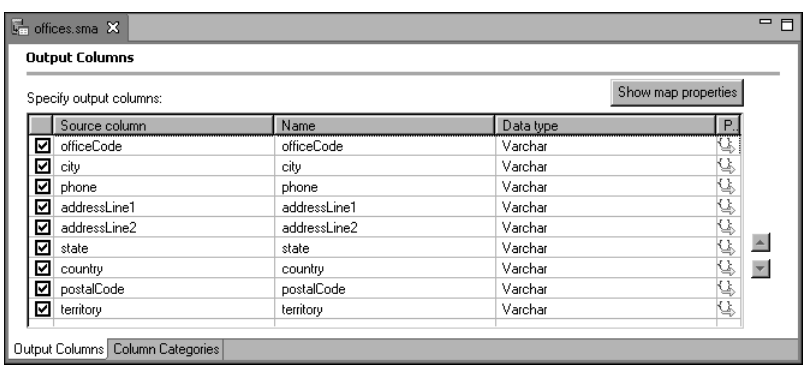


Figure 13-5 Creating a map by selecting an offices table

4 Create an information object using data from one or more maps. You also can build an information object from other information objects. Figure 13-6 shows an information object that uses data from two maps. An information object file name has an .iob extension.

The tabs at the bottom of Figure 13-6 represent the different ways you can manipulate the data in an information object. For example, to organize columns in categories, choose Column Categories. To limit the data that the information object returns, choose Filters.

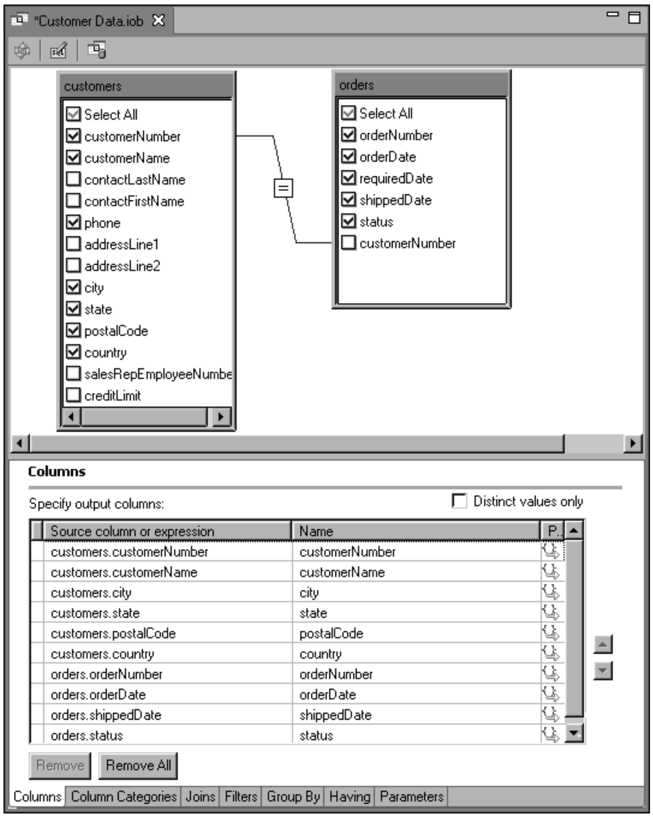


Figure 13-6 Defining an information object

Publishing information objects

When you develop an information object project, the files and folders are saved in the same folder structure on the development volume as they are in the workspace. The iServer Explorer view displays the folder structure on the development volume. It is not necessary to publish the project to the development volume.

For users to use an information object, however, you must publish all the files in the project to the production volume. If you publish only the information object (.iob) file, the BIRT Studio user can see and select the information object, but without the data connection definition and map files, the information object returns no data.

You can publish the project files in any folder in an Encyclopedia volume on which you have write privilege. Within the volume folder in which you publish the project, the files are stored in the same project folder hierarchy as shown in the Navigator view. The top-level folder is the project folder. The information object (.iob) files are saved in a subfolder named Information Objects. Data connection definition (.dcd) files and map (.sma) files that information objects use, are saved in subfolders within a subfolder named Data Sources.

To publish a project, in the Navigator view, select the project, then choose File➛Publish Information Objects. When you choose this option, a Publish Information Objects dialog box, shown in Figure 13-7, prompts you to select the project, a folder, or individual files to publish, and the Encyclopedia volume folder in which to save the files.

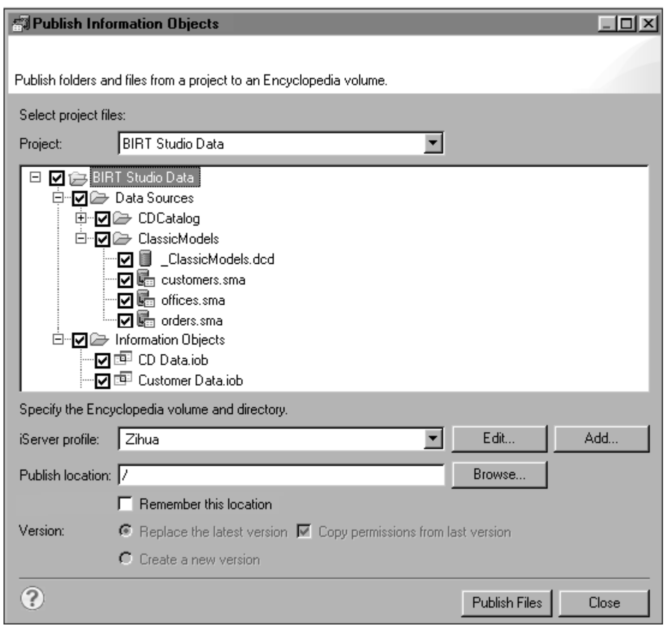


Figure 13-7 Selecting files to publish

The first time you publish a project to an Encyclopedia volume, you must grant users access to the appropriate folders and files. After that, the privileges are copied from the previous version. For more information about assigning privileges for folders and files on an Actuate iServer Encyclopedia volume, see *Managing an Encyclopedia Volume.*

Simplifying access to information objects

As discussed previously, to select an information object for a report, the user traverses the information object project folder hierarchy within the parent folder in which you publish the project. At a minimum, assuming you published the information object project at the root level of the volume and granted the user access to those folders, the user navigates two folder levels to access the information objects. If the organization of a volume's contents requires that you publish an information object project within other folders—which is typical—the user must navigate more than two levels of folders. If each level displays multiple folders, the probability of the user finding the correct path to the information objects the first time decreases.

To address this navigation problem, you can:

- Configure BIRT Studio so that the Select Information Object dialog box initially shows the Information Objects folder where the information objects reside.
- Hide the folders that are not required.

You can use both techniques, each described in the following sections.

Configuring the starting folder for finding information objects

By default, the Select Information Object dialog box displays one of these folders as the starting folder from which the user navigates to find information objects:

- The user's home folder, if the volume administrator created a home folder for the user.
- The root folder in the volume, if the user does not have a home folder. Figure 13-2 shows an example of the folder hierarchy that the user traverses from the root folder to find information objects published to the volume.

You can simplify user navigation by setting the starting folder to the Information Objects folder where the information objects reside, as Figure 13-8 shows. Using this setting, the user sees all the information objects immediately.

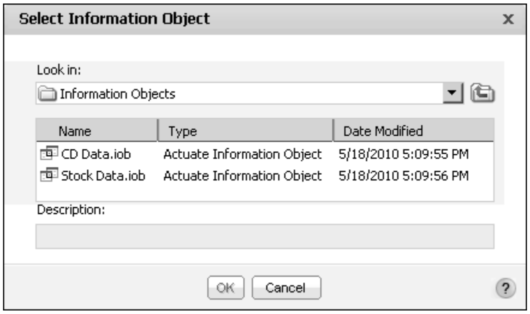


Figure 13-8 Information Objects folder

You specify the Information Objects folder as the starting folder if all the information objects available to the user are stored in one folder. If, however, you organized your information objects in multiple projects, each project is a folder on the volume, and each project folder contains its own Information Objects folder. In this case, you typically set the starting folder at a higher level, so that the user first selects the desired project folder, then the Information Objects folder within the project folder. Figure 13-9 shows an example of starting navigation from a folder that contains multiple information object project folders.

If your installation of Actuate iServer supports multiple volumes, you can set a different starting folder for each volume. Setting a starting folder does not mean that the Select Information Object dialog box displays only the specified folder and its contents. The user can navigate up to other folders for which she has been granted access. In the example Figure 13-9, the user can navigate to folders above the BIRT Studio Data folder.

To set the starting folder from which the user navigates to find information objects, you edit erni_config.xml. erni_config.xml resides in the following location:

```
<context root>\WEB-INF
```

In erni_config.xml, the <iserverconfigs> element defines the Actuate iServer volumes for which you configure the starting folders. Each <iserverconfig> element under <iserverconfigs> defines the iServer name, iServer port, volume name, and starting folder name.

Listing 13-1 shows the two sample definitions of <iserverconfig>, which are commented out in erni_config.xml.

Listing 13-1 Sample <iserverconfig> elements in erni_config.xml

```
<iserverconfigs>
     <defaultIOStartingDir>/design</defaultIOStartingDir>
     <iserverconfig>
        <server>speedy</server>
        <port>8000</port>
        <volume>speedy</volume>
        <ioStartingDir>/AQ</ioStartingDir>
     </iserverconfig>
     <iserverconfig>
        <server>ddykm7b1</server>
        <port>8000</port>
        <volume>ddykm7b1</volume>
        <ioStartingDir>/design</ioStartingDir>
     </iserverconfig>
</iserverconfigs>
```

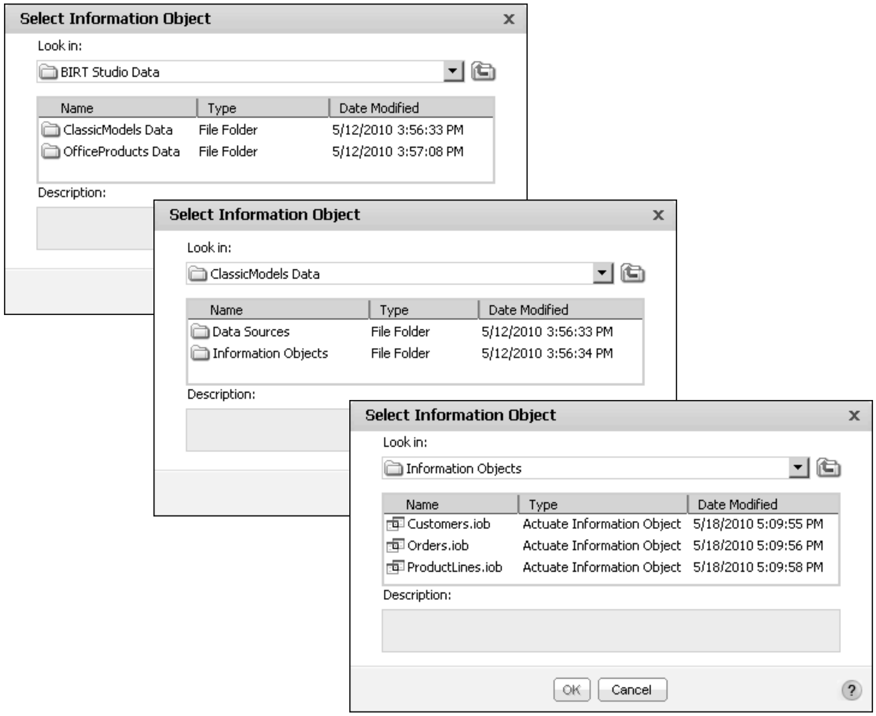


Figure 13-9 Navigating folders in BIRT Studio

If you are working with a server that supports multiple volumes, and you want to set a different starting folder for each volume, define an <iserverconfig> entry for each volume. If you configure only one volume, or if you want to use the same starting folder for multiple volumes, specify the path to the starting folder in the <defaultIOStartingDir> element.

Be careful when editing this section of erni_config.xml. Comment out <iserverconfig> elements that you do not use. Read the attribute descriptions in erni_config.xml to get information about valid values to supply for each element.

Listing 13-2 shows an example of the <iserverconfigs> section, edited to set the starting folder to the /ClassicModels Data/Information Objects folder. The changes are shown in bold. Notice that the sample <iserverconfig> elements are commented out. You edit the <iserverconfigs> section as the example shows for a single-volume server or if you want all volumes to use the same starting folder.

Listing 13-2 Setting the <defaultIOStartingDir> element to a folder

```
<iserverconfigs>
    <defaultIOStartingDir>/ClassicModels Data/Information
  Objects</defaultIOStartingDir>
```

```
<!--
     <iserverconfig>
        <server>speedy</server>
        <port>8000</port>
        <volume>speedy</volume>
        <ioStartingDir>/AQ</ioStartingDir>
     </iserverconfig>
     <iserverconfig>
        <server>ddykm7b1</server>
        <port>8000</port>
        <volume>ddykm7b1</volume>
        <ioStartingDir>/design</ioStartingDir>
     </iserverconfig>
-->
</iserverconfigs>
```

How to configure the starting folder for finding information objects

1 Open erni_config.xml for editing. This file is in the following location:

```
<context root>\WEB-INF
```

2 Find the <iserverconfigs> section at the end of the file. This entire section is commented out.

3 Move the closing comment marker, -->, which appears after </iserverconfigs> to the line after the documentation about the feature. This edit keeps the documentation as a comment, and uncomments the <iserverconfigs> element.

4 Edit the <iserverconfigs> section to specify the starting folder for a single volume or for multiple volumes.

5 Restart the appropriate Windows service for the changes to take effect.

Hiding folders

All the files in an information object project are required for a report to use data in an information object, so you must publish all the files to an Actuate iServer Encyclopedia volume. The user, however, only needs to select the information object. You can simplify the user's folder navigation by hiding the Data Sources folder and its contents and displaying only the Information Objects folder and the contents of that folder.

You accomplish this task by using Management Console to assign the appropriate folder and file privileges to the user, using these general steps:

1 Create a security role, for example, BRS User.

2 Create a user profile for each BIRT Studio user, and assign the BRS User role to the users.

3 Select each folder, and assign the appropriate privileges to the BRS User role. By default, users do not have access to folders or files created by someone else. When you publish files, you must explicitly assign privileges to roles or users.

- To enable the user to see the information objects, select the top-level project folder and the Information Objects subfolder, select the BRS User role, then assign the Visible privilege.
- To enable the user to execute an information object, assign the Trusted Execute privilege to the .iob file. The Trusted Execute privilege permits the user to execute an information object without having privileges on the information object's underlying data sources.

For more information about creating roles and users, and assigning privileges for folders and files in an Actuate iServer Encyclopedia volume, see *Managing an Encyclopedia Volume.*

Creating a custom data source

Another way to provide users access to data is to create a custom user interface to a data source and integrate the user interface with BIRT Studio. BIRT Studio supports the usage of BIRT and custom data sources. There are three supported scenarios you can implement:

- You use a BIRT JDBC data source, and the out-of-box functionality to configure the user interface for selecting the data.
- You use a BIRT driver to connect to your data source. You create your own user interface for selecting the data. This scenario is appropriate in all cases when your data source is not a BIRT JDBC data source, but it is still supported by BIRT, such as XML, Web Services, etc.
- You use your own custom ODA driver to connect to your data source and you build your own user interface for selecting the data.

Actuate provides two examples that reference the supported scenarios:

- Sample ODA data source. The example demonstrates how to configure a user interface for selecting data when using the BIRT JDBC driver.
- Extended sample ODA data source. The example creates a custom user interface for selecting data and uses a BIRT driver to connect to the data source.

Sample ODA data source

Creating a custom user interface to connect to a BIRT JDBC data source does not require additional coding. You must configure the data source and data set and describe the data set columns in the erni_config.xml file as shown in Listing 13-3.

Listing 13-3 Configuring the sample data source in erni_config.xml

```
<odaconfig>
   <name>OdaSample</name>
   <displayName>Sample ODA data source</displayName>
   <description>A sample ODA data source</description>

   <datasourceExtensionId>org.eclipse.birt.report.data.oda.jdbc
   </datasourceExtensionId>
   <datasourceDisplayName>CLASSICMODELS</datasourceDisplayName>
         <datasetExtensionId>org.eclipse.birt.report.data.oda.jdbc.
   JdbcSelectDataSet</datasetExtensionId>
   <datasetDisplayName>CLASSICMODELS.PAYMENTS</datasetDisplayName>
         <odaDriverClass>org.eclipse.birt.report.data.oda.sampledb.
   Driver</odaDriverClass>
   <odaURL>jdbc:classicmodels:sampledb</odaURL>
   <odaUser>ClassicModels</odaUser>
   <!-- <odaPassword></odaPassword> -->
<!-- Data Type can be DECIMAL(3)/INTEGER(4)/FLOAT(6)
/DATE-TIME(93)/DATE(91)/TIME(92)/BOOLEAN(16)/STRING(12)-->
   <odaColumns>
      <odaColumn>
         <name>CUSTOMERNUMBER</name>
         <dataType>INTEGER</dataType>
      </odaColumn>
      <odaColumn>
         <name>PAYMENTDATE</name>
         <dataType>DATE</dataType>
      </odaColumn>
      <odaColumn>
         <name>CHECKNUMBER</name>
         <dataType>STRING</dataType>
      </odaColumn>
      <odaColumn>
         <name>AMOUNT</name>
         <dataType>FLOAT</dataType>
      </odaColumn>
   </odaColumns>
   <dataObject>CLASSICMODELS.PAYMENTS</dataObject>
   <!-- <queryText></queryText> -->
   <enabledInWorkgroupMode>false</enabledInWorkgroupMode>
   <enabledInEnterpriseMode>true</enabledInEnterpriseMode>
   <entryPoint></entryPoint>
</odaconfig>
```

To test the Sample ODA data source with BIRT Studio, you must first enable the data source, named OdaSample, by setting the data source's

<enabledInEnterpriseMode> attribute in erni_config.xml to true. The erni_config.xml file is stored in the following location:

```
<context root>\WEB-INF
```

Typically, in standard installations the <context root> is as follows:

In iServer:

```
<ACTUATE_HOME>\iServer\servletcontainer\iportal
```

In Information Console:

```
<ACTUATE_HOME>\iPortal\iportal
```

After you enable the sample data source, restart the appropriate Windows service for iServer or Information Console, and open BIRT Studio. When prompted, select one of the standard report templates. The Data Source dialog box, shown in Figure 13-10, prompts you to select a data source. Select Sample ODA data source.

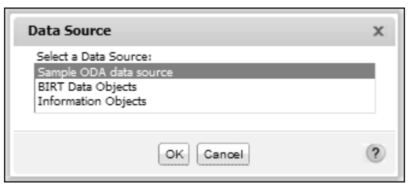


Figure 13-10 Selecting data source

Based on the XML definition in erni_config.xml, the product displays a sample ODA Editor with a table, containing the configured database columns, as shown in Figure 13-11.

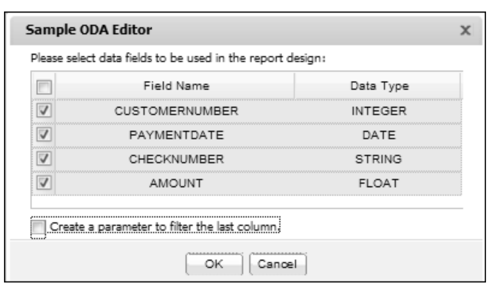


Figure 13-11 Selecting data in sample ODA editor

The check boxes appearing next to the column names allow the user to select the data for the report. The editor gives the user ability to create a parameter for filtering the data in the last column.

Configuring a sample ODA data source

Table 13-1 lists all the configuration attributes and their descriptions.

Table 13-1 Attributes of a sample ODA data source

Attribute	Description
<name>	The unique name of the data source. This name follows certain naming conventions. There can be no spaces, for example.
<displayName>	The data source name that appears in the Data Source dialog box, shown in Figure 13-10.
<description>	The data source description that appears in the Available Data pane.
<datasetExtensionId>	Identifies the ID of the data set extension of the data driver.
<datasourceDisplayName>	The display name of the data source.
<datasourceExtensionId>	Identifies the ID of the data source extension of the data driver.
<datasetDisplayName>	The display name of the data set.
<odaURL>	The database URL to use to connect to the data source.
<odaUser>	The user name, used to connect to the database.
<odaPassword>	The database password.
<odaColumn><name>	The name of the database column to be included in the user interface.
<odaColumn><dataType>	The type of the database column. Can be one of the following: `DECIMAL(3), INTEGER(4), FLOAT(6), DATE-TIME(93),DATE(91), TIME(92), BOOLEAN(16), STRING(12)`
<dataObject>	The name of the database table if single table is used.
<queryText>	The text of the query if the columns are from different tables.
<enabledInWorkgroupMode>	Not used.
<enabledInEnterpriseMode>	Indicates whether this data source is available to users.

Table 13-1 Attributes of a sample ODA data source

Attribute	Description
<entryPoint>	A URL or servlet that points to the first web page of your custom query builder.

Configuring the ODA data source requires knowledge about your JDBC data source and data set. If you do not know this information, you can obtain the database connection properties from your database administrator. It is a good practice to create first a sample report in BIRT Designer Professional and validate the connection and the database query. You use the XML source of the created report to identify the correct values for the sample ODA configuration. Look for the attribute values in the <data-sources> and <data-sets> tags at the beginning of the report XML. Listing 13-4 shows portions of the report XML that contain the configuration attributes and their values.

Listing 13-4 Report XML Source

```
<data-sources>
   <oda-data-source
   extensionID="org.eclipse.birt.report.data.oda.jdbc" name="Data
   Source" id="7">
   …
   <property
   name="odaDriverClass">org.eclipse.birt.report.data.oda.sampledb
   .Driver</property>
   <property name="odaURL">jdbc:classicmodels:sampledb</property>
   <property name="odaUser">ClassicModels</property>
   </oda-data-source>
</data-sources>
<data-sets>
         <oda-data-set extensionID=
   "org.eclipse.birt.report.data.oda.jdbc.JdbcSelectDataSet"
   name="Data Set" id="8">
   …
   <xml-property name="queryText"><![CDATA[select *
   from CLASSICMODELS.PAYMENTS]]></xml-property>
```

Extended sample ODA example

To implement an extended custom data source, you perform the following programming and configuration tasks:

- Develop an Open Data Access (ODA) data driver to connect to and retrieve data from a data source.
- Develop a query builder that prompts the user to select data, processes the user's selection, and creates the query to get the data.

- Configure the custom data source for use with BIRT Studio.

An example of an extended ODA data source installs with BIRT Studio. The data source accesses data from an integrated sample database named ClassicModels. To test this custom data source with BIRT Studio, you must first enable the data source, named OdaSampleExt, by setting the data source's <enabledInEnterpriseMode> attribute in erni_config.xml to true. The erni_config.xml file is stored in the following locations:

In iServer:

```
<ACTUATE_HOME>\iServer\servletcontainer\iportal\WEB-INF
```

In Information Console:

```
<ACTUATE_HOME>\iPortal\iportal\WEB-INF
```

Listing 13-5 shows in bold the <enabledInEnterpriseMode> attribute whose value you change from false to true. Look for the ODA data source named OdaSampleExt.

Listing 13-5 Configuring the extended sample data source in erni_config.xml

```
<odaconfig>
   <name>OdaSampleExt</name>
   <displayName>Extended sample ODA data source</displayName>
   <description>An extended sample ODA data source</description>
   <datasourceExtensionId>org.eclipse.birt.report.data.oda.jdbc</
   datasourceExtensionId>

   <datasetExtensionId>org.eclipse.birt.report.data.oda.jdbc.JdbcS
   electDataSet</datasetExtensionId>
   <enabledInWorkgroupMode>true</enabledInWorkgroupMode>
   <enabledInEnterpriseMode>true</enabledInEnterpriseMode>
   <entryPoint>/OdaSample</entryPoint>
</odaconfig>
```

After you enable the extended sample data source, restart the appropriate Windows service, and open BIRT Studio. When prompted, select one of the standard report templates. The Data Source dialog box, shown in Figure 13-12, prompts you to select a data source. Select Extended sample ODA data source.

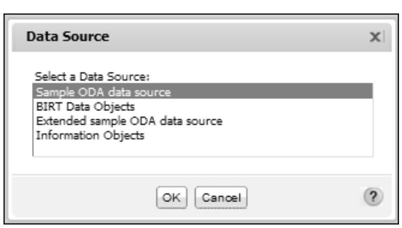


Figure 13-12 Selecting a data source

A query builder page, shown in Figure 13-13, prompts you to select the data to use in the report.

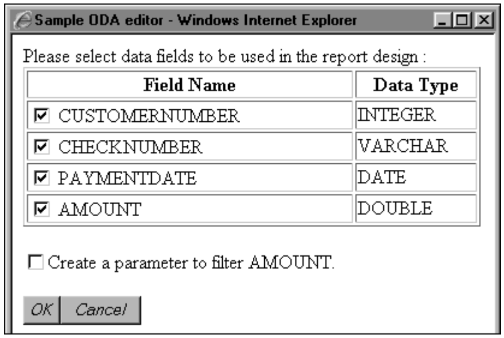


Figure 13-13 Building a query

After you select the data and choose OK, the query builder closes, and the data you selected appears under Available Data, as shown in Figure 13-14.

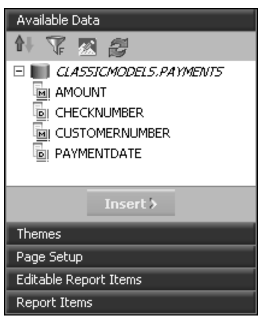


Figure 13-14 Selected data appears under Available Data

Developing an Open Data Access (ODA) data driver

You develop an ODA data driver by implementing run-time interfaces that the ODA framework defines. The ODA framework is an Eclipse Data Tools Platform (DTP) project component that provides a way for an application to access data from standard and proprietary data sources. For information about ODA and developing data drivers, go to the following site:

```
http://www.eclipse.org/datatools/project_connectivity
   /connectivity_doc/OdaOverview.htm
```

Another resource is the Addison-Wesley's *Integrating and Extending BIRT* book. Actuate also provides an example about how to develop a flat file ODA driver. You can find the source code for this example in:

```
<ACTUATE_HOME>\oda\examples\FlatFileExample
```

The extended data source example uses a predefined ODA JDBC driver to connect to the ClassicModels sample database. The source code for this driver (org.eclipse.birt.report.data.oda.jdbc) is available in the Eclipse DTP CVS repository.

After you develop a custom ODA driver, you must copy the driver to the following folders, depending on the deployment environment:

In iServer:

```
<ACTUATE_HOME>\iServer\MyClasses\eclipse\plugins
```

In Information Console:

```
<ACTUATE_HOME>\MyClasses\eclipse\plugins
```

Developing a custom query builder

You develop a query builder using standard web application components, JavaServer Pages (JSPs) and servlets. After creating the necessary components, you perform a series of tasks to integrate the components with Information Console. This section describes the sample query builder, on which you can base your own query builder. This section also describes the procedures for compiling and deploying your own query builder components.

The sample query builder consists of the following components:

- ClassicModelQueryBuilder.jsp. This JSP displays the page in Figure 13-13. It also contains JavaScript functions to build the string that contains information about the data that the user selected and to send the request to a servlet. To view the code, see ClassicModelQueryBuilder.jsp in the following location:

  ```
  <context root>\bizRD\oda\sample
  ```

- SampleServlet.java. This servlet communicates with ClassicModelQueryBuilder.jsp, creates a design session, and creates the query using information from the JSP. To view the code, see SampleServlet.java in the following location:

  ```
  <context root>\iportal\examples\oda\classes\com
     \actuate\erni\oda\ClassicModels
  ```

Creating the servlet

A servlet performs the main tasks for getting and managing data for BIRT Studio users, and it is the key piece of any custom query builder. The example servlet, SampleServlet.java, extends the HTTPServlet class, and performs the following tasks:

- Manages design sessions for concurrent BIRT Studio users
- Uses ODA API to define the basic methods for constructing a query, result set, and column objects

- Processes the information sent by ClassicModelQueryBuilder.jsp, and creates a query
- Returns dynamic content that appears in Available Data, as shown in Figure 13-14

The servlet contains declarations of the ODA data driver's data source and data set extensions:

```
private final static String Datasource_Extension_Id =
   "org.eclipse.birt.report.data.oda.jdbc";

private final static String Dataset_Extension_Id =
   "org.eclipse.birt.report.data.oda.jdbc.JdbcSelectDataSet";
```

These extension IDs are used to construct the definitions of the data source connection and data set query in the report design.

The servlet supports the GET method only. Its doGet method retrieves request parameters, creates a unique ODA design session, and stores a session ID in the ODA session, so that other requests in the same session can access the same ODA session object. In any given session, the servlet can receive multiple requests with different parameter values. Table 13-2 describes the supported parameters.

Table 13-2 Session request parameters

Parameter	Description
inedit	Indicates whether the user has started editing the values on the query builder page. Values are null or true.
state	Shows the editing status. Values are null, ok, or cancel.
selection	Contains information about the selected data fields.
sessionId	Contains the session ID.

The servlet executes a different action, depending on the parameter values. Table 13-3 describes the actions taken with the different inedit and state values.

Table 13-3 Actions corresponding to inedit and state values

inedit value	state value	Action
null	any	Generates a new session ID. Stores the ID and the callback URL in the session map. Changes the inedit parameter to true and sends a response to the calling page.
not null	null	Redirects the response to ClassicModelQueryBuilder.jsp.
not null	ok	The user has finished selecting data fields.

(continues)

Table 13-3 Actions corresponding to inedit and state values (continued)

inedit value	state value	Action
not null *(continues)*	ok	The servlet deletes the session ID, and parses the value in the selection parameter to build the query. The response is redirected to a BIRT Studio page.
not null	cancel	The user cancelled out of the query builder page. The servlet deletes the session ID.

Compiling the servlet

After you develop your servlet, you must compile the class. You can use a javac compiler from the command prompt or any Java IDE, such as Eclipse. To compile a servlet class, the following JAR files must be in your Java classpath:

- com.actuate.iportal.jar
- org.eclipse.emf.common.jar
- org.eclipse.emf.ecore.jar
- org.eclipse.datatools.connectivity.oda.design.jar

  These files are in the following location:

  ```
  <context root>\WEB-INF\lib
  ```

- servlet.jar

  You can find this file in different places, depending on the Actuate products installed on your computer. For example, servlet.jar can be found in the following location:

  ```
  <ACTUATE_HOME>\iServer\servletcontainer\webapps\acrsse\WEB_INF\
    lib
  ```

Deploying the servlet

After you compile the servlet class, deploy the servlet to your application. You can deploy your servlet as a class file, or packaged as a JAR file. The SampleServlet.class servlet is deployed to your application packaged in iportal.jar.

If you deploy the servlet using a JAR file, copy the JAR file to the following location:

```
<context root>\WEB-INF\lib
```

If you deploy the servlet as a class file, copy the servlet class to the following location:

```
<context root>\WEB-INF\classes
```

Registering the servlet

After you compile your servlet, you also need to register the servlet with the web application. To register the servlet, you add two entries to web.xml, which is stored in the following location:

```
<context root>\WEB-INF
```

The first entry, under the <servlet> element, defines a name for the servlet and specifies the compiled class that executes the servlet. The following example shows the <servlet> entry for the sample servlet:

```
<servlet>
  <servlet-name>OdaSampleServlet</servlet-name>
  <servlet-class>
   com.actuate.erni.oda.ClassicModels.SampleServlet
   </servlet-class>
</servlet>
```

The second entry, under the <servlet-mapping> element, defines the URL pattern that calls this servlet. The following example shows the <servlet-mapping> entry for the sample servlet:

```
<servlet-mapping>
  <servlet-name>OdaSampleServlet</servlet-name>
  <url-pattern>/OdaSample</url-pattern>
</servlet-mapping>
```

Configuring the extended custom data source

After you finish developing all the components of a custom data source, you configure the data source for use with BIRT Studio. Data sources available to BIRT Studio are defined in the following configuration file:

```
<context root>\WEB-INF\erni_config.xml
```

Listing 13-5 shows the definition of the extended sample data source. To add your custom data source, create a new <odaconfig> element in erni_config.xml. Table 13-4 describes each attribute, within the <odaconfig> element, that you configure.

Table 13-4 Attributes of an extended custom data source

Attribute	Description
<name>	The unique name of the data source. This name follows certain naming conventions. There can be no spaces, for example.
<displayName>	The data source name that appears in the Data Source dialog box, shown in Figure 13-10.

(continues)

Table 13-4 Attributes of an extended custom data source (continued)

Attribute	Description
<description>	The data source description that appears in the user interface.
<datasetExtensionId>	Identifies the ID of the data-set extension that was created in the ODA data driver. The value should match the data-set extension ID that is specified in the servlet.
<datasourceExtensionId>	Identifies the ID of the data-source extension that was created in the ODA data driver. The value should match the data-source extension ID that is specified in the servlet.
<enabledInWorkgroupMode>	Not used.
<enabledInEnterpriseMode>	Indicates whether this data source is available to users.
<entryPoint>	A URL or servlet that points to the first web page of your custom query builder.

Chapter

14

Configuring BIRT Studio

This chapter covers the following topics:

- Assigning license options to a user
- Enabling or disabling functionality
- Configuring the application environment

Assigning license options to a user

If you have named user licensing, Actuate iServer requires the assignment of license options to each user to support the user accessing, running, and viewing BIRT reports in an Encyclopedia volume. Use Management Console to create a user and a home folder, and assign license options in an Encyclopedia volume. A user uses the home folder to access, run, and view BIRT reports using a web browser.

How to create a user and a home folder, and assign license options

In this task, create a new user, and specify a home folder for the user, using Management Console.

1 Depending on your operating system, complete one of the following tasks:

- For Windows XP, choose Start➛All Programs➛Actuate 11➛iServer Management Console.
- For Windows Server 2003, choose Start➛Programs➛Actuate 11➛iServer Management Console.
- For a non-Windows operating system, start Management Console by opening a browser window and typing the following URL in the address bar:

```
http://localhost:8900/acadmin/login.jsp
```

The Management Console login page appears, as shown in Figure 14-1.

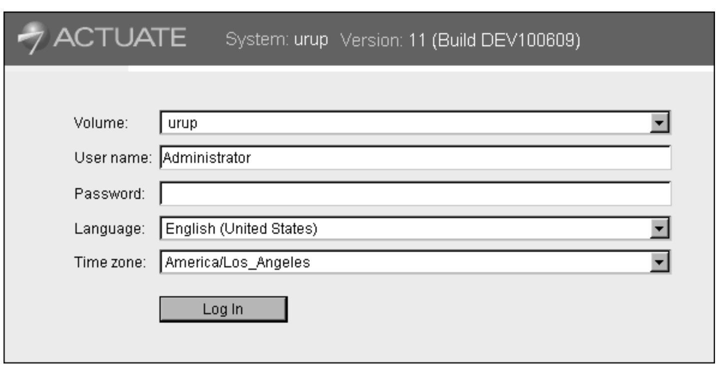


Figure 14-1 Logging in to Management Console

2 In User name type Administrator. Choose Log In. Management Console—Files and Folders appears, as shown in Figure 14-2.

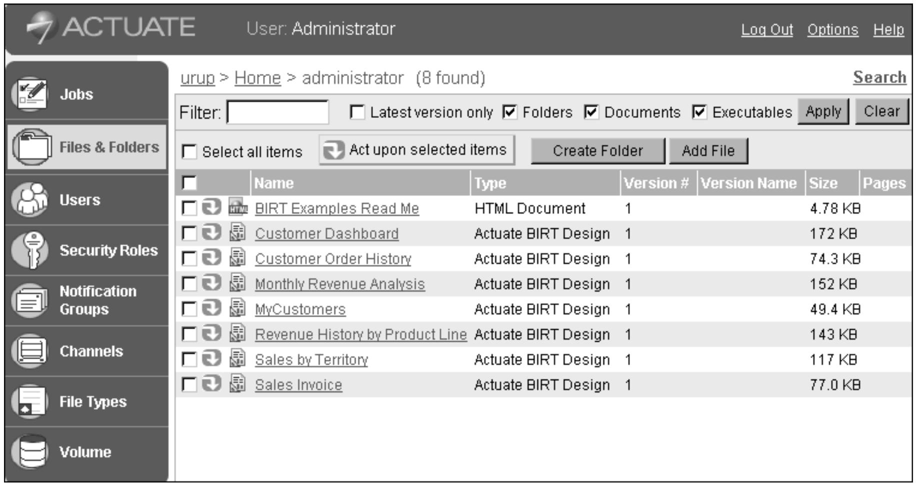


Figure 14-2 Files and Folders

3 In the side menu, choose Users. Management Console—Users appears.

4 To create a new user, complete the following tasks:

1 Choose Create User, as shown in Figure 14-3.

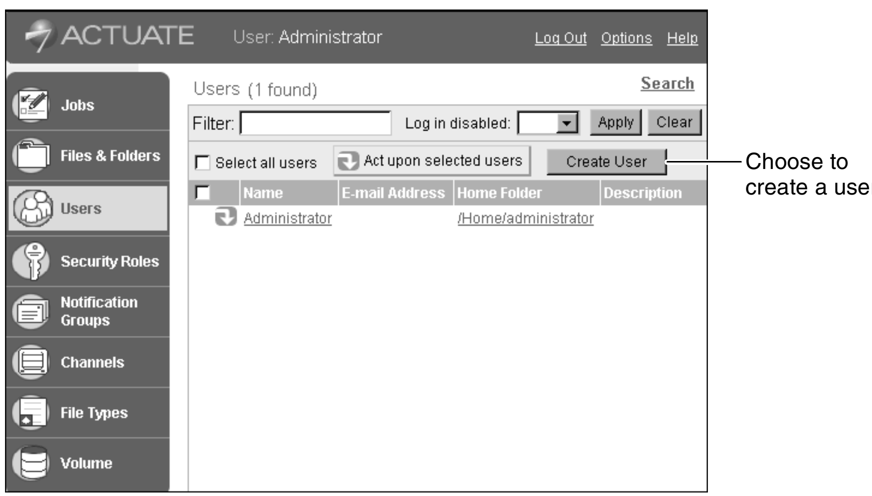


Figure 14-3 Management Console—Users

2 In New User—General, as shown in Figure 14-4, in Name, type the new user's name, for example:

```
pcastillo
```

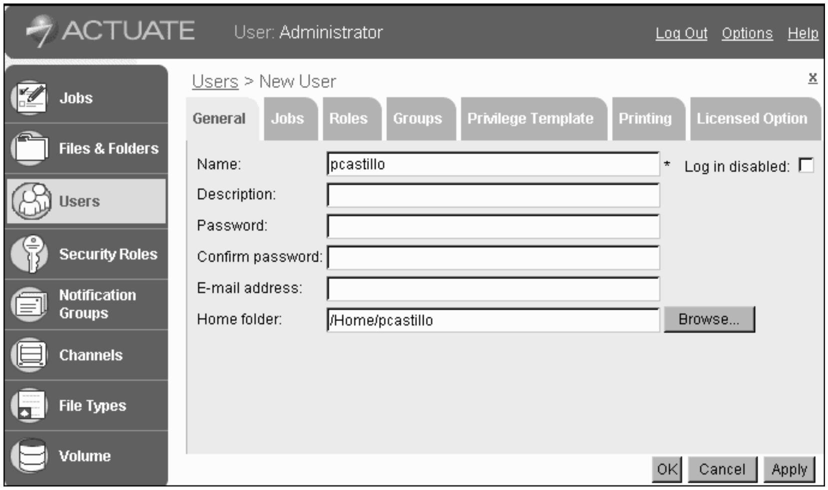


Figure 14-4 Creating a new user

3 Type a password.

4 In Home folder, type the path to the home folder, for example:

```
/Home/pcastillo
```

5 To provide the new user access to the Actuate products, choose Licensed Option.

6 In New User—Licensed Option, in Available, press Ctrl and select:

- BIRT Studio Option
- BIRT Option

Choose the right arrow. The options appear in Selected, as shown in Figure 14-5.

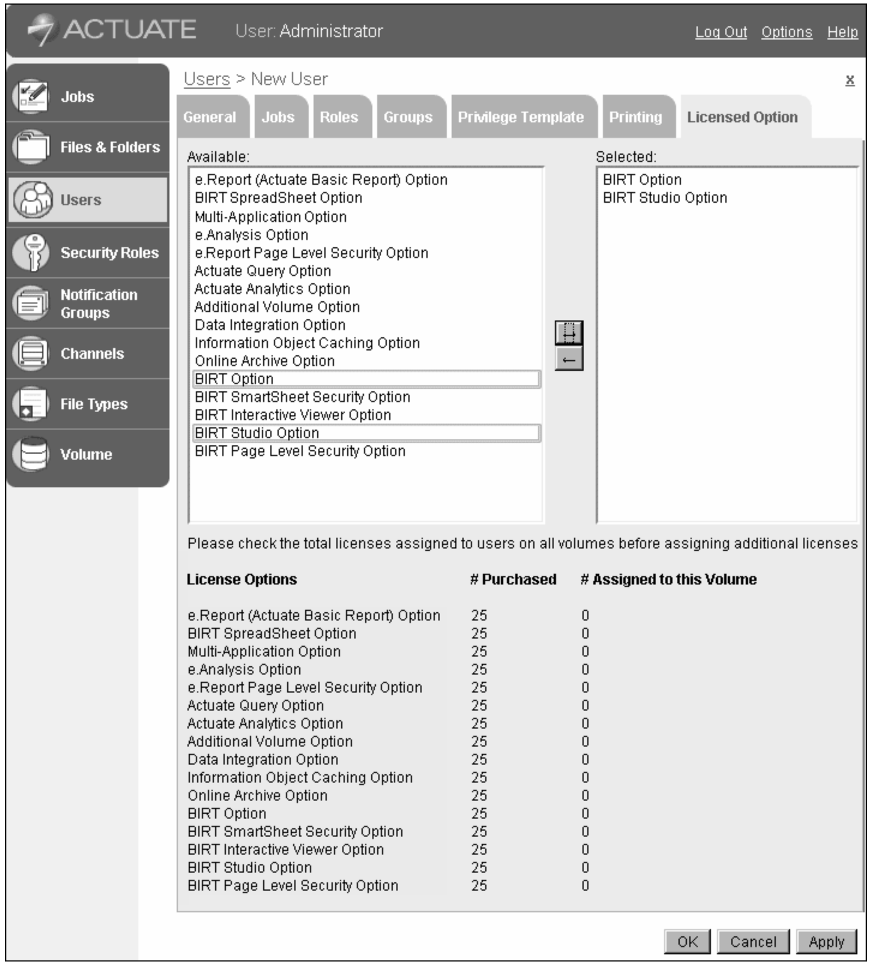


Figure 14-5 Selecting licensed options

Choose OK. Management Console—Users displays the new user, and the path to the user’s home folder, as shown in Figure 14-6.

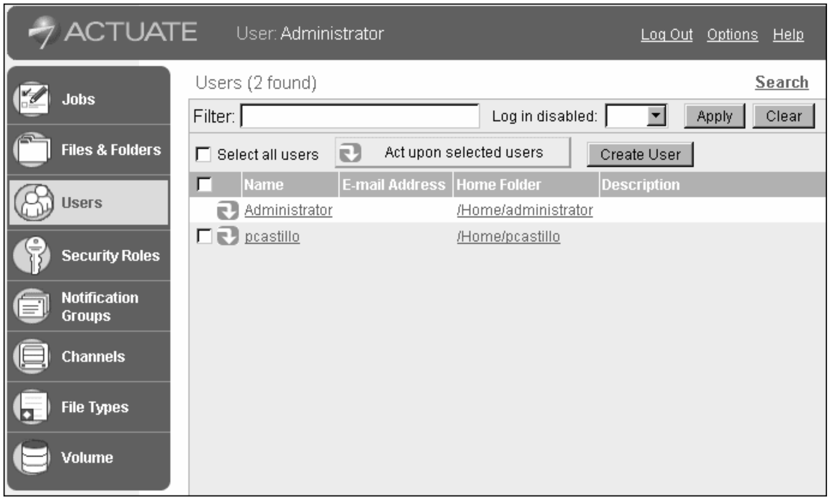


Figure 14-6 Management Console—Users, displaying the new user

5 To log out of Management Console, choose Log out.

Enabling or disabling functionality

BIRT Studio provides a full range of tools to support the report design process. Administrators can limit BIRT Studio functionality by user roles. For example, if the BIRT Studio users you support have little experience designing reports, you can simplify the design process by disabling more advanced functionality, such as creating calculated columns, aggregating data, and joining multiple information objects. Alternatively, if you want users to format report content only by selecting a corporate-designed theme, you can disable the formatting functionality.

Configuring toolbar and context menu items

You configure the toolbar and context menu functionality that is available to users by editing attributes in the BIRT Studio configuration file, erni_config.xml. This file is located in:

```
<context root>\WEB-INF
```

Customizations you make to erni_config.xml apply at the application level. If you want different sets of functionality available to different groups of users, you need to create multiple instances of the web application, then customize the functionality of each BIRT Studio instance.

In erni_config.xml, the <actionSets> element defines all the user actions that can be enabled or disabled. The actions are organized by category, for example, file operations, calculations, and formatting. The <actionSet> element defines the category, and the <action> element defines a specific action. Listing 14-1 shows the hierarchy of elements.

Listing 14-1 An example of an <actionSet> element in erni_config.xml

```
<actionSets>
   <actionSet>
      <name>FileOperations</name>
      <visible>true</visible>
      <action>
         <name>New</name>
         <enabled>true</enabled>
      </action>
      <action>
         <name>Open</name>
         <enabled>true</enabled>
      </action>
      <action>
         <name>Save</name>
         <enabled>true</enabled>
      </action>
      <action>
         <name>SaveAs</name>
         <enabled>true</enabled>
      </action>
   </actionSet>
```

By default, all actions are enabled. You can disable actions in the following ways:

- To disable a particular action, change the action's <enabled> attribute from true to false.
- To disable all actions within a category in one step, change the action set's <visible> attribute from true to false.

For the changes to take effect, restart the appropriate Windows service. When you relaunch BIRT Studio, the toolbar displays different buttons and the context menus display different items, depending on which action or actions you disabled.

Configurable actions

Table 14-1 lists toolbar and context menu actions that you can enable or disable. Some of the actions appear in both the toolbar and context menus, and some appear in context menus only. While you can disable any of the actions defined in erni_config.xml, it does not make sense to disable all the actions. For example, disabling both the New and Open actions under file operations prevents a user from creating or opening reports.

Table 14-1 User actions that you can enable or disable through erni_config.xml

Action Set	Action	Description
Calculations	Aggregation	Performs a calculation over a specified set of data rows.
	Calculation	Creates a calculated column, based on a specified expression.
	ChangeSubtotal	Changes the subtotal function, applied to a column in a summary table.
	Chart	Inserts a chart.
	Data Fields	Shows the data fields in the report, and supports adding or deleting fields in the report.
	EditCalculation	Changes a calculated column.
	Filter	Filters table rows, based on a specified condition.
ColumnHeader Operations	DeleteRow	Deletes the row of the selected column header.
	InsertRow	Inserts a row above or below the selected column header.
	Merge	Merges the selected column header with the header on the right, left, or above.
	Split	Splits the selected, merged columns.
ColumnOperations	ColumnWidth	Changes the width of the selected column.
	HideColumn	Hides the selected column.
	MergeColumns	Merges the selected columns.
	MoveToDetail	Moves the selected item in the group header row to the table's detail row.
	MoveToGroup	Moves the selected item in the table's detail row to the group header row.
	NoRepeat	If duplicate data values appear in the selected column, displays only the first instance.

Table 14-1 User actions that you can enable or disable through erni_config.xml (continued)

Action Set	Action	Description
ColumnOperations *(continued)*	ReorderColumns	Changes the order of the columns in the table.
	RepeatValues	Displays duplicate data values in the selected column.
	ShowColumns	Shows the selected columns.
DeleteColumn	DeleteColumn	Deletes a column from a table in the report.
EditText	EditText	Enables editing of the selected static text.
FileOperations	New	Creates a new report design file.
	Open	Opens an existing report design.
	Save	Saves the current report design.
	SaveAs	Saves the current report design file under a different name or in a new location.
Formatting	AlignCenter	Centers the text in the selected column.
	AlignLeft	Aligns the left sides of text in the selected column.
	AlignRight	Aligns the right sides of text in the selected column.
	Border	Draws a border around the selected column.
	ConditionalFormat	Formats data in a selected column, based on a specified condition.
	Data	Formats the display of data in the selected column.
	Font	Formats the font of data in the selected column.
GeneralOperations	Parameter	Displays the parameters, if any, for the current report.
	SwitchSummary Mode	Switches between summary table mode and detail table mode for the selected table.

(continues)

Table 14-1 User actions that you can enable or disable through erni_config.xml (continued)

Action Set	Action	Description
Grouping	CreateSection	Adds a report section, which provides an additional level of data grouping.
	DeleteSection	Removes the selected report section.
	GroupBy	Groups table rows by values in the selected column.
	HideDetail	Hides the detail rows in a report section.
	PageBreak	Adds page breaks before or after a report section.
	ShowDetail	Shows the detail rows in a report section.
	UngroupBy	Removes groups in the selected column.
Help	Help	Shows help information.
PageLayout	PageLayoutIn Toolbar	Displays page layout toggle under toolbar. Disabled by default.
Preview	PreviewHTML	Shows a preview of the report in HTML format.
SectionOperations	SectionHeading	Shows the data fields in the report, and supports adding fields to the selected section heading.
Sorting	AdvancedSort	Sorts the table rows by the values of multiple columns.
	SortAscending	Sorts, in ascending order, the table rows by the values of the selected column.
	SortDescending	Sorts, in descending order, the table rows by the values of the selected column.
TemplateTable Operations	AutoSummarize On	If set to true, creates a summary table by default. If set to false, creates a detail table by default.
UndoRedo	Redo	Redo the last action.
	Undo	Undo the last action.

Configuration examples

This section provides examples of editing attributes in erni_config.xml, and the resulting changes to the BIRT Studio page. Figure 14-7 shows the default BIRT Studio page with all actions enabled. The formatting actions on the toolbar and context menu are called out, so that you can see the difference in the toolbar and context menu when these actions are disabled.

Listing 14-2 shows a change to the Formatting action set. Its <visible> attribute, shown in bold, is set to false. Note, however, that all the actions under the Formatting action set are still enabled.

Listing 14-2 Visibility of the Formatting action set, changed to false

```
<actionSet>
   <name>Formatting</name>
   <visible>false</visible>
   <action>
      <name>AlignLeft</name>
      <enabled>true</enabled>
   </action>
   <action>
      <name>AlignCenter</name>
      <enabled>true</enabled>
   </action>
   <action>
      <name>AlignRight</name>
      <enabled>true</enabled>
   </action>
   <action>
      <name>Font</name>
      <enabled>true</enabled>
   </action>
   <action>
      <name>Border</name>
      <enabled>true</enabled>
   </action>
   <action>
      <name>ConditionalFormat</name>
      <enabled>true</enabled>
   </action>
   <action>
      <name>Data</name>
      <enabled>true</enabled>
   </action>
</actionSet>
```

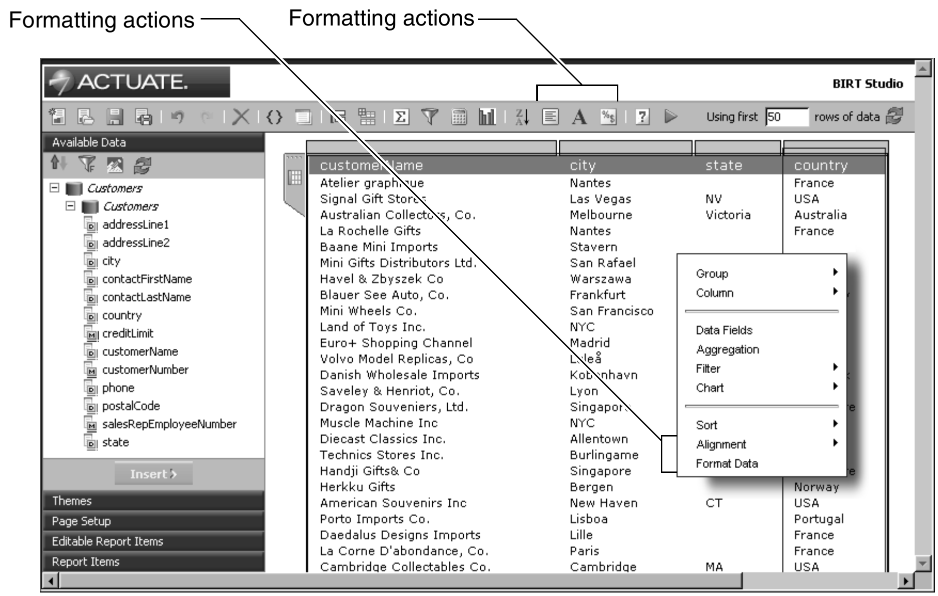


Figure 14-7 Default BIRT Studio page

Figure 14-8 shows the updated BIRT Studio page. None of the formatting actions appear in the toolbar or the context menu. Setting the <visible> attribute of an action set to false disables all actions within the action set.

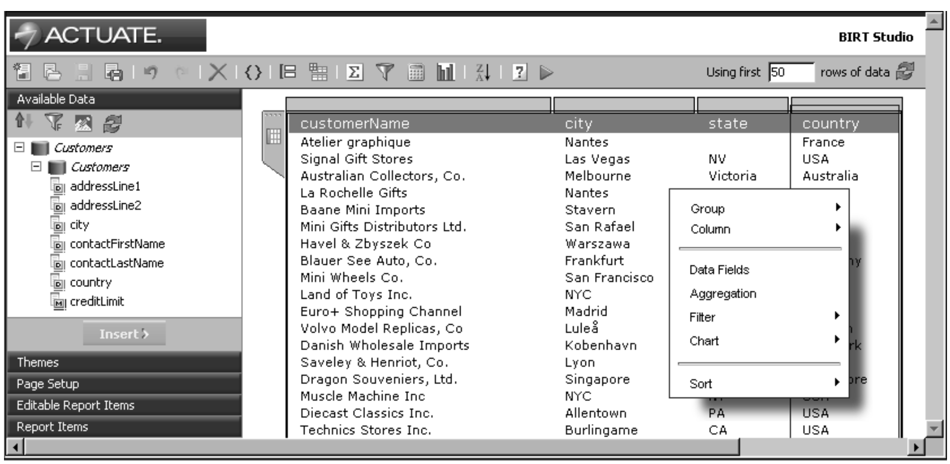


Figure 14-8 Updated toolbar and context menu, without any formatting functions

Listing 14-3 shows changes to the Font and Data actions within the Formatting action set. The Formatting action set's <visible> attribute is set to true. The Font and Data actions are disabled. The other actions in the action set are still enabled.

Listing 14-3 Font and Data (formatting) actions disabled

```
<actionSet>
  <name>Formatting</name>
  <visible>true</visible>
  <action>
    <name>AlignLeft</name>
    <enabled>true</enabled>
  </action>
  <action>
    <name>AlignCenter</name>
    <enabled>true</enabled>
  </action>
  <action>
    <name>AlignRight</name>
    <enabled>true</enabled>
  </action>
  <action>
    <name>Font</name>
    <enabled>false</enabled>
  </action>
  <action>
    <name>Border</name>
    <enabled>true</enabled>
  </action>
  <action>
    <name>ConditionalFormat</name>
    <enabled>true</enabled>
  </action>
  <action>
    <name>Data</name>
    <enabled>false</enabled>
  </action>
</actionSet>
```

Figure 14-9 shows the updated BIRT Studio page. The alignment actions are available on the toolbar and on the context menu, but not the font and data formatting actions.

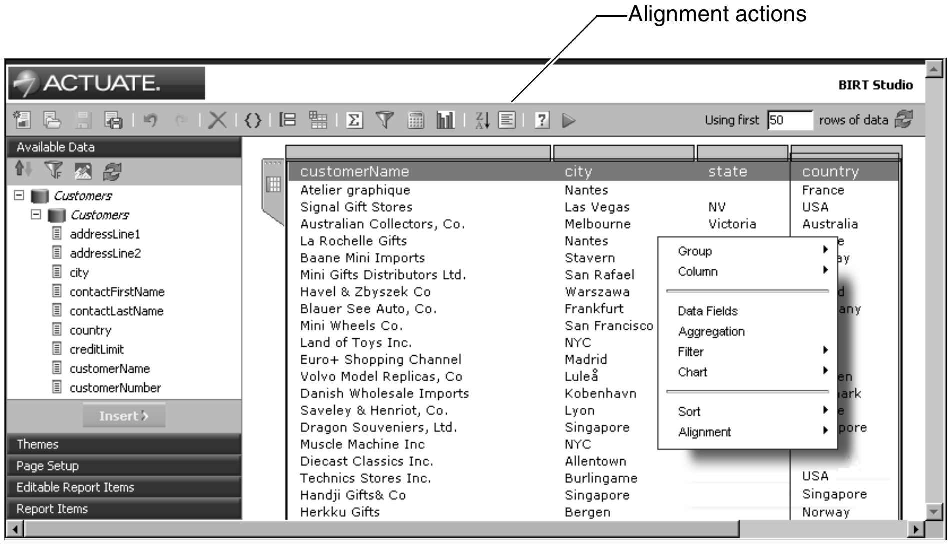


Figure 14-9 Updated toolbar and context menu, without the font and data formatting actions

Specifying the default position of aggregate values

In Release 10 and earlier, aggregate values were placed in a group's footer by default. In Release 11 and later, aggregate values are placed in a group's header by default. To revert to the behavior prior to Release 11, set EnableNewAggregationStyle to false in erni_config.xml:

```
<featureConfigs>
   <featureConfig>
      <name>EnableNewAggregationStyle</name>
      <value>false</value>
   </featureConfig>
</featureConfigs>
```

Configuring advanced data operations

You can enable or disable the following advanced data options in Available Data in the report design area of BIRT Studio:

- Modify enables the user to change the data set by joining it with one or more information objects.
- Synchronize Data Sets enables the user to update the data set in the report design with the current data in the information object on the volume.

Figure 14-10 shows the data options in Available Data.

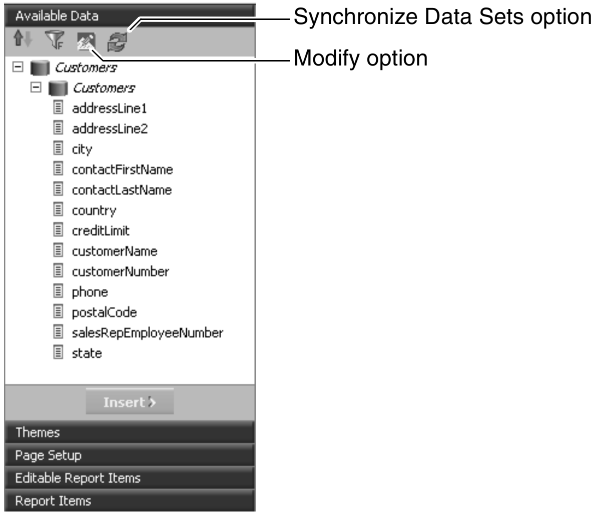


Figure 14-10 Advanced data options

By default, these data options are available only to users with the administrator or Active Portal Advanced security roles and they are in effect only when the data sources are information objects. When users with other roles log into BIRT Studio, the Modify and Synchronize Data Sets buttons do not appear.

You can enable these options for users with different roles by editing attributes in one of Information Console's configuration files, functionality-level.config. This configuration file defines roles, or levels, for Information Console users, from Basic to Administrator. Each role has a predefined set of functionality that the role can execute in Information Console. For information about all the functionality options listed in functionality-level.config, see *Information Console Developer Guide*.

Listing 14-4 shows the Intermediate and Advanced levels in functionality-level.config. The listing highlights in bold the advanced data option that is available to the Active Portal Advanced role but not the Active Portal Intermediate role.

Listing 14-4 Intermediate and Advanced levels in functionality-level.config

```
<Level>
     <Name>Intermediate</Name>
     <Role>Active Portal Intermediate</Role>
     <FeatureID>Jobs</FeatureID>
     <FeatureID>Documents</FeatureID>
     <FeatureID>Channels</FeatureID>
     <FeatureID>Search</FeatureID>
     <SubfeatureID>DeleteFile</SubfeatureID>
```

```
    <SubfeatureID>
      SelfNotificationWithAttachment
    </SubfeatureID>
    <SubfeatureID>SubscribeChannel</SubfeatureID>
    <SubfeatureID>InteractiveViewing</SubfeatureID>
    <SubfeatureID>DashboardDeveloper</SubfeatureID>
    <AnalyticsExperienceLevel>
      Novice
    </AnalyticsExperienceLevel>
    <AnalyticsExperienceLevel>
      Standard
    </AnalyticsExperienceLevel>
    <AnalyticsExperienceLevel>
      Advanced
    </AnalyticsExperienceLevel>
  </Level>
  <Level>
    <Name>Advanced</Name>
    <Role>Active Portal Advanced</Role>
    <FeatureID>Jobs</FeatureID>
    <FeatureID>Documents</FeatureID>
    <FeatureID>Channels</FeatureID>
    <FeatureID>Search</FeatureID>
    <SubfeatureID>DeleteFile</SubfeatureID>
    <SubfeatureID>
      SelfNotificationWithAttachment
    </SubfeatureID>
    <SubfeatureID>SubscribeChannel</SubfeatureID>
    <SubfeatureID>CreateFolder</SubfeatureID>
    <SubfeatureID>DeleteFolder</SubfeatureID>
    <SubfeatureID>ShareFile</SubfeatureID>
    <SubfeatureID>JobPriority</SubfeatureID>
    <SubfeatureID>InteractiveViewing</SubfeatureID>
    <SubfeatureID>AdvancedData</SubfeatureID>
    <SubfeatureID>DashboardDeveloper</SubfeatureID>
    <AnalyticsExperienceLevel>
      Novice
    </AnalyticsExperienceLevel>
    <AnalyticsExperienceLevel>
      Standard
    </AnalyticsExperienceLevel>
    <AnalyticsExperienceLevel>
      Advanced
    </AnalyticsExperienceLevel>
  </Level>
```

To enable the advanced data options for other roles, add the following line in the corresponding <Level> definition:

```
<SubfeatureID>AdvancedData</SubfeatureID>
```

You must add this line after the last <SubfeatureID> element. For example, to add the line to the Intermediate level, you must add it after <SubfeatureID>DashboardDeveloper</SubfeatureID>.

Unlike the functionality options you set in erni_config.xml, the options you set in functionality-level.config apply at the user role level, not at the application level. A single instance of BIRT Studio displays or hides the advanced data options, depending on the user login.

Configuring the application environment

You can change the values of the configuration parameters in the following file:

```
<context root>\WEB-INF\web.xml
```

BIRT Studio parameters control how BIRT Studio operates a web application and how it interoperates with Information Console. Table 14-2 describes the configuration parameters for BIRT Studio.

Table 14-2 BIRT Studio web.xml parameters

Parameter name	Description
BIRT_REPORT_DESIGN_ CACHE_TIMEOUT	Specifies the amount of time, in seconds, before a cached BIRT report design is purged if it has not been used. The default value it 1800, which is 30 minutes.
BIRT_REPORT_DESIGN_ CACHE_TOTAL_ NUMBER_OF_ENTRIES	Specifies the maximum number of BIRT report designs to cache. The default value is 50.
BIRT_REPORT_ DOCUMENT_CACHE_ ENABLED	Specifies whether to cache BIRT report documents when they are previewed or generated. The default value is true.
BIRT_REPORT_ PAGE_COUNT_CACHE_ ENABLED	Specifies whether to cache the number of pages in transient or persistent BIRT report documents when they are previewed or generated. The default value is true.
BIRT_RESOURCE_PATH	Path to Actuate BIRT shared resources, including libraries and templates for Actuate BIRT report designs and BIRT Studio. The default value is <context root>\resources.
BIRT_SCRIPT_LIB_PATH	Path to script libraries. The default value is <context root> \scriptlib.

(continues)

Table 14-2 BIRT Studio web.xml parameters (continued)

Parameter name	Description
DEFAULT_DATA_ CACHE_ROW_COUNT	The number of data rows to display in BIRT Studio when designing a report. The default value is 100.
DEFAULT_LOCALE	The default locale. The default locale is en_US. Users can select a locale when they log in.
DEFAULT_PAGE_BREAK_ INTERVAL	The number of rows to display on one page when viewing a report. A value of 0 indicates no page breaks.
DEFAULT_REPORT_ TEMPLATE_CATEGORY_ NAME	The default BIRT Report template category to load when a user opens BIRT Studio. The default value is Standard.
DEFAULT_TIMEZONE	The default time zone. The default time zone is Pacific Standard Time (PST).
MAX_BRSTUDIO_ DESIGN_SESSION	The maximum number of designs a user can edit concurrently in BIRT Studio. The default is 10.
MAX_BRSTUDIO_USER_ SESSION	The maximum number of concurrent BIRT Studio sessions on the server. The default is 256.
MAX_DATA_CACHE_ ROW_COUNT	Limits the number of data rows that a user can choose to display in Actuate BIRT Studio when designing a report. The default value is 200.
MAX_NUMBER_OF_ VALUES_FOR_ DYNAMIC_PARAMETER	The number of values shown in the parameter dialog box for a dynamic value parameter in BIRT Studio: ■ A positive number value N means only the first N values appear the parameter dialog box. ■ A value of 0 means all values from the data source appear in the parameter dialog box. 0 is the default value. ■ A value of -1 means only the first N values appear where N is the current data cache row count setting for the current design session. MAX_NUMBER_OF_VALUES_FOR_DYNAMIC_ PARAMETER only applies to a dynamic value parameter. All the values appear for a static value parameter no matter how many values it has. For a static value parameter, the full list appears in the parameter dialog box when the user chooses Save and View.
MEMORY_DATA_ CACHE_ROW_COUNT	Specifies the number of data rows to cache in memory. The default value is 50.
MORE_VALUE_ROW_ COUNT	Specifies the number of rows to fetch when a user chooses to filter a report on a column in BIRT Studio. The default value is 200.

Table 14-2 BIRT Studio web.xml parameters (continued)

Parameter name	Description
PERSISTENT_ ARCHIVEFILECACHE_ TIMEOUT_SECONDS	Specifies the amount of time, in seconds, before a cached file that was created from a repository file is purged if it has not been used. The default value it 7200, which is 120 minutes.
SEARCH_ENABLE_ COLUMN_HEADERS	Indicates whether to include column headings in report search results when the output format is CSV or TSV. Set this parameter to true, the default value, to include column headings.
SEARCH_USE_ QUOTE_DELIMITER	Indicates whether to enclose search results in quotation marks when the output format is CSV or TSV. The default value is true, which encloses the results in quotes.
TRANSIENT ARCHIVEFILECACHE_ TIMEOUT_SECONDS	Specifies the amount of time, in seconds, before a cached file generated without saving it to the repository if it has not been used. The default value it 1200, which is 30 minutes.

Chapter

15

Actuate BIRT Studio URIs

This chapter contains the following topics:

- Accessing BIRT Studio using a URI
- Using the BIRT Studio servlet
- Using the BIRT Studio URLs

Accessing BIRT Studio using a URI

BIRT Studio is a web application that is initiated by a Java servlet. The BIRT Studio servlet manages binary content and performs tasks such as uploading and downloading binary files.

You invoke the BIRT Studio servlet using the following syntax:

```
http://<application server>:<port>/<context root>/wr
```

where

- application server is the name of the machine hosting the application server.
- port is the port on which the application server listens for requests.
- context root is the BIRT Studio context root.
- wr is the name to which the servlet is mapped in the web application's web.xml file. A typical location for web.xml is C:\Program Files\Actuate11 \iPortal\iportal\WEB-INF.

Servlet names are case sensitive. Do not modify the servlets, their names, or their mapping in web.xml.

Using the BIRT Studio servlet

The BIRT Studio servlet loads the BIRT Studio user interface and establishes a connection to a report repository. A report repository is required in order to use the servlet.

Name com.actuate.erni.servlet.ERNIViewerServlet

Invoke the BIRT Studio servlet as:

```
http://<web server>:<port>/<context root>/wr?<parameters>
```

URI parameters The BIRT Studio servlet requires repository parameters in order to operate. Table 15-1 lists and describes the URI parameters for the BIRT Studio servlet.

Table 15-1 BIRT Studio URI parameters

URI parameter	Description
repositoryType	The repository type. Use Enterprise for an Encyclopedia volume.
serverURL	The URL of an iServer machine.
volume	The name of an Encyclopedia volume that is managed by the iServer URL to which you connect.

Table 15-1 BIRT Studio URI parameters

URI parameter	Description
__vp	The name of a server configured in VolumeProfile.xml. BIRT Report Studio uses the volume information in a VolumeProfile entry except when a volume parameter specifies a different one.

Using the BIRT Studio URLs

You can log in to BIRT Studio by typing a URL in a web browser's address field. After you type a URL and press Enter, the login page appears. What happens after you log in depends on which URL you use. In addition to the initial BIRT Studio page, you can open BIRT Studio with:

- A specific report design
- A specific template
- A report design that accesses a specific information object
- A report design that accesses a specific information object and a report template

In the example URLs in the following topics, special characters are represented by codes, as shown in Table 15-2.

Table 15-2 Codes for special characters in URLs

Character	Code
Colon (:)	%3a
Slash (/)	%2f
Period (.)	%2e
Space ( )	%20

How to log in

To access the BIRT Studio login page, use a URL like the one in the following example:

```
http://urup:8700/iportal/wr?repositoryType=Enterprise
   &serverurl=http%3a%2f%2furup%3a8000&__vp=urup
```

where

- urup:8700 is the name of the computer on which Information Console is installed and the port you use to access Information Console.

- iportal is the context root for Information Console.
- wr is the default context root for accessing BIRT Studio.
- ? indicates the beginning of a parameter that indicates where to access BIRT Studio files.
- repositoryType=Enterprise indicates that the repository type is Encyclopedia volume.
- &serverurl=http%3a%2f%2furup%3a8000&v__vp=urup specifies the URL to the Actuate iServer and Encyclopedia volume from the urup volume profile in which you work.

To log in and go directly to the BIRT Studio design environment, add userid and password parameters to the URL, as shown in the following example:

```
http://urup:8700/iportal/wr?repositoryType=Enterprise
   &userid=MyUser
   &password=MyPwd
   &serverurl=http%3a%2f%2furup%3a8000
   &__vp=urup
```

Sending security information such as the user name and password in a URL is not a recommended approach.

How to open BIRT Studio and load an existing report design

To open an existing report design in BIRT Studio, use a URL like the one shown in the following example:

```
http://urup:8700/iportal/wr?repositoryType=Enterprise
&serverurl=http%3a%2f%2furup%3a8000
&__report=%2fHome%2fPublic%2fCustomers%2erptdesign
&pCountry=USA
```

where

- __report=%2fHome%2fPublic%2fCustomers%2erptdesign is the path to the report design to use.
- pCountry=USA is a parameter-value pair for the report design.

Index

Symbols

Numerics

A

B

C

D

E

F

G

H

I

J

L

M

P

Q

R

S

T