

# ActuateOne™

One Design  
One Server  
One User Experience

**Using BIRT Studio - Java Component 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](http://www.actuate.com)

[www.birt-exchange.com](http://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](http://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](http://www.castor.org)), ExoLab Project ([www.exolab.org](http://www.exolab.org)), and Intalio, Inc. ([www.intalio.org](http://www.intalio.org)): Castor. Codejock Software: Xtreme Toolkit Pro. DataDirect Technologies Corporation: DataDirect JDBC, DataDirect ODBC. Eclipse Foundation, Inc. ([www.eclipse.org](http://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 Hsueh and Kenton Varda ([code.google.com](http://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](http://www.zlib.net)): zlib. Matt Ingenthron, Eric D. Lambert, and Dustin Sallings ([code.google.com](http://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](http://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, JTIty, Simple API for CSS. XFree86 Project, Inc.: ([www.xfree86.org](http://www.xfree86.org)): xvfb. Yuri Kanivets ([code.google.com](http://code.google.com)): Android Wheel gadget, licensed under the Apache Public License (APL). ZXing authors ([code.google.com](http://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. 110812-2-761301 August 5, 2011

# Contents

<b>About <i>Using BIRT Studio - Java Component Edition</i> . . . . .</b>	<b>.ix</b>
--	------------

## Part 1

## **Designing reports using BIRT Studio**

### Chapter 1

<b>Getting started . . . . .</b>	<b>3</b>
About BIRT Studio . . . . .	4
Starting BIRT Studio . . . . .	4
Creating a report . . . . .	5
Selecting a template . . . . .	6
Selecting a data source . . . . .	8
Editing and formatting a report . . . . .	13
Saving a report . . . . .	14
Running and viewing a report . . . . .	15
Working with bookmarks . . . . .	15

### Chapter 2

<b>Editing and formatting report content . . . . .</b>	<b>17</b>
Editing and formatting options . . . . .	18
Removing items from a report . . . . .	18
Editing labels . . . . .	19
Applying a theme . . . . .	19
Selecting an item for formatting . . . . .	20
Changing font properties . . . . .	21
Changing the alignment of text . . . . .	22
Applying a number format . . . . .	22
Selecting a standard number format . . . . .	23
Defining a custom number format . . . . .	24
Applying a date-and-time format . . . . .	25
Selecting a standard date-and-time format . . . . .	25
Defining a custom date-and-time format . . . . .	26
Applying a string format . . . . .	27
Selecting a standard string format . . . . .	27
Defining a custom string format . . . . .	28
Formatting data based on conditions . . . . .	29
Specifying a condition . . . . .	30
Comparing to a literal value . . . . .	31
Comparing to a value in another column . . . . .	32
Specifying multiple conditional formatting rules . . . . .	33

Reverting to default formats .....	34
Changing the order of columns .....	35
Merging data into one column .....	36
Creating a merged column .....	37
Working with data in a merged column .....	38
Adding a new column header row .....	38
Changing the page layout to fit data .....	41
Adjusting column widths .....	42
Changing the layout type, page size, and orientation .....	43

## Chapter 3

<b>Inserting calculated data .....</b>	<b>45</b>
About calculated data .....	46
Creating a computed column .....	46
About expressions .....	47
Creating aggregate data in a computed column .....	49
Adding data fields for an expression .....	49
Writing expressions .....	51
Using numbers and dates in an expression .....	52
About reserved characters in an expression .....	52

## Chapter 4

<b>Organizing data in a report .....</b>	<b>55</b>
Sorting data .....	56
Sorting on a single column .....	57
Sorting on multiple columns .....	57
Restoring data to its original order .....	58
Organizing data in groups .....	58
Grouping data on a date-and-time column .....	61
Grouping on multiple columns .....	62
Changing the grouping order .....	62
Deleting a group .....	63
Organizing data in sections .....	63
Displaying additional information in a section heading .....	65
Deleting a section .....	66
Aggregating data .....	68
Formatting aggregate values .....	70
Filtering aggregate values .....	71
Hiding details .....	72
Starting each group or section on a new page .....	74

## Chapter 5

<b>Filtering data .....</b>	<b>77</b>
About data filtering .....	78

About filtering options .....	78
Creating a filter .....	79
Creating a filter at the report table level .....	79
Selecting multiple values for a filter condition .....	81
Excluding data .....	82
Filtering empty or blank values .....	82
Displaying top or bottom values .....	82
Specifying a date as a comparison value .....	84
Comparing to a string pattern .....	84
Comparing to a value in another column .....	85
Defining multiple filter conditions .....	86
Adding a condition .....	86
Changing a condition .....	91
Deleting a filter condition .....	91
Creating a filter at the data-set level .....	92
Prompting for filter values at run time .....	94
About report parameters .....	94
Choosing a parameter type .....	94
Making a filter parameter optional .....	95
Accepting multiple values .....	95
Specifying a display type .....	96
Providing the user with a list of values .....	97
Creating a static filter parameter .....	97
Creating a dynamic filter parameter .....	104
Creating multiple filter parameters .....	108

## Chapter 6

### **Working with summary tables ..... 111**

About summary tables .....	112
About columns in a summary table .....	112
Designing a summary table .....	112
Creating a summary table .....	114
Modifying a summary table .....	117
Modifying labels .....	120
Working with a dimension column .....	120
Modifying a dimension column .....	121
Moving a dimension column .....	122
Deleting a dimension column .....	122
Working with an attribute column .....	122
Inserting an attribute column .....	122
Modifying an attribute column .....	124
Moving an attribute column .....	124
Deleting an attribute column .....	124
Working with a measure column .....	124

Inserting a measure column .....	124
Modifying a measure column .....	125
Adding an aggregate calculation .....	126
Moving a measure column .....	126
Deleting a measure column .....	126
Formatting aggregate data values .....	126
Inserting a chart in a summary table .....	127
Changing table types .....	128

## Chapter 7

### **Presenting data in a cross tab ..... 129**

About cross tabs .....	130
Obtaining data for a cross tab .....	131
Creating a cross tab .....	131
Selecting data for a cross tab .....	131
Laying out data in a cross tab .....	136
Displaying aggregate values .....	137
Saving and viewing a cross tab .....	139
Formatting data in a cross tab .....	139
Analyzing data in a cross tab .....	140

## Chapter 8

### **Presenting data in a chart ..... 141**

About charts .....	142
Creating a chart .....	143
Inserting a chart .....	144
Choosing a chart type .....	145
About area charts .....	146
Stacked area chart .....	146
Overlay area chart .....	147
Percent stacked area chart .....	147
About bar charts .....	148
Side-by-side bar chart .....	148
Stacked bar chart .....	148
Percent stacked bar chart .....	149
Horizontal bar chart .....	150
About line charts .....	150
Overlay line chart .....	151
Stacked line chart .....	151
Percent stacked line chart .....	152
About meter charts .....	153
Superimposed meter chart .....	153
Standard meter chart .....	153
About pie charts .....	153

About scatter charts .....	154
About stock charts .....	155
Candlestick stock chart .....	155
Bar stick stock chart .....	155
Selecting data for a chart .....	156
Selecting data for an area, bar, or line chart .....	158
Selecting data for a pie chart .....	161
Selecting data for a meter chart .....	165
Selecting data for a scatter chart .....	168
Selecting data for a stock chart .....	170
Displaying additional series information .....	172
Formatting a chart .....	173
Changing the size of a chart .....	174
Setting a chart's width .....	174
Setting a chart's height .....	175
Editing and formatting titles and labels .....	175
Making a chart look three-dimensional .....	176
Changing the position of the legend .....	177
Formatting labels to fit on the x-axis .....	178
Rotating x-axis labels .....	179
Staggering x-axis labels .....	179
Displaying x-axis labels by interval .....	180
Changing the range of values on the y-axis .....	181
Working with Flash charts .....	182
Creating a Flash chart .....	183
Selecting data for a Flash chart .....	184
Formatting a Flash chart .....	186
Formatting a column, line, or bar chart .....	186
Specifying chart properties .....	186
Formatting the legend .....	187
Formatting the category (x) axis .....	187
Formatting the value (y) axis .....	188
Formatting a pie or doughnut chart .....	188
Deleting a Flash chart .....	189
Limitations .....	190
Displaying a chart without the table data .....	190
 Chapter 9	
<b>Functions and operators .....</b>	<b>191</b>
Functions .....	192
Functions used in computed column expressions .....	192
ABS() .....	192
ADD_DAY() .....	192

ADD_HOUR()	193
ADD_MINUTE()	193
ADD_MONTH()	194
ADD_QUARTER()	194
ADD_SECOND()	195
ADD_WEEK()	195
ADD_YEAR()	195
BETWEEN()	196
CEILING()	197
DAY()	197
DIFF_DAY()	198
DIFF_HOUR()	198
DIFF_MINUTE()	199
DIFF_MONTH()	199
DIFF_QUARTER()	200
DIFF_SECOND()	200
DIFF_WEEK()	201
DIFF_YEAR()	202
FIND()	202
IF()	203
IN()	204
ISNULL()	205
LEFT()	205
LEN()	206
LIKE()	206
LOWER()	207
MATCH()	207
MOD()	208
MONTH()	208
NOT()	209
NOTNULL()	209
NOW()	210
QUARTER()	210
RIGHT()	211
ROUND()	211
ROUNDDOWN()	212
ROUNDUP()	213
SEARCH()	214
SQRT()	214
TODAY()	215
TRIM()	215
TRIMLEFT()	216
TRIMRIGHT()	216



UPPER( ) .....	216
WEEK( ) .....	217
WEEKDAY( ) .....	217
YEAR( ) .....	218
Functions used in aggregate calculations .....	218
Operators .....	220
Operators in computed column expressions .....	220
Operators in conditional formatting and filter condition expressions .....	222

## Part 2

# Customizing and Integrating BIRT Studio

## Chapter 10

### **Introduction to BIRT Studio customization ..... 227**

About BIRT Studio and the Actuate Java Components .....	228
Required Actuate software .....	228
Understanding the context root .....	229

## Chapter 11

### **Creating and publishing report templates ..... 231**

About report templates .....	232
Design considerations .....	232
Separating or combining visual and data elements .....	232
Designing themes .....	233
Improving usability .....	233
Creating a report template .....	235
Providing data with a report template .....	238
Using a CSV file as a data source .....	238
Excluding a data set .....	239
Creating themes for a report template .....	239
Publishing a template .....	243
Setting the default template category .....	244
Publishing resources associated with a template .....	244

## Chapter 12

### **Providing data ..... 247**

Types of data sources .....	248
Creating a custom data source .....	248
Sample ODA data source .....	249
Configuring a sample ODA data source .....	250
Extended sample ODA example .....	252
Developing an Open Data Access (ODA) data driver .....	254
Developing a custom query builder .....	254

Creating the servlet .....	255
Compiling the servlet .....	256
Deploying the servlet .....	257
Registering the servlet .....	257
Configuring the extended custom data source .....	258
 Chapter 13	
<b>Configuring BIRT Studio .....</b>	<b>259</b>
Enabling or disabling functionality .....	260
Configuring toolbar and context menu items .....	260
Configurable actions .....	261
Configuration examples .....	264
Specifying the default position of aggregate values .....	267
Configuring the application environment .....	268
 Chapter 14	
<b>Actuate BIRT Studio URIs .....</b>	<b>271</b>
Accessing BIRT Studio using a URI .....	272
Using the BIRT Studio servlet .....	272
Using the BIRT Studio URLs .....	273
BIRT Studio file manager URIs overview .....	273
BIRT Studio file manager URIs quick reference .....	274
Common URI parameters .....	275
BIRT Studio file manager URIs reference .....	275
banner page .....	276
browse file page .....	277
delete file status page .....	277
detail page .....	277
drop page .....	278
error page .....	279
execute report page .....	280
index page .....	282
list page .....	282
page not found page .....	284
parameters page .....	284
Actuate BIRT Viewer URIs reference .....	284
 <b>Index .....</b>	<b>285</b>

# About Using BIRT Studio - Java Component Edition

---

*Using BIRT Studio - Java Component 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 - Java Component 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 create bookmarks in a report design.
- *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.
- *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. Working with summary tables.* This chapter describes how to create and modify summary tables, as well as how to change table types.

- *Chapter 7. 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 8. Presenting data in a chart.* This chapter describes the types of charts you can create and discusses the procedures for displaying data in regular charts and 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.
- *Part 2. Customizing and Integrating BIRT Studio.* This part is a guide to customizing and integrating BIRT Studio.
- *Chapter 10. 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 11. 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 12. Providing data.* This chapter describes the different ways to provide BIRT Studio users with data for their reports. The chapter provides guidelines for developing custom data sources.
- *Chapter 13. 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 14. 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**



# 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 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.

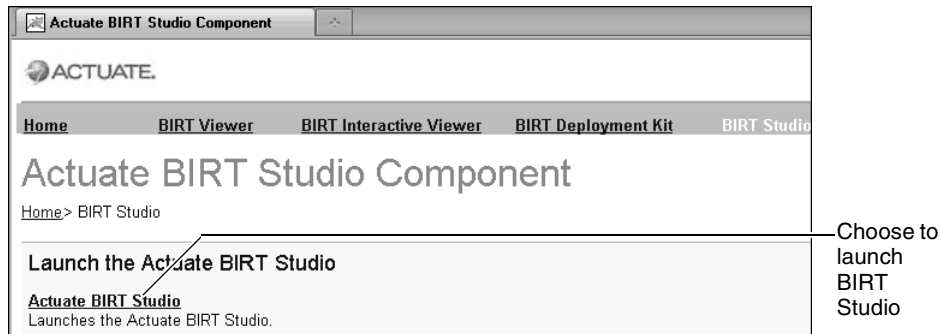
BIRT Studio is a highly customizable application. A system or application administrator creates and integrates custom data sources and templates that you use to create reports. An administrator can also customize the user interface and functionality in BIRT Studio to tailor the tool to your organization's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

---

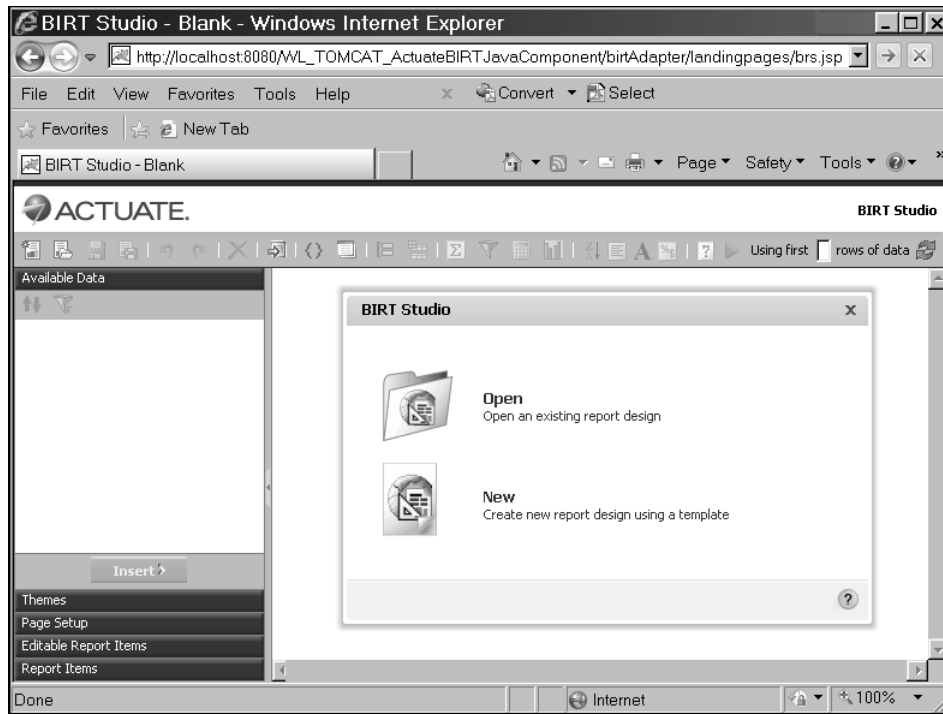
## Starting BIRT Studio

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. In a typical installation, you start BIRT Studio after you log into another application. Figure 1-1 shows an example of an application that includes a link you choose to launch BIRT Studio.



**Figure 1-1** Launching BIRT Studio from an application  
BIRT Studio opens in your web browser, as shown in Figure 1-2.





**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.

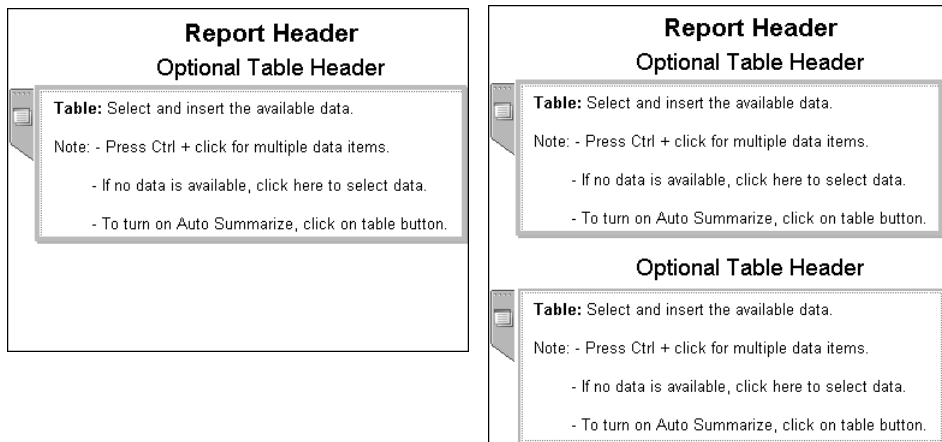
## Selecting a template

Every report you create in BIRT Studio is based on a template. A template determines the basic structure for a report 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, page number, or author name

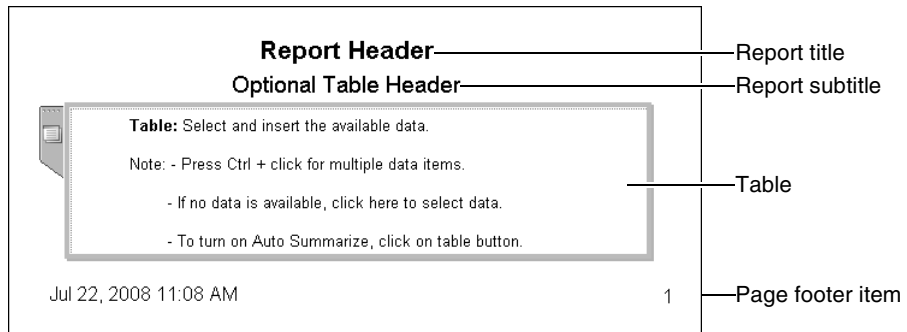
shows two templates included in the default installation of BIRT Studio.

Your version of BIRT Studio most likely provides custom templates designed for the types of reports you need to create. BIRT Studio organizes templates into categories. A category is simply a convenient way to group similar templates. Templates can be organized by report types or by departments, or they can 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.



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

Figure 1-4 identifies the items a template typically includes. Notice that the table contains instructions for inserting data. Templates often provide information about using items. After you select a template and data source, a typical next step is to insert data in a table.

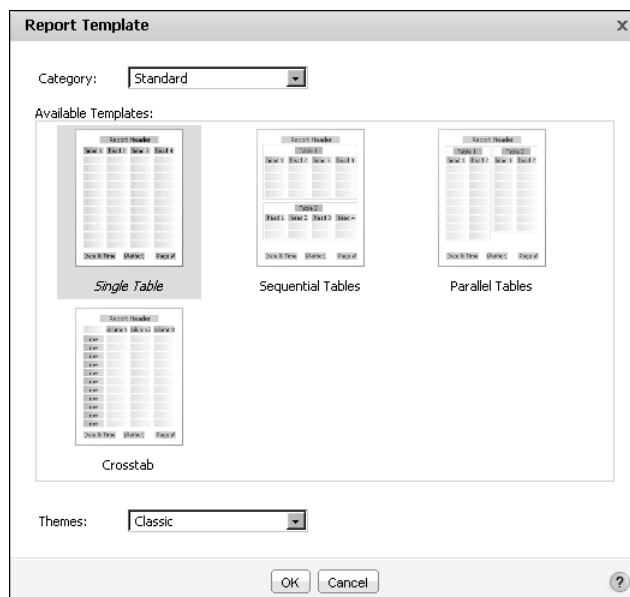


**Figure 1-4** Items in a typical template

### How to select a template for your report

- 1 When you create a new report, BIRT Studio displays Report Template. On 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 can be one of the following:

- A data set included in the selected template
- A custom data source, such as an ODA data source

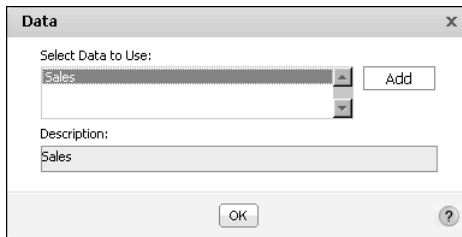
These data sources contain all the information to connect to a data repository, such as a database, and retrieve a specific set of data. Both types of data sources provide easy access to the data fields that contain the data for a report. For example, a data set named CustomerData with customer information 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. The data in the data set is often organized into categories called dimensions, measures and attributes. This classification is set by the developer at the time of designing the data set, and cannot be modified in BIRT Studio.

You can limit the data available to a report design by applying filters to a data set in a template. 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.

BIRT Studio also provides a convenient Table Builder wizard that enables you select data fields from a data set in a template, and insert them into the report table. Alternately, if you want to preview the data as you select and insert the data fields into the report design, you can manually select and insert fields in a report design. If you use a custom data source, you must first select the data fields in the ODA editor, and then manually select and insert them into a report design. The following section describes how to create a report design.

### How to create a report design using a data set in a template

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



**Figure 1-6** Selecting a data set

Data displays the data sets included in the template.

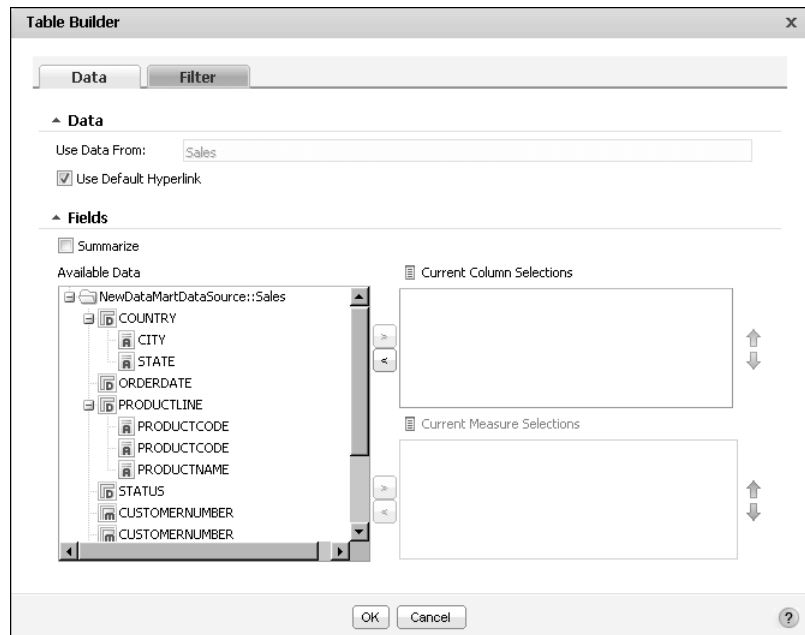
**2** Select a data set, then choose OK.

Table Builder appears listing the data fields in the selected data set. 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.
- Select Cancel on Table Builder to manually select and insert data fields in a report table. If you choose this option, see “How to manually select and insert data fields in a report table.”

**3** On Table Builder, shown in Figure 1-7, complete the following steps:

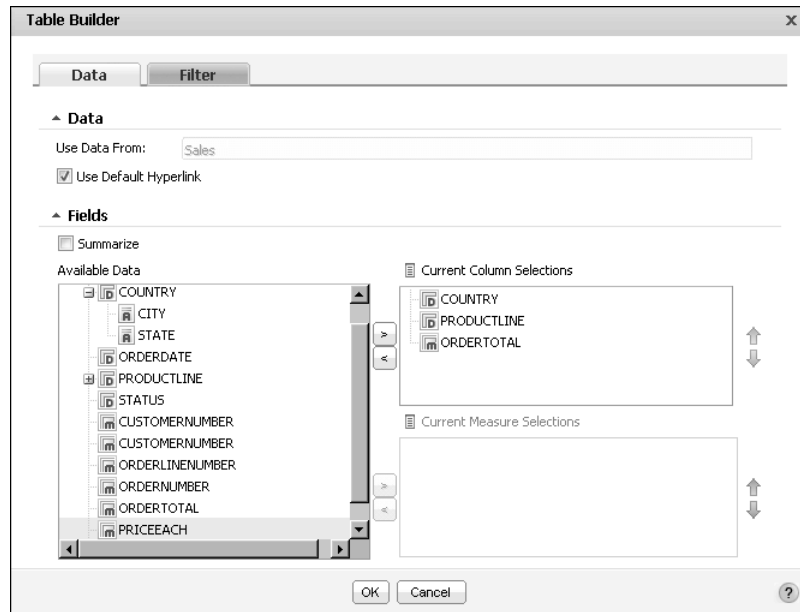
- 1 In Data, in Use Data From, select a different data set, if necessary.
- 2 In Fields, do one of the following:
  - To create a summary table, select Summarize. For more information about creating a summary table see Chapter 6, “Working with summary tables.”
  - To create a detail report, deselect Summarize.



**Figure 1-7** Table Builder displaying fields in the selected data set

- 3** In Available Data, press Ctrl, and select each data field to use. Then select the right arrow. The selected data fields appear in Current Column Selections, as shown in Figure 1-8. Use the up and down arrows to

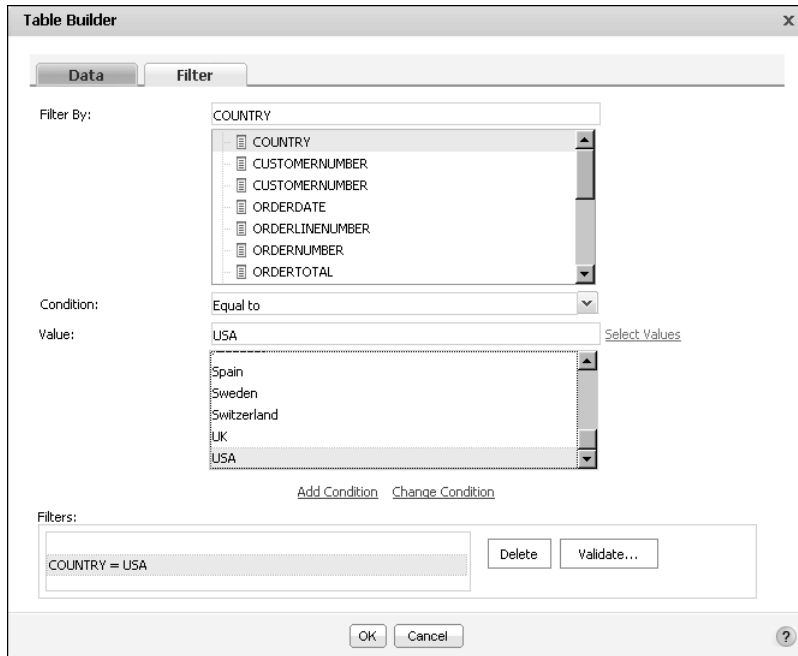
rearrange the order in which the data fields are displayed in the report design.



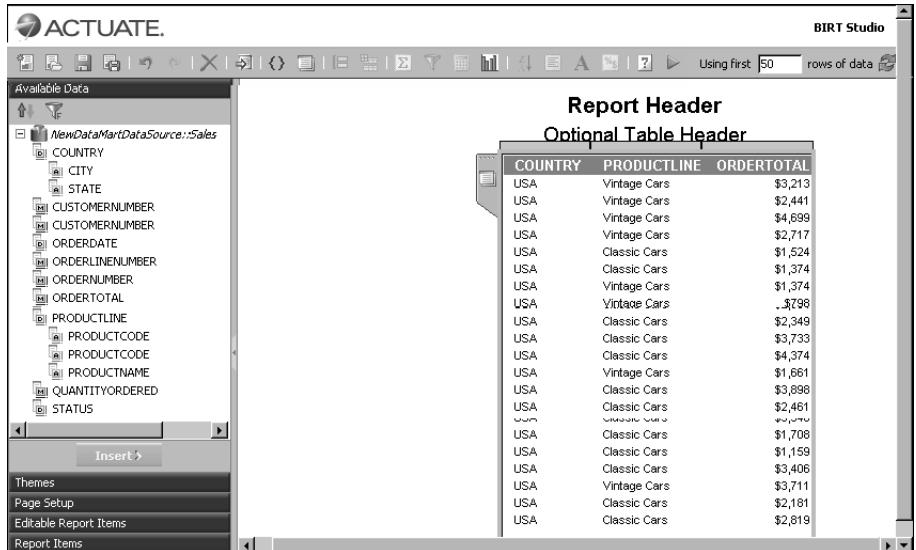
**Figure 1-8** Current Column Selections displaying the selected data fields

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.

- 4 In Table Builder, select Filter.
- 5 In Filter By, select the column to evaluate.
- 6 In Condition, select a comparison operator from the drop-down list. Based on your selection, either one or two fields appear in Value.
- 7 In Value, do one of the following:
  - Type a value.
  - Choose Select Values, then select a value from the list of values that appears.
- 8 Choose Add Condition. The condition appears in Filters, as shown in Figure 1-9.
- 9 Choose Validate to verify that the syntax of the condition is valid. After a message appears confirming that the syntax of the filter condition is valid, choose OK.



**Figure 1-9** Specifying a data-set filter condition



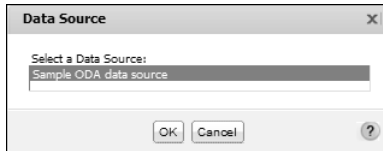
**Figure 1-10** BIRT Studio displaying the selected data

BIRT Studio displays the selected data fields in the report design, as shown in Figure 1-10.

## How to create a report design using a custom data source

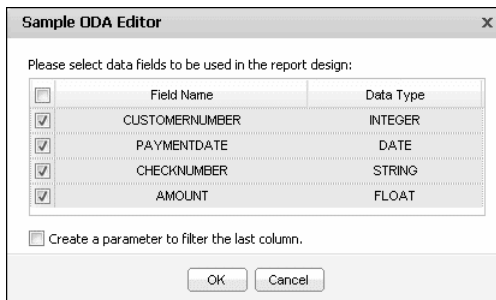
If you use a custom data source, you must manually select and insert data fields in a report table.

- 1 On Report Template, select a template that does not contain a data set. Choose OK.
- 2 Data Source appears, displaying available sample and custom data sources, as shown in Figure 1-11. Select the data source to use. Choose OK.



**Figure 1-11** Selecting a data source

- 3 Sample ODA Editor appears, as shown in Figure 1-12. Select the data fields to use in the report design.



**Figure 1-12** Selecting data fields

- 4 Optionally select Create a parameter to filter the last column.
  - Select this option to activate any existing parameters set for the last data field, Amount, shown in Figure 1-12. After you insert the data field in the report design, choose Parameters and make your selection to specify viewing parameters for data in that column.
  - Deselect this option to view all the data in the data field.

Choose OK. The selected data fields appear in Available Data in BIRT Studio. The next step is to select and insert the data fields to use in the report table, by following the procedure in the following section, “How to manually select and insert data fields in a report table.”

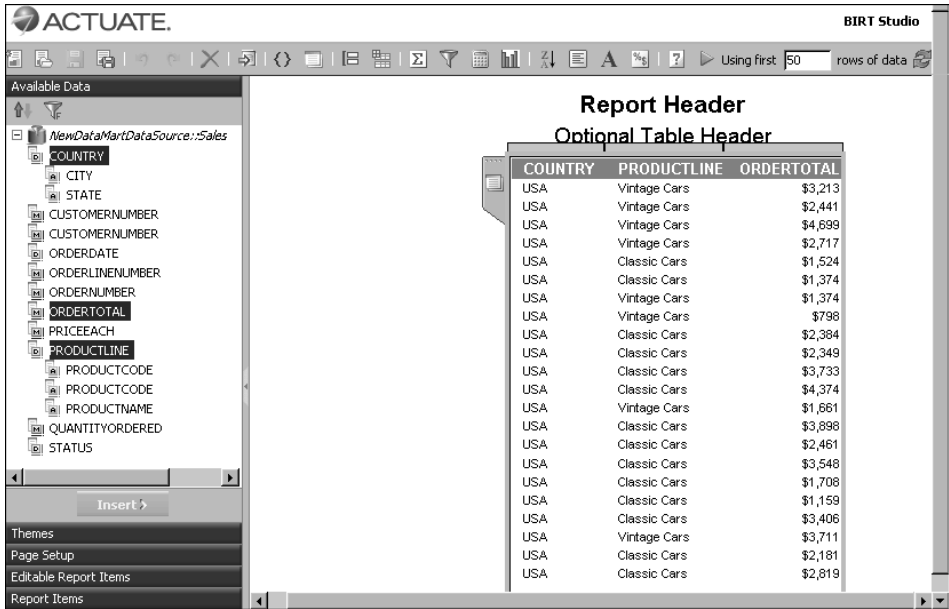
## How to manually select and insert data fields in a report table

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 1-13 shows an example of three data fields, Country, Product Line, and Order Total, inserted in the report table.

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.



**Figure 1-13** BIRT Studio displaying the selected data

You are now ready to format the report design. For more information on how to format a report, see Chapter 2, “Editing and formatting report content.”

## Editing and formatting a report

So far, you have learned how to create a simple listing report in three steps. The next step is to enhance the report design. At the very least, the example report shown in Figure 1-13 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 the 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

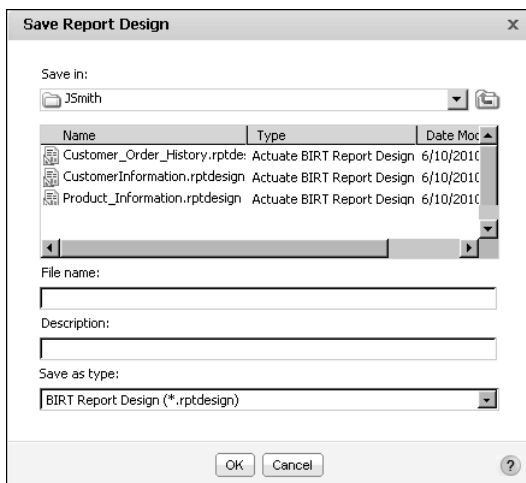
Depending on the platform on which you are working, you save a report to a folder in a file system. Your access privileges determine the folders in which you can save a report.

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, the report is saved 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 1-14. This dialog box typically displays your home folder.
- 2 Navigate through the folder structure to select a different folder.



**Figure 1-14** Save Report Design

- 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.

## 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**.

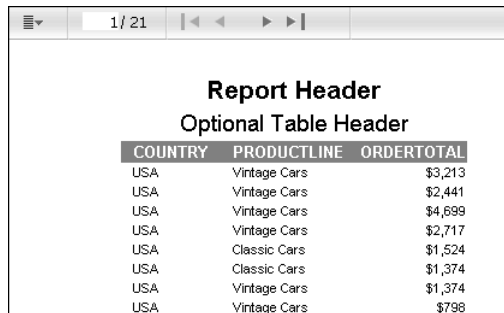
---

## Running and viewing a report



As you design a report, the data you see 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**. The generated report appears in **Actuate BIRT Viewer** in your web browser. Figure 1-15 shows the viewer displaying the finished version of the report design that was shown in Figure 1-13.

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*.



The screenshot shows the Actuate BIRT Viewer interface. At the top, there is a navigation bar with a dropdown menu, a page indicator '1 / 21', and navigation buttons. The main content area displays a report with a 'Report Header' section, followed by an 'Optional Table Header' section. Below this is a table with three columns: 'COUNTRY', 'PRODUCTLINE', and 'ORDERTOTAL'. The table contains eight rows of data.

Report Header		
Optional Table Header		
COUNTRY	PRODUCTLINE	ORDERTOTAL
USA	Vintage Cars	\$3,213
USA	Vintage Cars	\$2,441
USA	Vintage Cars	\$4,699
USA	Vintage Cars	\$2,717
USA	Classic Cars	\$1,524
USA	Classic Cars	\$1,374
USA	Vintage Cars	\$1,374
USA	Vintage Cars	\$798

**Figure 1-15** Actuate BIRT Viewer displaying a report

---

## Working with bookmarks

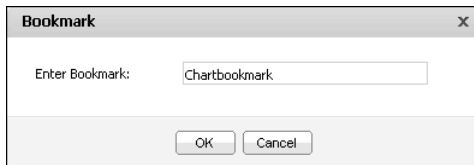
A bookmark is a marker that defines a place in a report. You can use **BIRT Studio** to create a bookmark to a report table, a regular chart or Flash chart in a report table or section, a summary table, or a cross tab.

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 1-16. In Enter Bookmark, type a name for the bookmark. Choose OK.



**Figure 1-16** 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*.

# 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
- Reverting to default formats
- 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 you can use 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 can choose which report items to include in your report design.

In a typical template, some report items are editable, and others are not. Examples of items you can edit 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. Templates save time and effort that you would otherwise spend formatting your report to make it visually appealing. Sometimes, however, you need to make formatting changes to refine the appearance of a report. A template serves as a good base for reports, but the particular contents in individual reports can require you to change, for example, the size of the page, the width of columns, or the size of the font.

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, by default, all the items in the template appear in the report. If there is an item, such as a report-generation date or a label, that you do not want to display in your report, you can exclude the item. If you change your mind, you can select the item for display again.

### **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.
- 2 Select the items to display in the report. If you are not sure which item in the list corresponds to which item in the report design, select or deselect one item at a time, choose OK, and see what item appears or disappears from the design.



**Figure 2-1** Report items included in a template

---

## 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

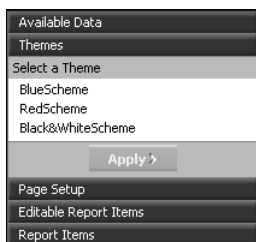
A theme in a template is similar to a theme in Microsoft Windows. A Windows theme helps you personalize, with one click, your computer display to show a particular background and custom color and fonts settings for various items, such as title bars, buttons, and icons. Similarly, a template theme can 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 in specific report items, the spacing between rows, the styles of borders, the alignment of text, and much 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 actions you can perform on a 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 2-4 shows an example of a selected column.

Customer	Country	Credit Limit
Australian Collectors, Co.	Australia	117300
Danish Wholesale Imports	Denmark	83400
Toys of Finland, Co.	Finland	96500
Atelier graphique	France	21000
La Rochelle Gifts	France	118200
Saveley & Henriot, Co.	France	123900

**Figure 2-3** A selected column header in a table



Customer	Country	Credit Limit
Australian Collectors, Co.	Australia	117300
Danish Wholesale Imports	Denmark	83400
Toys of Finland, Co.	Finland	96500
Atelier graphique	France	21000
La Rochelle Gifts	France	118200
Saveley & Henriot, Co.	France	123900

**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

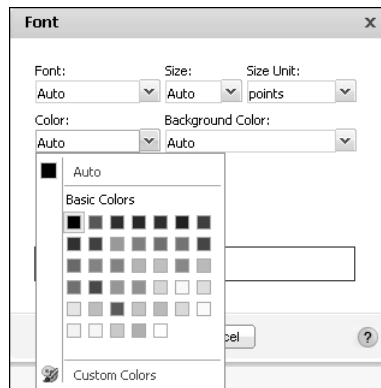
You can change the font properties of editable labels, column headers, and data in a column. The properties you can set include the font, font size and color, and the background color, and you can make the text bold, italic, or underlined.

### How to change font properties

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



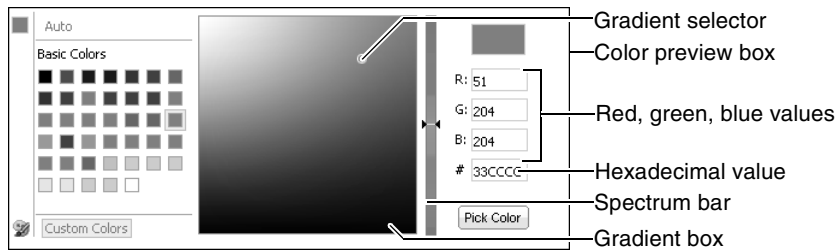
- 2 Choose Font. Font, shown in Figure 2-5, displays the current font property values. A value of Auto means the property uses a default value, which is specified by a theme, the template, or the software.



**Figure 2-5** Selecting a Font

- 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.



**Figure 2-6** Specifying a custom color using color picker

The selected options appear in the Preview field. The RGB value of the selected color appears in Color. Choose OK.

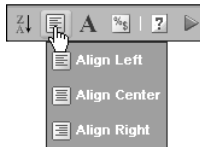
## Changing the alignment of text



You can change the alignment of text 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.

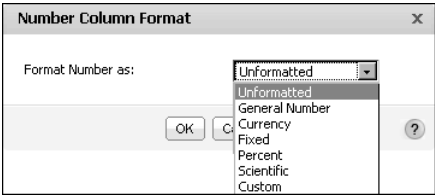
**How to apply a number format**

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 appears different. 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.

*(continues)*

**Table 2-1** Standard number formats (continued)

Format	Example of data display
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.
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 where to place currency symbol, as well as which characters to use for the thousands separator, and the decimal separator. The thousands separator and the decimal separator are locale-dependent. In the US (English) locale, for example, the thousands separator is the comma, and the decimal separator is the period. In most European locales, the thousands separator is a space, and the decimal separator is the comma. Table 2-2 shows examples of custom format patterns and their effects on numeric data.

**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, 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, and so on.

### 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 and Figure 2-10 show the sample formats that appear in the English (United States) and the Japanese locales, respectively.

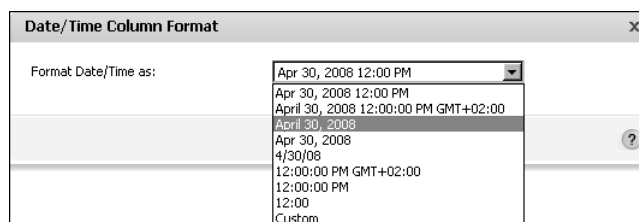


Figure 2-9 Date-and-time formats displayed in English (United States) locale



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, shown in Table 2-3, to construct a format pattern. Use custom date formatting only for reports 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 date-month-year format, the date 01-10-06 is interpreted as October 1, 2006.

**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** Results of custom date formats

Format pattern	Result of formatting
MM-dd-yy	04-15-06

**Table 2-4** Results of custom date formats (continued)


Format pattern	Result of formatting
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)
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

**Table 2-5** Standard string formats (continued)

Format	Description
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. See Table 2-7 for examples.
&	Same as @, except if the string has fewer characters, spaces do not appear. See Table 2-7 for examples.
!	Specifies that placeholders are to be filled from left to right. See Table 2-7 for examples.
>	Converts string characters to uppercase.
<	Converts string characters to lowercase.

Table 2-7 shows examples of custom string format patterns and their effects on string data.

**Table 2-7** Results 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



**Table 2-7** Results of custom string formats (continued)

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

# Formatting data based on conditions

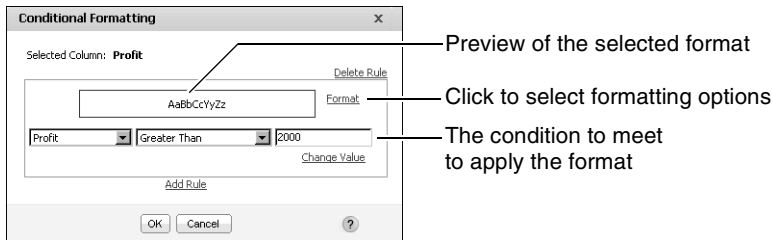
When you format data in a selected column, the format applies to all the values. Often, it is useful to change the format of data when a certain condition is true. For example, you can display sales numbers in red if the value is a negative number and in black if the value is a positive number.

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 to define when and how to change the appearance of data. You can apply conditional formats only to data in columns. The following example shows an example of a rule specified for a selected column, Profit.

## 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. Conditional Formatting appears, as shown in Figure 2-11.



**Figure 2-11** Conditional Formatting displaying a rule

- 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.
- 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.

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	<b>\$3,268.86</b>
	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	<b>\$2,778.30</b>
	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	<b>\$2,115.33</b>
	1958 Setra Bus	\$2,554.44	\$918.54
	1996 Peterbilt 379 Stake Bed with Outtrigger	\$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

## Specifying a condition

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

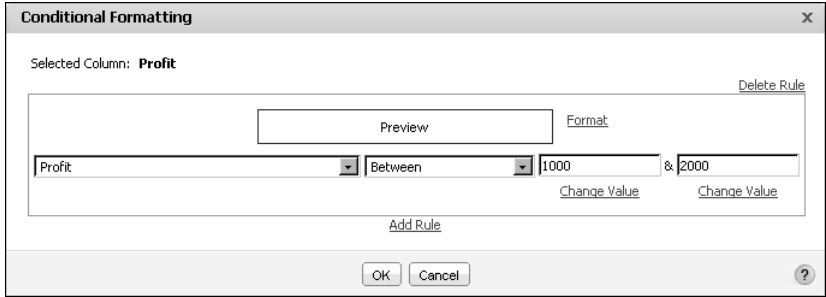
If the order total is less than 1000  
 If the customer credit limit is between 10000 and 20000  
 If the sales office is Tokyo  
 If the order date is 7/21/2008

The Conditional Formatting dialog box helps you construct the If expression by breaking it down to its logical parts. In Figure 2-12, 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

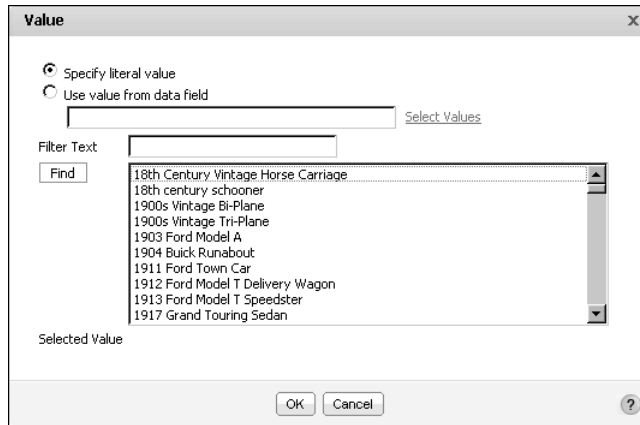
## Comparing to a literal value

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.
- 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.



**Figure 2-14** Selecting a comparison value

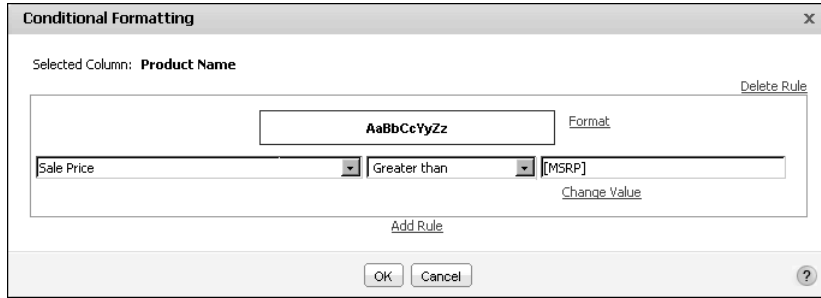
## Comparing to a value in another column

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 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.

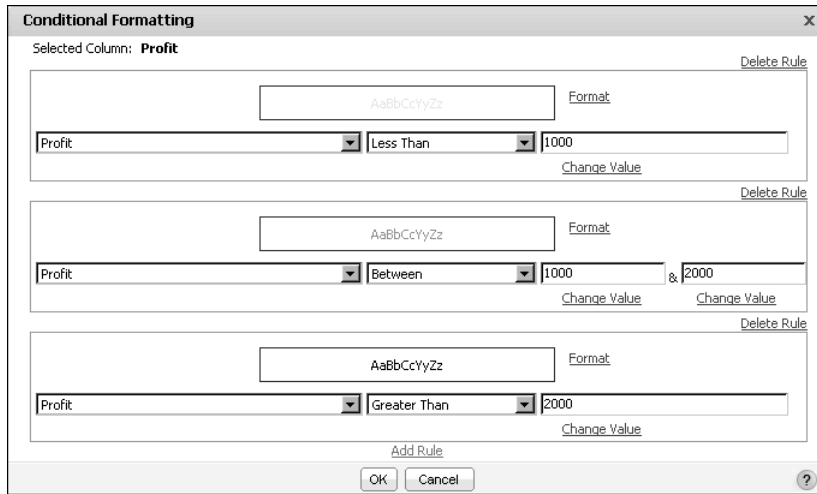
Product Line	Product Name	Sale Price	MSRP
Classic Cars			
	1948 Porsche 356-A Roadster	\$66.99	\$69.30
	<b>1956 Porsche 356A Coupe</b>	\$139.03	\$126.39
	1957 Corvette Convertible	\$133.92	\$133.92
	<b>1961 Chevrolet Impala</b>	\$76.80	\$72.76
	<b>1968 Dodge Charger</b>	\$110.39	\$105.70
	1968 Ford Mustang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

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.



**Figure 2-17** Defining multiple conditional formatting rules

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

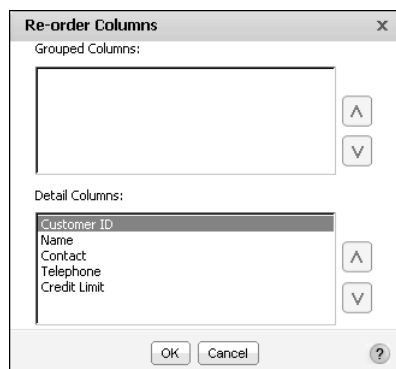
Use one of the following methods to change the order of columns:

- 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 2-18.

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 Plestrzeniewicz	(26) 642-7555	\$0
128	Blauer See Auto, Co.	Roland Kettel	+49 69 66 90 2555	\$59,700
129	Mini Wheels Co.	Julie Murphy	6505555787	\$64,800
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	\$63,400
146	Saveley & Henriot, Co.	Mary Saveley	78.32.5555	\$123,900
148	Dragon Souvenirs, 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

**Figure 2-18** Moving a column

- Right-click any selected column and choose Column>Reorder Columns. Reorder Columns appears as shown in Figure 2-19.



**Figure 2-19** Changing the order of columns

On Re-order Columns, shown in Figure 2-19, select each column to move, and 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.

# 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.

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
Mini VWheels Co.	(650) 555-5787	5557 North Pendale Street	San Francisco	CA	94217
Muscle Machine Inc	(212) 555-7413	4092 Furth Circle	NYC	NY	10022
Online Mini Collectables	(617) 555-7555	7635 Spinnaker Dr.	Brickhaven	MA	58339
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
Toys4GrownUps.com	(626) 555-7265	78934 Hillside Dr.	Pasadena	CA	90003
Vitachrome Inc.	(212) 555-1500	2678 Kingston Rd.	NYC	NY	10022

**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 VWheels Co.	(650) 555-5787	5557 North Pendale Street San Francisco CA 94217

**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

**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

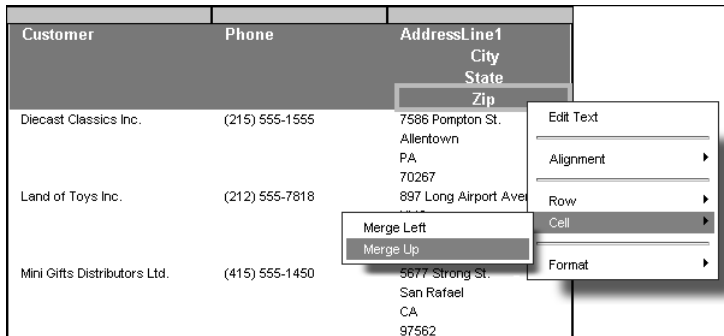
- 1 Select all the columns that contain the data to merge. Press Ctrl as you select each column.
- 2 Right-click one of the selected columns, then choose Column→Merge Columns, as shown in Figure 2-23.

Name	Phone	AddressLine1	City	State	Zip
Cambridge Collectables Co.	(617)555-5555	4858 Baden Av.	Cambridge	MA	51247
Gift Depot Inc.	(203)555-2570	25593 South Bay Ln.			97562
Vitachrome Inc.	(212)555-1500	2678 Kingston Rd.			
Auto-Moto Classics Inc.	(617)555-8428	16780 Pompton St.			
Online Mini Collectables	(617)555-7555	7635 Spinnaker Dr.			
Toys4GrownUps.com	(626)555-7265	78934 Hillside Dr.			
Boards & Toys Co.	(310)555-2373	4097 Douglas Av.	Glendale	CA	92561
Collectable Mini Designs Co.	(760)555-8146	361 Furth Circle	San Diego	CA	91217

**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.



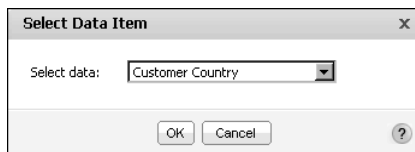
**Figure 2-24** Merging column headers

**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
Vitachrome Inc.	(212) 555-1500	2678 Kingston Rd.	NYC	NY	10022

**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.
- 2 Right-click the column header, then choose Row>Insert Row Above to add a row above the selected column header. Choose Row>Insert Row Below to add a row below the selected column header.

Select a column header, then right-click it to choose a row insertion option

Customer	Phone	AddressLine1	City	State	Zip
American Souvenirs Inc.		149 Spinnaker Dr.	New Haven	CT	97823
Cambridge Collectables Co.		4658 Baden Av.	Cambridge	MA	51247
Diecast Classics Inc.		7586 Pompton St.	Allentown	PA	70267
Gift Depot Inc.		25593 South Bay Ln.	Bridgewater	CT	97562
Land of Toys Inc.			NYC	NY	10022
Mini Gifts Distributors Ltd.			San Rafael	CA	97562
Mini Wheels Co.			San Francisco	CA	94217
Muscle Machine Inc.		4092 Furth Circle	NYC	NY	10022
Signal Gift Stores	(702) 555-1838	8489 Strong St.	Las Vegas	NV	83030

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

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 2-28** A report design displaying an empty row above 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 formatting as the row that follows.

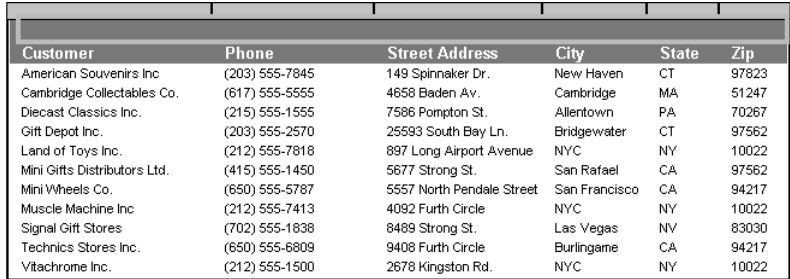
**How to merge column headers to type text that spans multiple columns**

- 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.

Customer	AddressLine1	City	State	Zip	
American Souvenirs Inc	149 Spinnaker Dr.	New Haven	CT	97823	
Cambridge Collectables Co.	4658 Baden Av.	Cambridge	MA	51247	
Diecast Classics Inc.	7586 Pompton St.	Allentown	PA	70267	
Gift Depot Inc.	25593 South Bay Ln.	Bridgewater	CT	97562	
Land of Toys Inc.		NYC	NY	10022	
Mini Gifts Distributors Ltd.		San Rafael	CA	97562	
Mini Wheels Co.		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 2-29** 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.



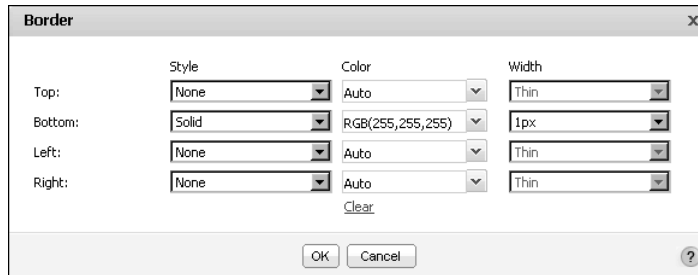
Customer					
American Souvenirs Inc	(203) 555-7845	149 Spinnaker Dr.	New Haven	CT	97823
Cambridge Collectables Co.	(617) 555-5555	4658 Baden Av.	Cambridge	MA	51247
Diecast Classics Inc.	(215) 555-1555	7586 Pompton St.	Allentown	PA	70267
Gift Depot Inc.	(203) 555-2570	25593 South Bay Ln.	Bridgewater	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** Results 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.
  - 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, one pixel wide. Choose **OK**.



**Figure 2-31** Adding borders around the merged column header



**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	9406 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	\$84,600	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** Default PDF document layout does not accommodate all the data

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

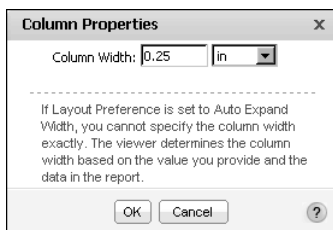
- Adjusting the widths of the columns
- Changing the layout type, page size, and orientation

## 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 width of other column affects the actual width.

### 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

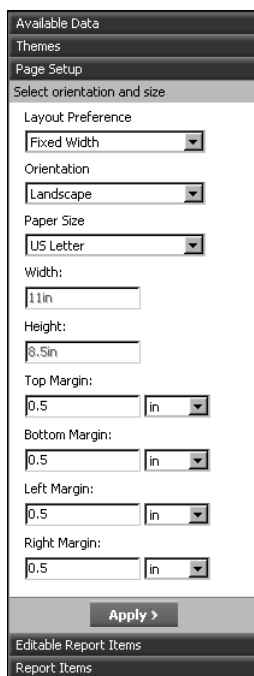
Choose OK.

## 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. Page Setup appears, as shown in Figure 2-36.



**Figure 2-36** 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.



# Inserting calculated data

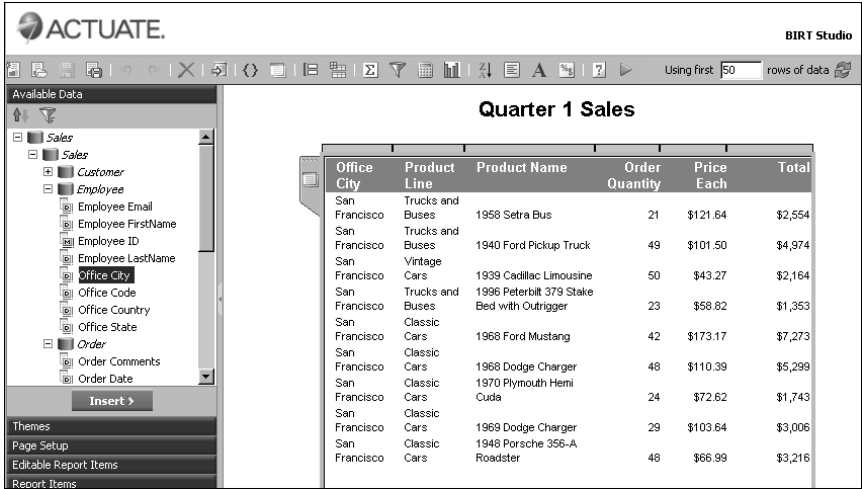
This chapter contains the following topics:

- 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 anticipated 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.



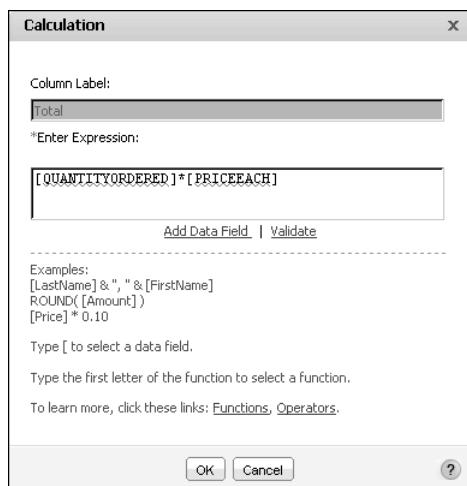
**Figure 3-1** A report design displaying a computed column, Total

# 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 shown 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 following expression specified for the computed column, Total:

[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** Calculation displaying 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:

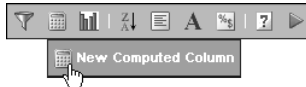
```
TRIM( [CustomerName] )
```

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

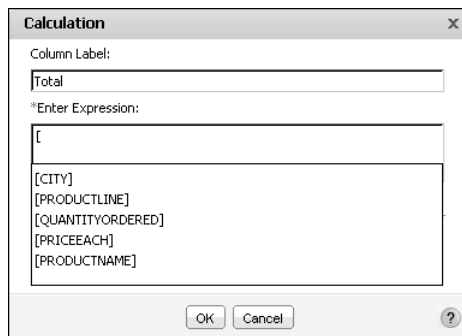
- 1 Select the table column to the left of the new computed column you want to insert.
- 2 On the toolbar, choose New Computed Column, as shown in Figure 3-3.



**Figure 3-3** Choosing New Computed Column

Calculation appears.

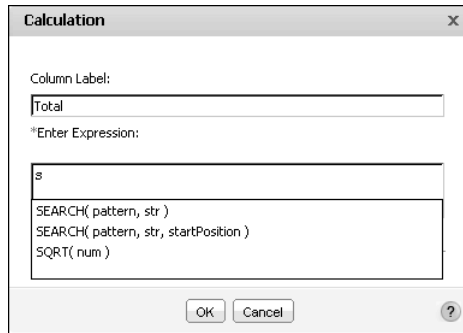
- 3 In Column Label, type a name for the computed column. The name you specify appears in the column header.
- 4 In Enter Expression, type the expression to calculate the values you want to display:
  - To use a data field in the expression, type the left bracket ([), then select a field from the list 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. This task is described later in this chapter.

- 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.
- 5 When you finish writing the expression, choose Validate. If the expression is syntactically correct, the dialog box displays the message, This Expression is valid. If the expression contains an error, the dialog box displays a message that describes the error.
  - 6 After you validate the expression, Choose OK. The computed column appears in the report design.



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

## Creating aggregate data in a computed column

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

---

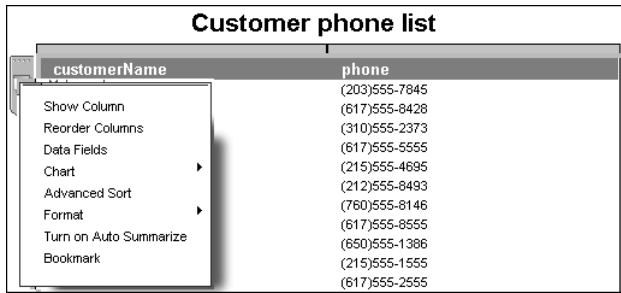
## Adding data fields for an expression

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.

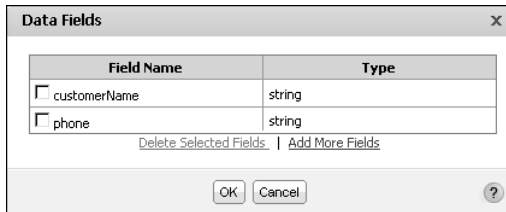
For example, if you insert a CustomerName field and a Phone field in a table, then you insert a computed column, the computed column only has access to the CustomerName and Phone fields. You cannot create an expression that combines, for example, values from the AddressLine1, City, State, and PostalCode fields. To create such a computed column, you must first add the data fields to the table, but 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.
- 2 Choose Data Fields. The Data Fields dialog box appears. Figure 3-7 shows an example of Data Fields displaying two fields, customerName and phone, currently used in the table.



**Figure 3-6** Adding a data field to a 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.

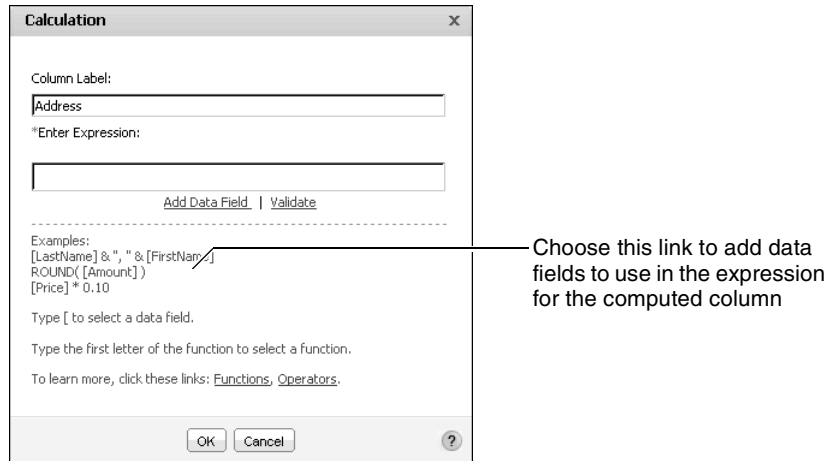


**Figure 3-8** Data Set displaying all the available fields

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

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 to 4 to add fields.



**Figure 3-9** Choose Add Data Field to add fields

## 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 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 net 30. 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 that 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:

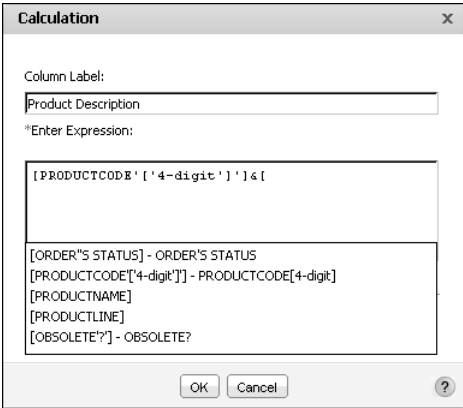
```
[  
]  
?  
' (single quotation mark)
```



If the name of a data field contains reserved characters, BIRT Studio removes the reserved characters when you select the data field to use in an expression. 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 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 third, fourth, 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['4digit']] - PRODUCTCODE[4-digit]
[OBSOLETE?'] - OBSOLETE?
```



**Figure 3-10** Calculation displaying a list of available data fields



# 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** Report designs displaying unsorted and sorted data

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.

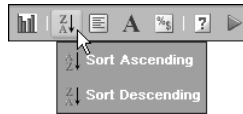
You can sort data in ascending or descending order, and you can sort data by multiple columns. For example, Figure 4-2 shows the results of sorting by country, then by customer name. For rows where the country names repeat, France and USA, the customer names appear in alphabetical order.

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** A report design 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

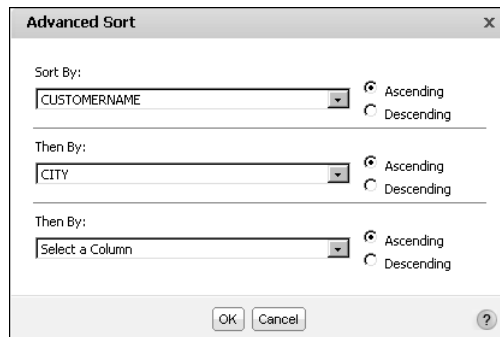
## Sorting on multiple columns

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.

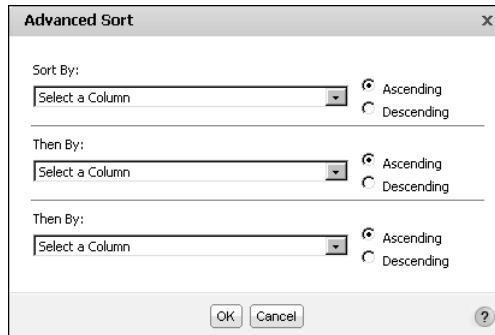


**Figure 4-4** Specifying advanced sorting

## 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** A report design 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 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 Fr			\$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 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.

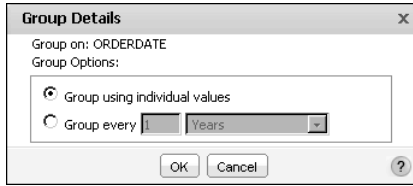
Shipment Date	Order	Shipped To
January, 2003		
	10100	Online Diecast Creations Co.
	10101	Blauer See Auto, Co.
	10102	Vitachrome Inc.
February, 2003		
	10103	Baane Mini Imports
	10104	Euro+ Shopping Channel
	10105	Danish Wholesale Imports
	10106	Rovelli Gifts
	10107	Land of Toys Inc.
March, 2003		
	10108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

**Figure 4-9** Comparing different date grouping options

### 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

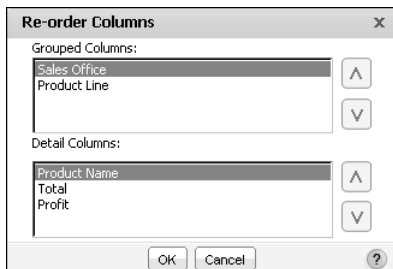
Just as you do when you sort on multiple columns, think about the order of precedence before you create groups. If you want to group customer data by state, then by city, create the groups in that order. The report design shown in Figure 4-7 groups sales data first by sales office, then by product line. 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.

Sales office group value

Product line group values

Sales Office	Product Line	Product Name	Total	Profit
San Francisco	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 Bus		\$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

Column headers

Detail rows

**Figure 4-13**      Using groups to organize data

Sales Office	San Francisco	
Product Line	Classic Cars	
Product Name	Total	Profit
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
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
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	\$34,547.42 \$13,247.66
Sum		\$34,547.42 \$13,247.66

**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 from 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 grouped.

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	\$8,880.80	\$3,613.70
Vintage Cars	1939 Cadillac Limousine	\$2,163.50	\$1,006.50
	Sum	\$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. In the sales office section heading, for example, you can add the address of the sales office or the e-mail of the sales representative. Figure 4-18 shows an example.

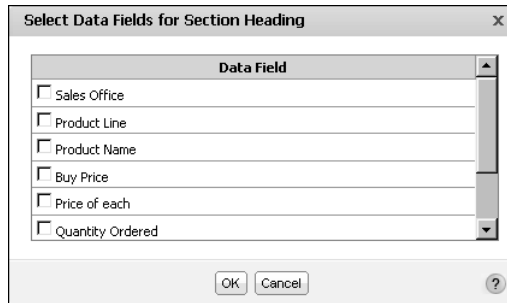
Sales Office	San Francisco
Email	ljennings@classic

**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 currently used in the report design.
- 3 Select the data field to add to the section heading, as shown in Figure 4-19. Choose OK. The data field appears in the section heading.

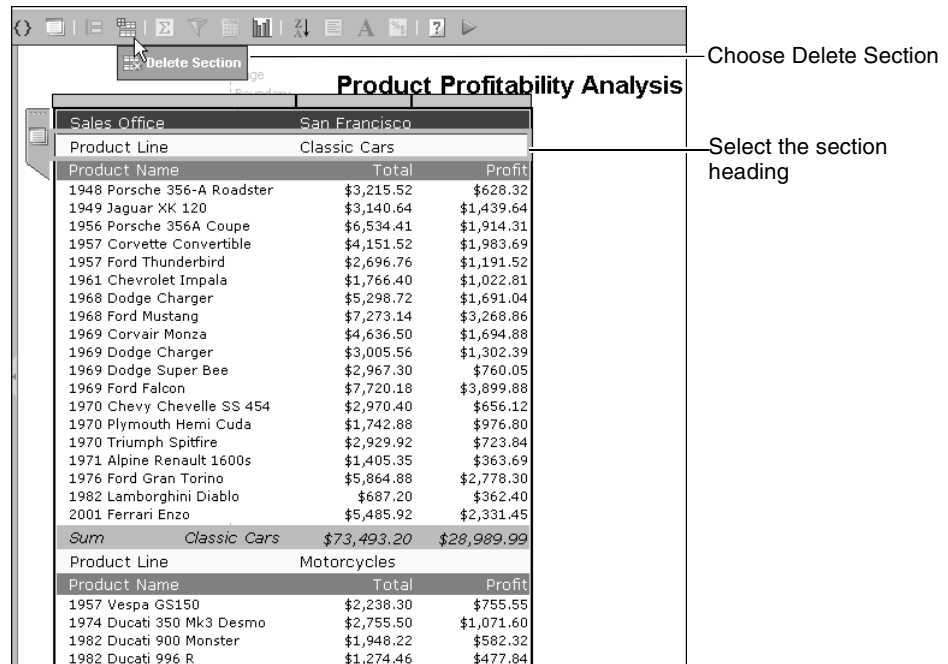


**Figure 4-19** Select Data Fields for Section Heading displaying fields that you can add to a section heading

## 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 4-20** Deleting a section

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	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
	1939 Cadillac Limousine	\$2,163.50	\$1,006.50
Sum	San Francisco	\$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.

- 3 In Select Function, select the aggregate function to use.
- 4 In Aggregate on:
  - 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.

The screenshot shows the 'Aggregation' dialog box in BIRT Studio. It is used to configure how data is aggregated. The 'Selected Column' is 'Profit' and the 'Select Function' is 'Sum'. Under 'Aggregate on', both 'table' and 'group' are selected, with 'header' and 'footer' options also available. The 'Enter Label' field contains 'Sum', and the label 'Sum(Profit)' is displayed. There are buttons for 'Delete Aggregation', 'Add Aggregation', 'OK', 'Cancel', and a help icon. A link to 'Functions' is also present.

**Figure 4-24** Aggregating data

Choose OK. Figure 4-25 shows two aggregations in addition to the one shown in Figure 4-24, defined for the report design in Figure 4-23.

## 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.

Selected Column: Profit  
Select Function: Max

Aggregate on: ☒ table ☐ header ☐ footer  
☒ group ☐ header ☐ footer

Max(Profit) Enter Label: Max

Selected Column: Profit  
Select Function: Average

Aggregate on: ☒ table ☐ header ☐ footer  
☒ group ☐ header ☐ footer

Average(Profit) Enter Label: Average

Add Aggregation

To learn more, click the link to [Functions](#).

OK Cancel ?

**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.

Sales Office		San Francisco	
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	
2002 Yamaha YZR M1	\$3,472.48	\$1,969.00	
Sum	Motorcycles	\$11,688.96	\$4,856.31
Max	Motorcycles	\$3,472.48	\$1,969.00
Average	Motorcycles	\$2,337.79	\$971.26
Sum	San Francisco	\$11,688.96	\$4,856.31
Max	San Francisco	\$3,472.48	\$1,969.00
Average	San Francisco	\$2,337.79	\$971.26
Sum		\$11,688.96	\$4,856.31
Max		\$3,472.48	\$1,969.00
Average		\$2,337.79	\$971.26

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

- 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.

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, while improving response time, and also reducing the load on the server. For more information, see Chapter 6, “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 within each product-line section.

### How to hide the details of a group

- 1 Select the grouped column for which to hide details.
- 2 Right-click the column, 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.

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

**Figure 4-27** Showing and hiding detail rows

Product Line grouped column selected

Sales Office	San Francisco		
Product Line	Product Name	Total	Profit
Classic Cars	1956 Porsche 356A Coupe	\$6,534.41	\$1,914.31
		\$5,298.72	\$1,691.04
		\$7,273.14	\$3,268.86
		\$5,864.88	\$2,778.30
Sum		\$24,971.15	\$9,652.51
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
Sum		\$8,216.48	\$2,887.31
Sum		\$33,187.63	\$12,539.82
Sum		\$33,187.63	\$12,539.82

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

### How to hide the details of a section

- 1 Select the section heading of the section whose details to hide.

- 2 Right-click the section heading, then choose **Section** ➤ **Hide Detail**, as shown in Figure 4-29.

Sales Office section heading selected

Section	Chart	Sort	Section Heading	Total	Profit
Sales Office				15.52	\$628.32
Classic Cars				\$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
Sum	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
Sum	Motorcycles			\$8,216.48	\$2,887.31
Sum	San Francisco			\$63,173.66	\$24,560.17
Sum				\$63,173.66	\$24,560.17

**Figure 4-29** Hiding the details of a selected section

## 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**.

## 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** Setting a 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, choose one of the following options:
    - 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, choose one of the following options:
    - 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.





# 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 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 4/1/2008 and 6/30/2008

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.

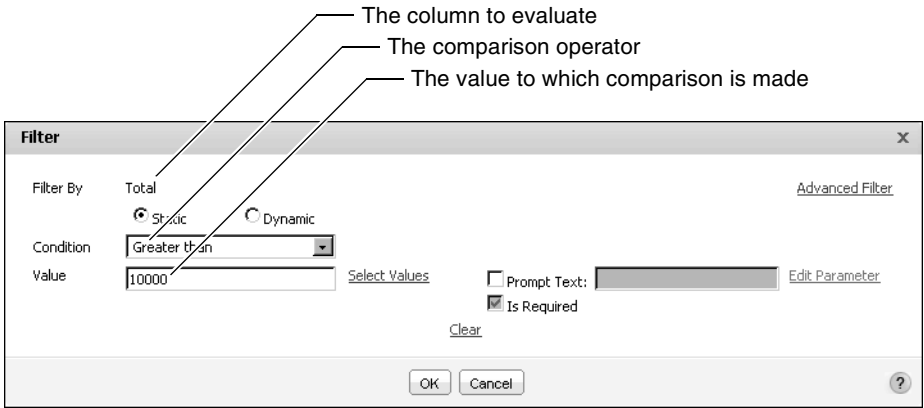


Figure 5-1 Filter displaying a filter condition

## Creating 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.

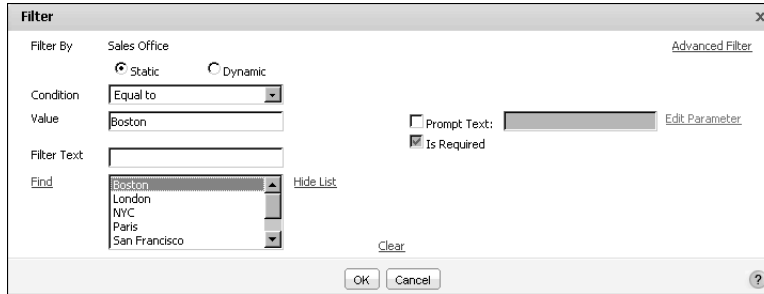
The Filter dialog box is shown. It has a title bar 'Filter' with a close button. Inside, 'Filter By' is set to 'Sales Office'. There are radio buttons for 'Static' (selected) and 'Dynamic'. The 'Condition' dropdown is set to 'Equal to'. Below it is a 'Value' field. To the right of the 'Value' field is a 'Select Values' link. Further right are checkboxes for 'Prompt Text:' (unchecked) and 'Is Required' (checked). There is a 'Clear' button below these checkboxes. On the far right is an 'Advanced Filter' link. At the bottom are 'OK' and 'Cancel' buttons, and a help icon (?) on the right.

**Figure 5-2** Filter displaying the name of a selected column

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

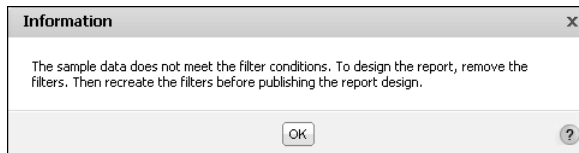
- 3 Select Static, as the type of filter, if necessary.
- 4 In Condition, select the comparison test, or operator, to apply to the selected column. Depending on the operator you select, Filter displays one or two additional fields, or a completed filter condition.
- 5 If you select an operator that requires 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 highlights this value in the list.
- 6 Choose OK. The filter takes effect. In some cases, the sample data used 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 shown in Figure 5-4.

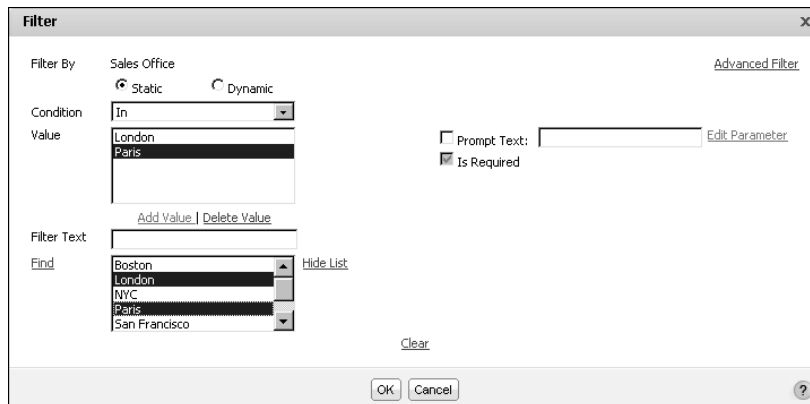


**Figure 5-4** The message displayed when the sample data in the report design does not meet the filter condition

## 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.

Figure 5-5 shows the selection of London and Paris from a list of sales office values.



**Figure 5-5** Selecting multiple comparison values

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.

## 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 Equal To 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:

```
e-mail Is Not Null
```

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

```
e-mail Is Null
```

## Displaying top or bottom values

For a report that presents a large amount of numeric data, you 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 100 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 are displayed.

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.

Filter

Filter By: Profit

Static (selected) Dynamic

Filter on: Entire Table On Group (selected) CITY\_25

Condition: Bottom Percent

Value: 10

Select Values

Prompt Text: Edit Parameter

Is Required

Clear

OK Cancel

**Figure 5-6** Specifying a Top/Bottom N 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

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

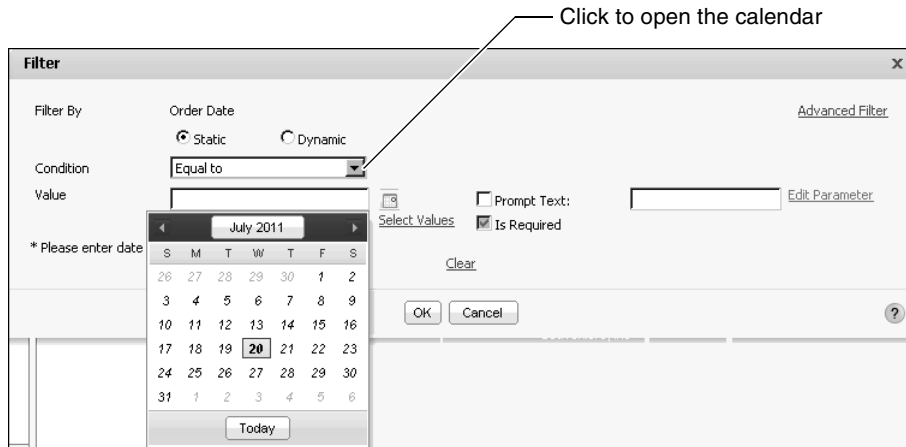
## Specifying a date as a comparison value

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\\\\%
```

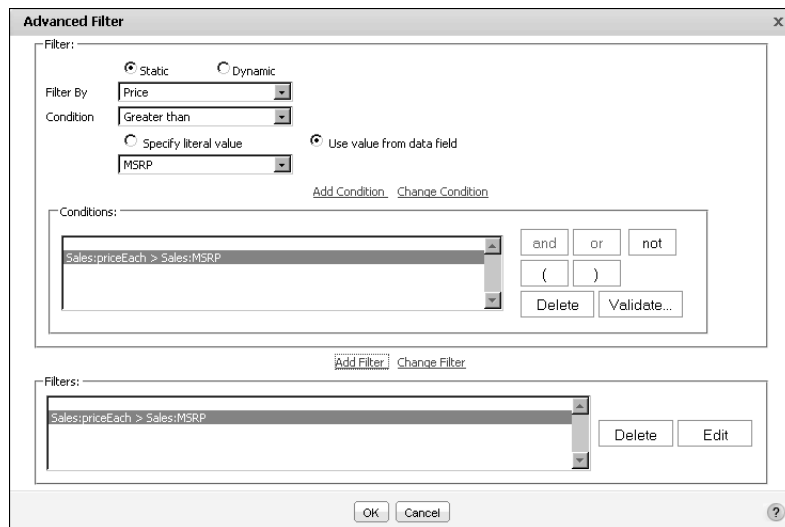
## Comparing to a value in another column

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), you can create a filter condition to compare the Price and MSRP of each product and display only 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.
- 2 On Filter, choose Advanced Filter. Advanced Filter appears, as shown in Figure 5-9.



**Figure 5-9** Building a comparison filter using 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.

Figure 5-9 shows an example of a filter condition that compares the values in the Price column with the values in the MSRP column.

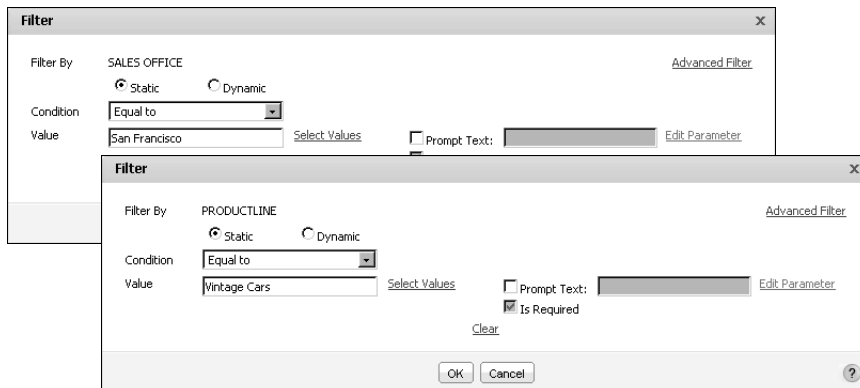
- 6 Choose Add Condition.
- 7 Choose Add Filter, then choose OK.

## 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.

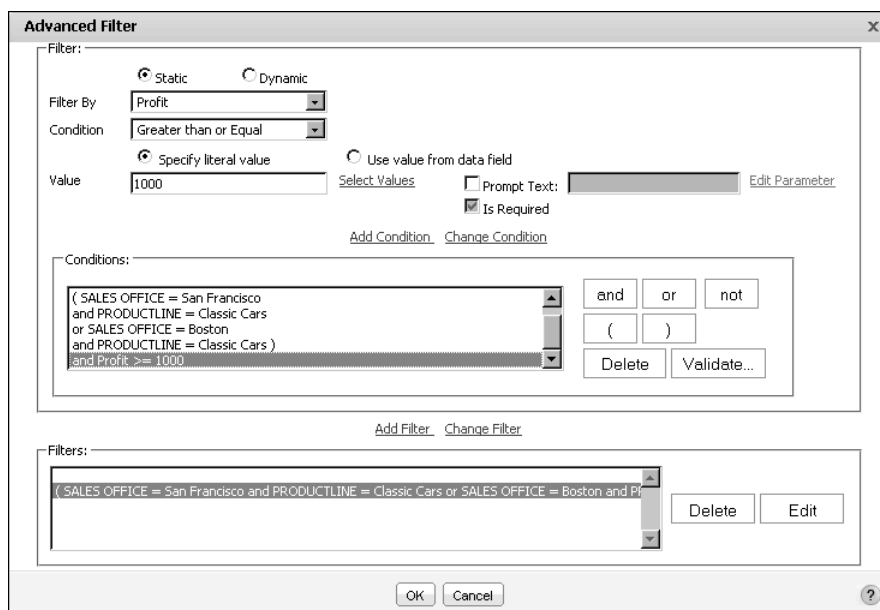


**Figure 5-10** Creating multiple filter conditions

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. If you need to create a filter to return data rows when either condition is true, or you need to create a more complex filter, use Advanced Filter, shown in Figure 5-11.



**Figure 5-11** Displaying multiple conditions in a complex filter

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.

### How to define multiple filter conditions



- 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.
  - 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:

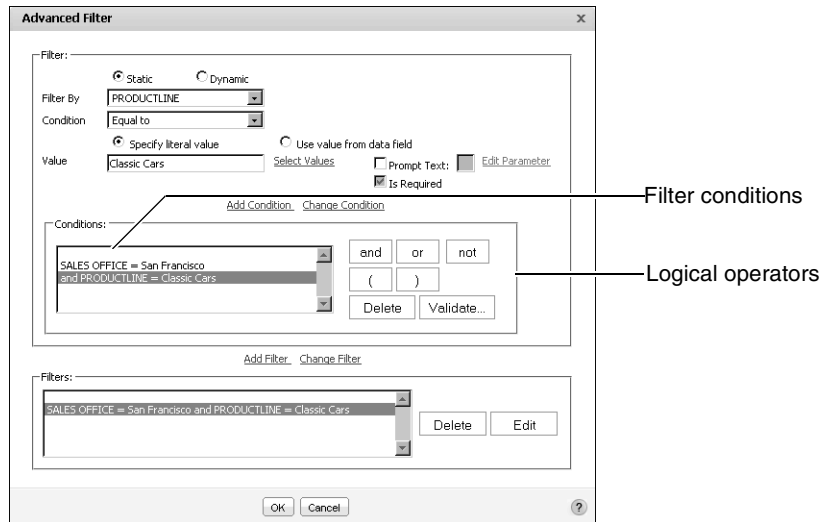
- 1 In Filter By, select another column.
- 2 In Condition, select a comparison operator.
- 3 Specify a value.
  - ❑ To type or choose a specific value, choose Specify a literal value.
    - ❑ In Value, type the value. Proceed to step 4.
    - ❑ To select from a list of values, choose Select Values. A 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 results you expect.

## 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, sold by 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
	San Francisco	Classic Cars	1952 Alpine Renault 1300	\$10,266.40
1972 Alfa Romeo GTA			\$8,616.96	\$3,133.44
Motorcycles		2003 Harley-Davidson Eagle Drag Bike	\$9,394.28	\$4,934.30
Sydney	Motorcycles	2003 Harley-Davidson Eagle Drag Bike	\$9,299.71	\$4,839.73
London				

**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)
```

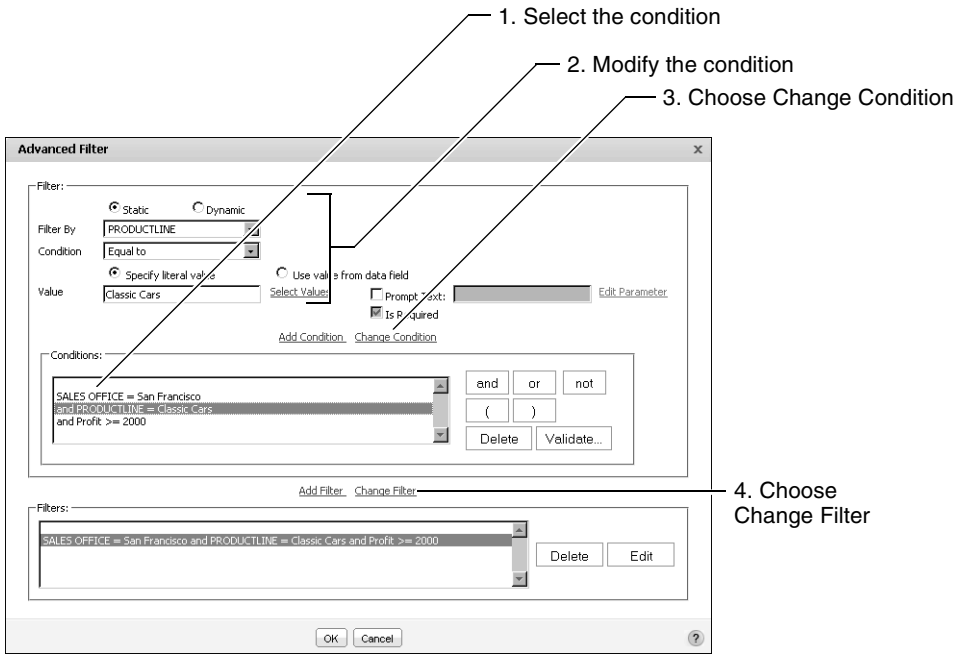
Figure 5-14 shows the generated report. This time, only two rows meet the conditions. The Sales Office = San Francisco and ProductLine = Classic Cars conditions must be true, and either the Total >= 8000 condition or the Profit >= 4500 condition is true.

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.



**Figure 5-15** Changing a filter condition in Advanced Filter

### Deleting a filter condition

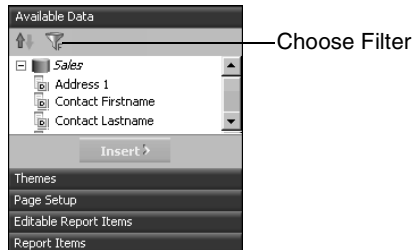
To delete a filter condition, in the Filters area of Advanced Filter, select the Filter, and choose Delete. Then, select the Condition and choose Delete. Choose OK. Verify that any remaining filter conditions display the expected results.

## Creating a filter at the data-set level

You create a filter at the data-set level using Advanced Filter.

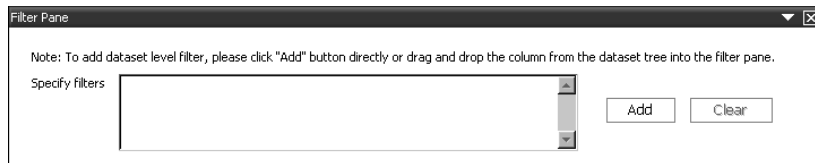
### How to create a filter at the data-set level

- 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 yet defined any filters at the data-set level.



**Figure 5-17** Filter Pane

- 2 Choose Add to create a new filter. Advanced Filter appears.
- 3 Define the filter condition. For more information about how to define a basic filter condition in Advanced Filter, see “How to define multiple filter conditions.”

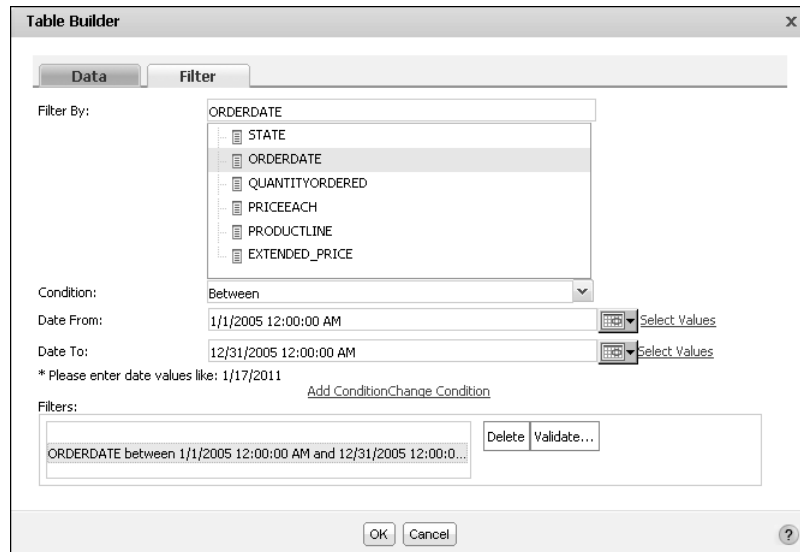
### How to create a data-set filter using Table Builder

- 1 Create a new report using a data set in a template, or a custom data source.
- 2 On Table Builder—Data, select the data fields to use in the report design.
- 3 Choose Filter. On Table Builder—Filter, to create a filter condition, complete the following steps:
  - 1 In Filter By, select a column from the drop-down list.
  - 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.



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

### 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.

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.

### 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.

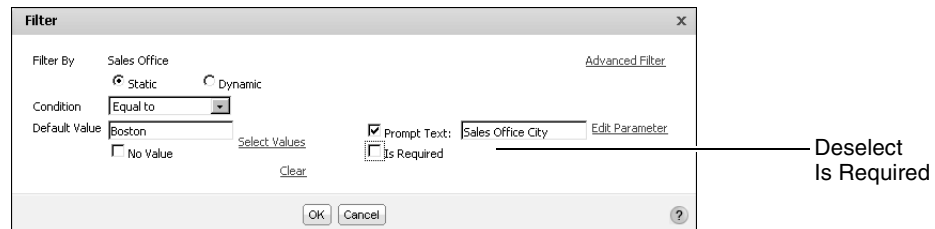
In BIRT Studio, a dynamic filter parameter is always optional.

### 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 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 as 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 data source. 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.

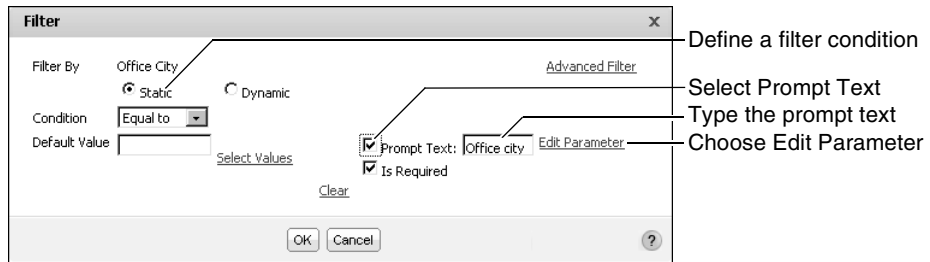
## Creating a static filter parameter

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.

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



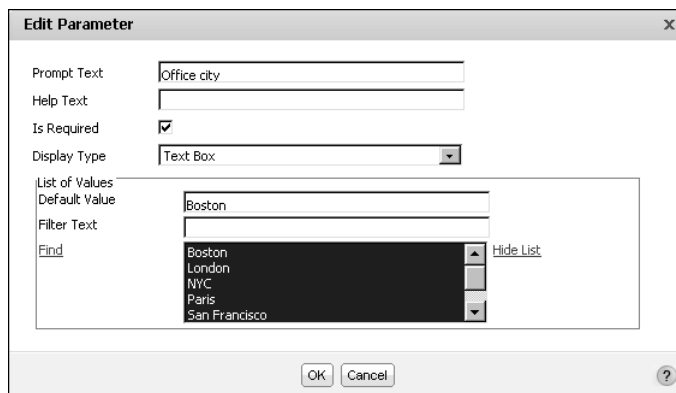
- 1 In BIRT Studio, select the column that contains the values for which you want to create a static prompt, then choose Filter.
- 2 Filter appears. In Filter, select Static, if necessary.
- 3 In Condition, select an operator.
- 4 Select Prompt Text, then type the text to display.
- 5 Choose Edit Parameter, as shown in Figure 5-20.



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

**6** Edit Parameter appears. On Edit Parameter, complete the following steps:

- Prompt Text displays the text you entered in step 4.
- In Help Text, optionally type a Tooltip to assist the user in selecting the values for which BIRT Studio displays data.
- Accept the default selection of Is Required. To make the parameter optional, deselect Is Required.
- In Display Type, select an option from the list. The example in Figure 5-21 shows the display type set to Text Box.
- 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.
- 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. Choose OK.



**Figure 5-21** Creating a static parameter

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

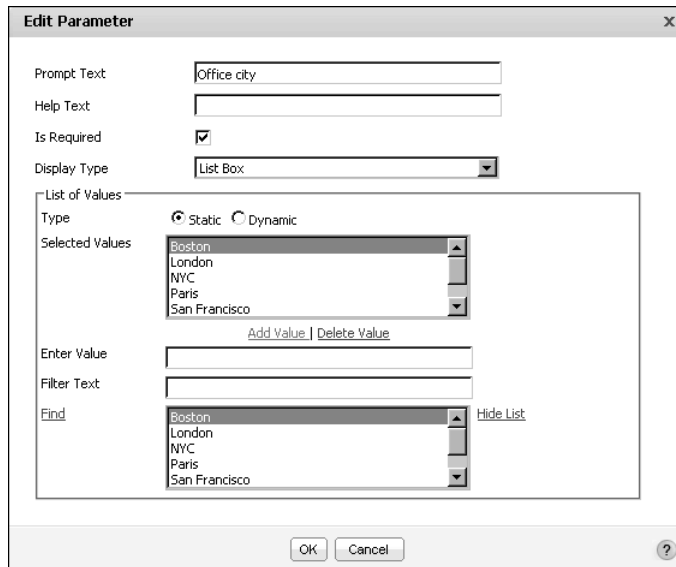
**Figure 5-22** Filter displaying the specified parameter properties

### 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.
- 5 In List of Values:
  - Select Static to create a static list of values, that the report developer specifies. If you select Static, proceed to step 7.
  - 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.

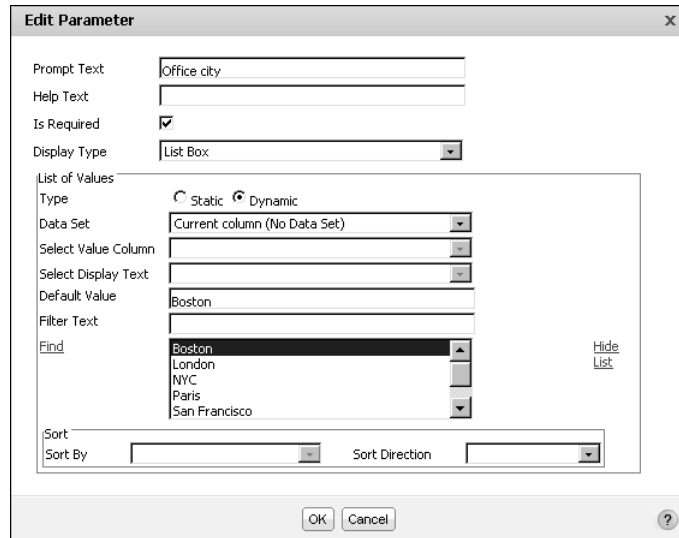
Select a value from the list of values that appears in Find, then press Ctrl and select any additional values you want to display. You can also type each value in Enter value, then choose Add value. The added values appear in Selected Values.



**Figure 5-23** Creating a static list of 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)**, or select an available data set. If you selected the current column complete the following steps 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, complete the following steps:

- 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.

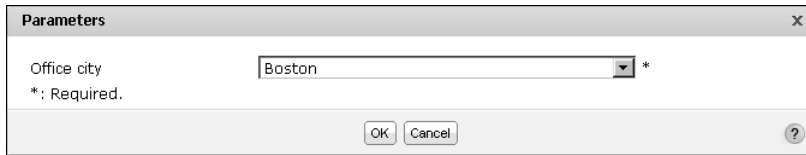
If you do not specify a sort direction, BIRT Studio displays the default value first, and arranges the remaining values in ascending order below the default value.

Choose OK.

**10** On Filter, choose OK.




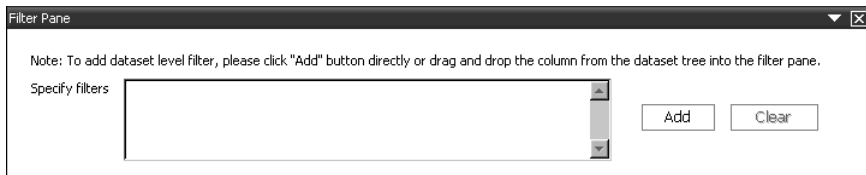
**11** Test the prompt in BIRT Studio by choosing Parameter. Parameters displays the prompt, as shown in Figure 5-25. In this example, the default parameter value is Boston.



**Figure 5-25** Parameters displaying the specified prompt

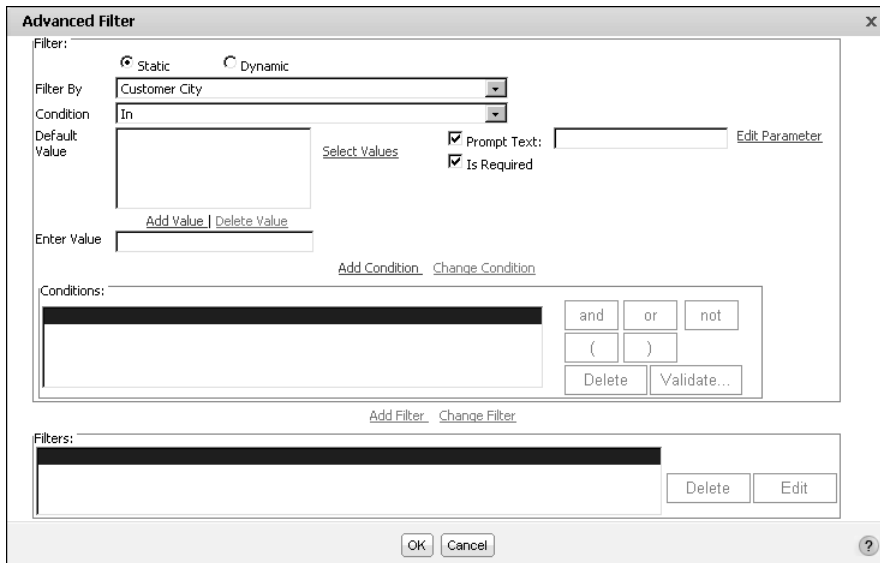
#### How to create a static filter parameter at the data-set level

- 1 Create a new report by selecting a template containing a data set.
- 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. Advanced Filter appears.



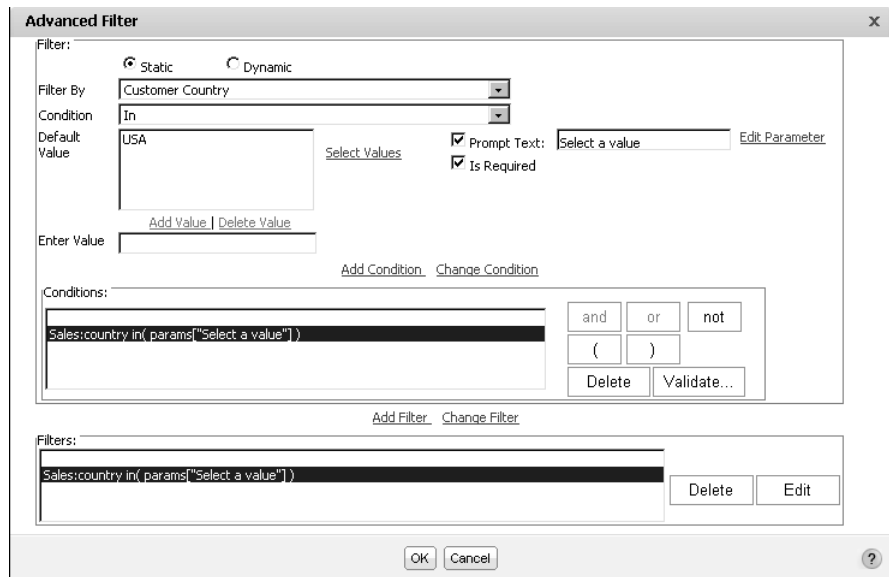
**Figure 5-26** Filter Pane

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



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

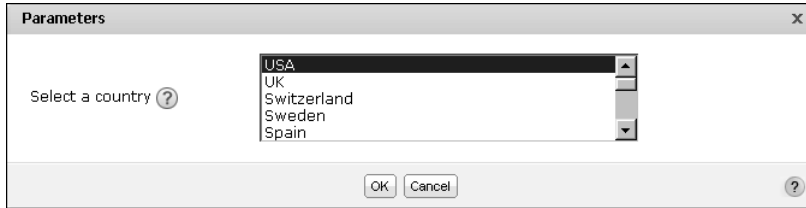
- 6 In Condition, select an operator from the drop-down list.
  - 7 Select Prompt Text, then type the text to display. Edit Parameter is highlighted, and Is Required appears selected, as shown in Figure 5-27.
  - 8 Choose Edit Parameter. In Edit Parameter, create the parameter condition as follows:
    - 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.”
    - To create a static filter parameter displaying a text box where the user types the value, complete step 6 in “How to create a static filter parameter at the report-table level.”
- 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.

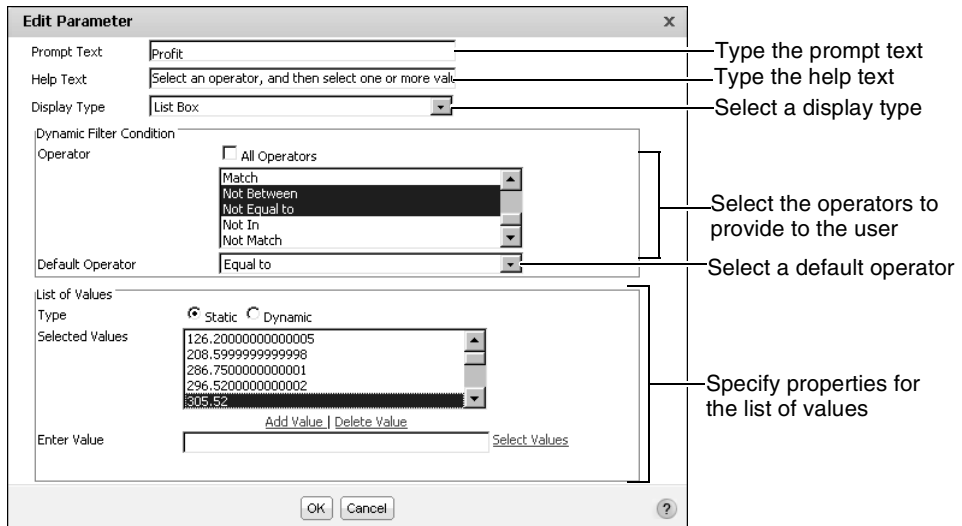


**Figure 5-29** Parameters displaying the static parameter condition

## 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.

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, as shown in Figure 5-30.



**Figure 5-30** Specifying properties for a dynamic filter parameter

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.
- 3 In Prompt Text, type a description for the column you selected in step 1. Then, choose Edit Parameter.

**Figure 5-31** Specifying filter parameter properties

- 4 In Edit Parameter, Prompt Text displays the text you typed in step 4. Complete the following steps:
  - 1 In Help Text, optionally type a Tooltip that assists the user in selecting parameter values, for which BIRT Studio displays data.
  - 2 In Display Type, choose List Box.
  - 3 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.
  - 4 Select a default operator from the drop-down list.
  - 5 In List of values:
    - 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 6:
    - Select Dynamic to create a dynamic list. A dynamic list contains values that are generated at run time from the data source. If you selected Dynamic, proceed to step 8:
  - 6 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.

- 7 In Selected values, select a value to set as the default.  
Choose OK. Proceed to 5.
- 8 In Data Set, select Current Column (No data set), or the available data set. If you selected the current column, complete the following steps:
  - ❑ 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 5.
- 9 In Data Set, if you selected an available data set, complete the following steps:
  - ❑ 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

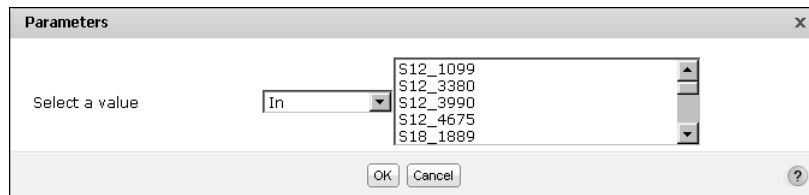
- ❑ In Default value, because the parameter is optional, do not supply a default value.
- ❑ In Sort, select a sort field, by which the data in the list of values is sorted. In the example shown in Figure 5-32, displayed values are sorted by Product Line.
- ❑ In Sort Direction, select Ascending or Descending. 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.



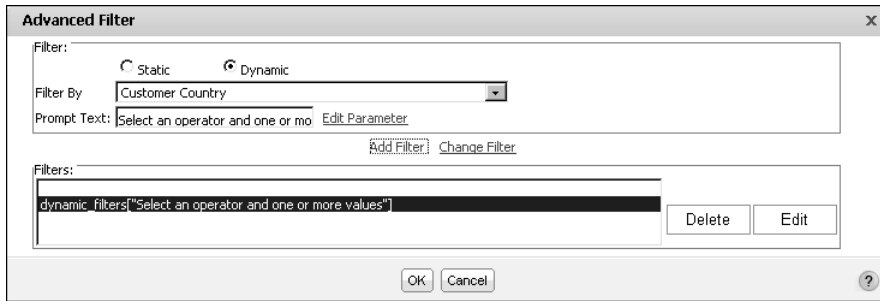
**Figure 5-33** Parameters displaying a dynamic prompt

### 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, or a custom data source.
- 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.
- 6 In Filter By, select a column from the drop-down list.
- 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.”

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

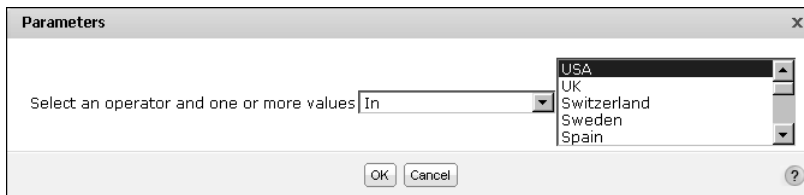
Choose Add Filter. The dynamic filter parameter condition appears in Filters, as shown in Figure 5-34. Choose OK. Filter Pane displays the condition.



**Figure 5-34** Advanced Filter displaying the dynamic filter parameter



- 8 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

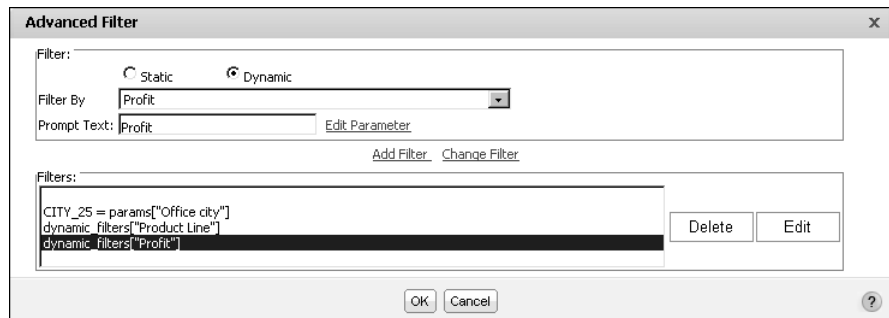
A report developer can also define multiple conditions, for which a user specifies values, further narrowing the scope of data displayed in a report. For example, you can define the first condition such that the user specifies a Sales Office value, the next condition where the user specifies one or more 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 9 in “How to create a dynamic filter parameter at the report-table level.”
  - 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.”
      - If you selected Dynamic in step 1, complete steps 1 to 9 in “How to create a dynamic filter parameter at the report-table level.”
    - 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

# 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. 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 6-1 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 6-1, 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 6-1 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 6-2 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 6-2 Summary table without groups displaying the grand total

The example shown in Figure 6-3 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 sub totals are not displayed. The classification of data fields into dimensions, attributes, and measures is defined in the data set.

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 6-3 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 a data source.
- 3** Select the dimension, attribute, and measure fields to use in the summary table.

As you did when creating a detail report, you do one of the following:

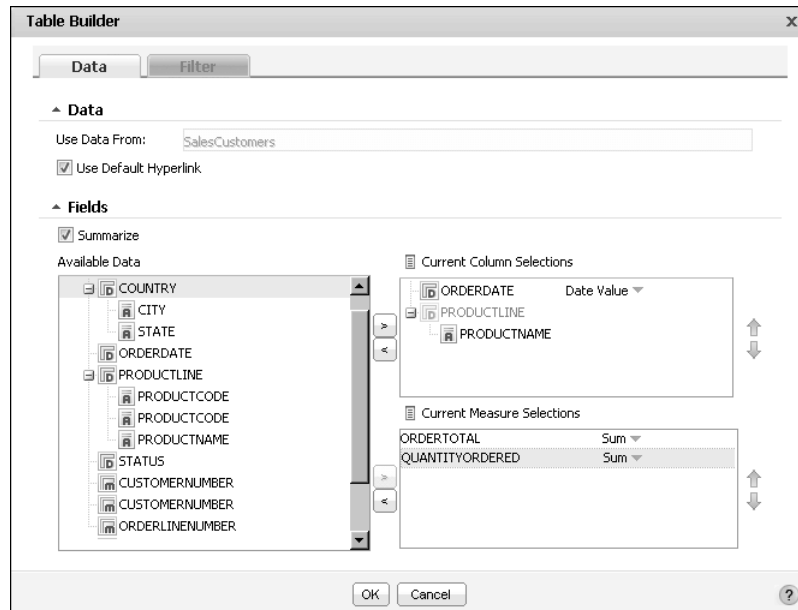
- Use Table Builder to select the data fields for the summary table. Using Table Builder, you can also specify the following information:
  - Group date-and-time dimension columns by an interval.
  - Specify one or more aggregate functions to use for the selected measure columns.
  - Optionally create a filter condition at the data-set level for a dimension or attribute column to limit the data displayed in the summary table.
- Manually select and insert data fields to create a summary table.

### How to create a summary table using Table Builder

This section describes the process to create a summary table using a data set in a template.

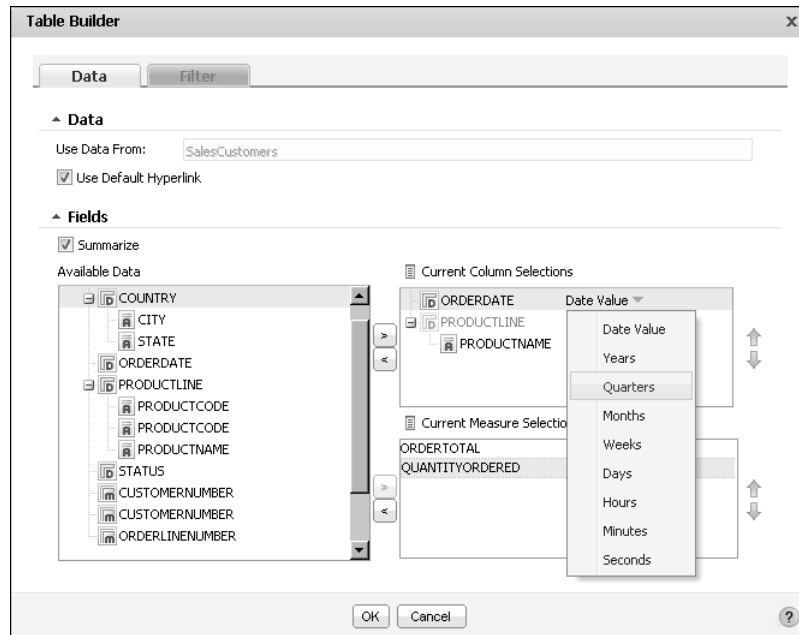
- 1** Launch BIRT Studio, select a template, then select a data set.
- 2** Table Builder appears. In Table Builder, complete the following steps:
  - 1** In Table Builder–Data, in Fields, select Summarize to create a summary table. Current Measure Selections is highlighted.
  - 2** In Fields, in Available 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 6-4. 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.

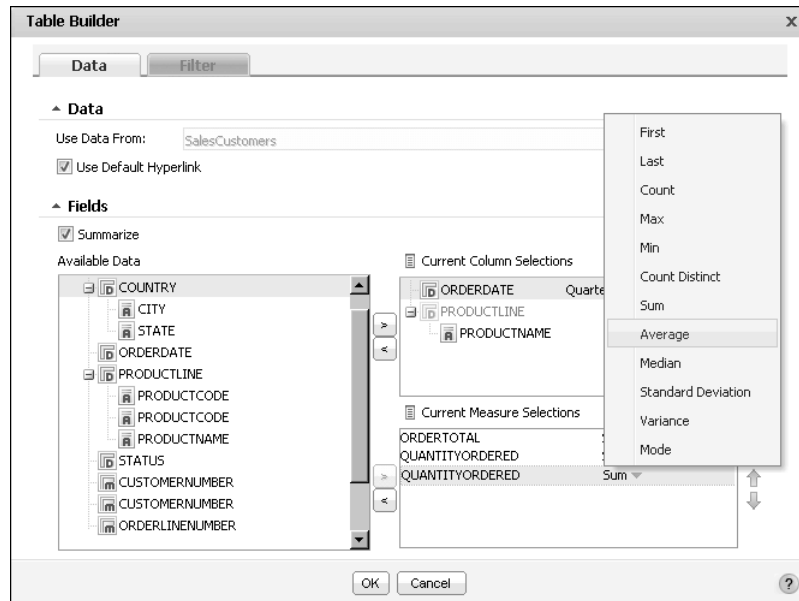


**Figure 6-4** Table Builder displaying the selected data fields

- 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 6-5 shows the Order Date field grouped in Quarters.
- 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 6-6 shows the average function applied to the Quantity Ordered field.



**Figure 6-5** Grouping date dimension column in quarterly intervals



**Figure 6-6** Aggregating measure columns




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. Choose OK. BIRT Studio displays the selected data fields in a summary table, as shown in Figure 6-7. 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 )		\$99,981	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
	1936 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 6-7** Summary table report design

### How to create a summary table using an ODA data source

The following section describes the process to create a summary table using an ODA data source.

- 1 In BIRT Studio, create a new report by selecting a template and selecting the data source to use.
- 2  Choose Turn on Auto Summarize.
- 3 In Available Data, select the dimension, attribute, and measure columns to use, then choose Insert.

A summary table that displays aggregate data for the selected columns appears in the report design. You are now ready to modify the summary table.

## 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 6-1 lists the

actions you can perform on dimension columns, measure columns, attribute columns, column headers, and the entire report table.

**Table 6-1**      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	✓	✓	✓		
Edit the column header text				✓	
Filter Top/Bottom N			✓		
Format data	✓	✓	✓		
Group date-and-time data in intervals	✓				
Hide column					
Hide details for a group	✓				
Insert a chart		✓			✓
Reorder columns	✓	✓	✓		

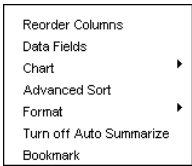
**Table 6-1**      Modifying data in a summary table

Action	Dimension column	Attribute column	Measure column	Column header	Report table
Sort column data	✓	✓	✓		✓
Specify alignment properties	✓	✓	✓	✓	✓
Specify column width	✓	✓	✓		
Specify conditional formatting rules			✓		
Specify font properties	✓	✓	✓	✓	✓

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 perform any of these actions, choose an option from the context menu of the report table, shown in Figure 6-8.



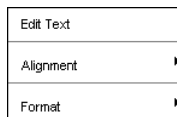
**Figure 6-8**      Modifying the report table

## 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 6-9.



**Figure 6-9** Modifying a column header

## Working with a dimension column

If there are one or more dimension columns in a report, 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.

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

Inner-most grouped date dimension

**Figure 6-10** Grouping the inner-most date-and-time dimension column

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 6-10 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 6-10, and subtotals are displayed for each value in the grouped 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 6-11. 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

Inner-most ungrouped dimension

**Figure 6-11** Grouping dimension columns

The innermost dimension column does not display subtotals, irrespective of the type of data in the column, as shown in Figure 6-10 and Figure 6-11.

When a summary table contains a single dimension column, and one or more associated attribute columns, you cannot group the data in the dimension column, and no subtotals are displayed as shown in Figure 6-12.

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
	1938 Cadillac V-16	
Vintage Cars	Presidential	
	Limousine	\$27,165
Grand Total		\$145,263

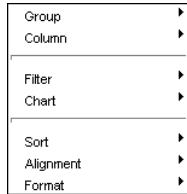
**Figure 6-12** A summary table displaying a grand total

### 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 of the context menu for a dimension column, shown in Figure 6-13, displays the available options.



**Figure 6-13** 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 data set.

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 6-14 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 6-14** 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 6-15 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.

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 6-16.

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 6-15** Attribute and measure columns

PRODUCTNAME	STATUS	CITY	SUM (QUANTITYORDERED)
1965 Aston Martin DB5	Shipped		
		Nantes	156
		Las Vegas	611
Sub Total ( Shipped )			767
Sub Total ( 1957 Ford Thunderbird )			767

New Attribute

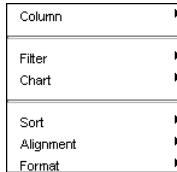
Dimension

Attribute

**Figure 6-16** 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 6-17.



**Figure 6-17** 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 in the data set. You can modify the default aggregate function and add additional aggregate functions to data in measure columns.

## Inserting a measure column

A measure column is typically inserted to the right of dimension and attribute columns in a summary table. When you insert a measure column in a summary table, aggregate data is displayed based on the default aggregation function set by developer at the time of creating the data set. You can modify the aggregate function and add additional aggregation functions using BIRT Studio.



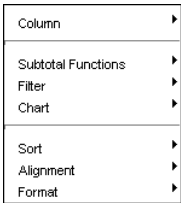
## Modifying a measure column

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 a column, and choose an option from the context menu shown in Figure 6-18.

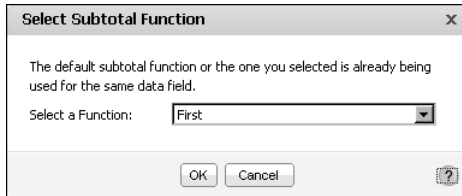


**Figure 6-18**      Modifying a measure column

## 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 Select Subtotal Function appears, as shown in Figure 6-19. Select an aggregate function from the drop-down list.



**Figure 6-19** Adding an aggregate function

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 data values

You can format aggregate data values displayed as sub totals, or grand totals as follows:

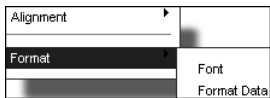
- Specify the alignment properties.
- Specify a font type, color, or style.
- Specify an available or custom data format to the numeric data values.

To modify alignment options for the aggregate value, right-click the data cell, and choose Alignment→Align Left, Align Right, or Align Center from the context menu.

To modify font properties for the aggregate value, right-click the data cell, and choose Format→Font from the context menu.

To apply a number format to the aggregate data value, right-click the data cell, and choose Format→Format Data from the context menu.

Figure 6-20 displays an example of the context menu that appears when you right-click an aggregate data cell.

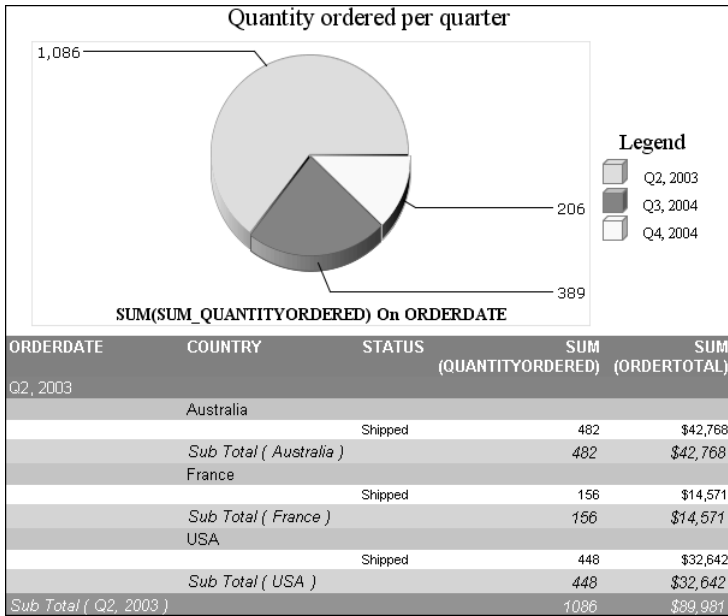


**Figure 6-20** Choosing formatting options from the context menu

## Inserting a chart in a summary table

You can insert 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 outer-most 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 6-21 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.



**Figure 6-21** A summary table displaying a pie chart

For more information about using charts, see Chapter 8, “Presenting data in a chart.”

---

## Changing table types

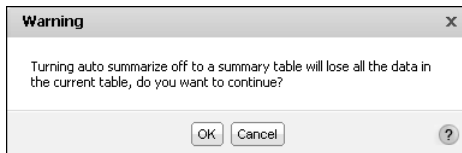
In BIRT Studio, you can convert a summary table to a detail table, or a detail table to a summary table. When you change table types, the data in the original report table must be re-created by re-inserting the selected data fields.

### 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 6-22. Choose OK.



**Figure 6-22** Warning dialog

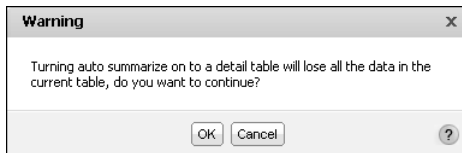
- 2 Choose Insert to re-insert the selected fields to create a new report design. BIRT Studio displays the chosen data fields in a detail table.

### 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 6-23. Choose OK.



**Figure 6-23** Warning dialog

- 2 Choose Insert to re-insert the selected fields to create a new report design. BIRT Studio displays the chosen data fields in a summary table.

# 7

## **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 is similar in appearance to a spreadsheet. 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 7-1 shows a cross tab that displays sales totals by state and by product line.

Total sales by product line and state						Row area Column area
	Classic Cars EXTENDED_PRICE	Motorcycles EXTENDED_PRICE	Trucks and Buses EXTENDED_PRICE	Vintage Cars EXTENDED_PRICE	Grand Total EXTENDED_PRICE	Detail area
CA	43912.8799999999966	11688.9600000000001	8880.8	2163.5	66646.1399999999967	
NV	58718.8900000000005			21462.0900000000001	80180.9800000000006	
Grand Total	102631.7699999999971	11688.9600000000001	8880.8	23625.5900000000001	146827.1199999999973	

**Figure 7-1**      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 7-1, there are five unique values in the productline field: Classic Cars, Motorcycles, Planes, Ships, and Trains. This area is 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 the example, there are eight unique values in the state field: CA, CT, MA, NH, NJ, NV, NY, and PA. This area is called 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. BIRT Studio calculates the revenue total using the SUM function on the values in the extendedprice field. This area is called 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 cube in a data object. A data object 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. When using BIRT Studio as a Java Component, you cannot use the data sets in a data object.

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.

---

## 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.

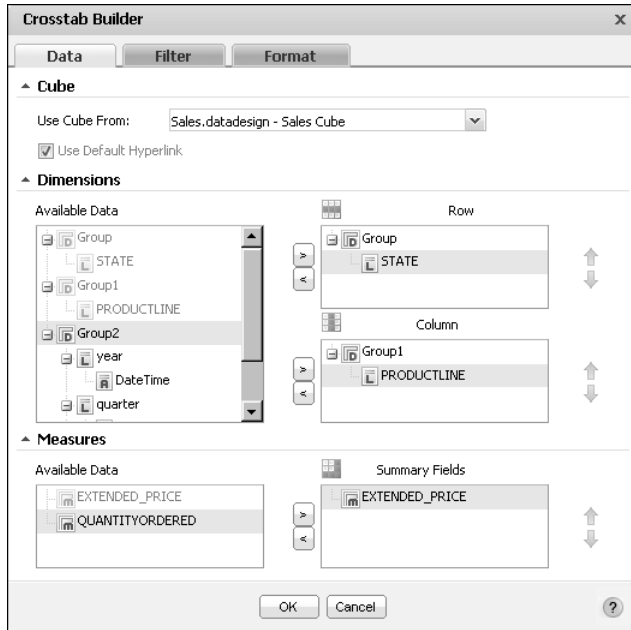
The following section describes each of these tasks in detail.

### 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 7-2 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.



**Figure 7-2** Displaying the dimensions and measures for a cross tab

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.

#### How to select data for a cross tab

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



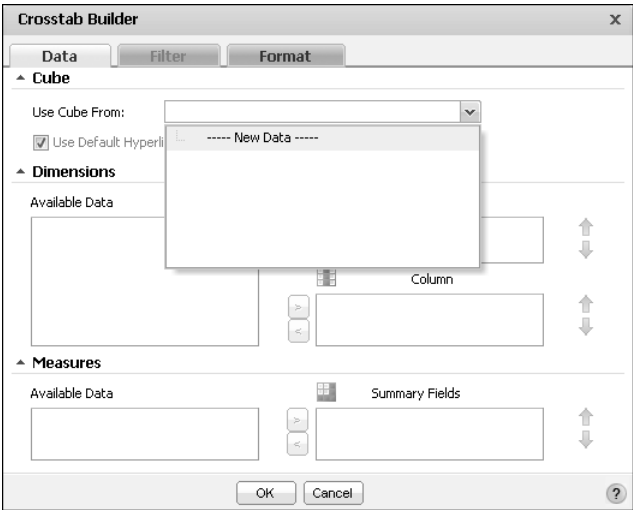
- 2 In Report Template, complete the following steps:
- 1 In Category, select Standard from the drop-down list.
  - 2 In Available Templates, select Crosstab, as shown in Figure 7-3.
  - 3 In Themes, select a theme from the drop-down list if your Crosstab template contains associated themes.



**Figure 7-3** 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 7-4.



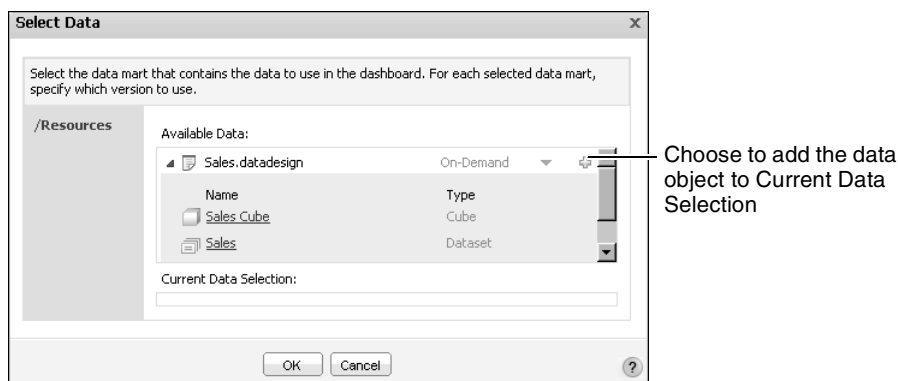
**Figure 7-4** Selecting a data object data source

Select Data appears, displaying the available data objects. If you use customized data objects, ensure that they are in the following folder location of the Actuate BIRT Java Components installation:

```
<context root>\WEB-INF\repository
```

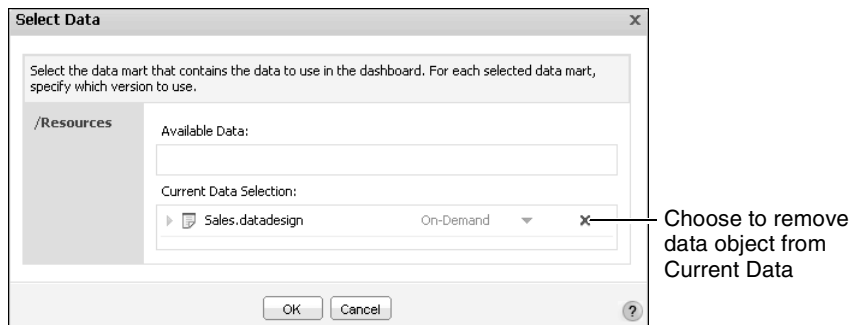
- 4 In Available Data, select a data object design or data object store file, that contains a cube.

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 one or more data objects from the list, and choose the + symbol next to each selection, as shown in Figure 7-5.



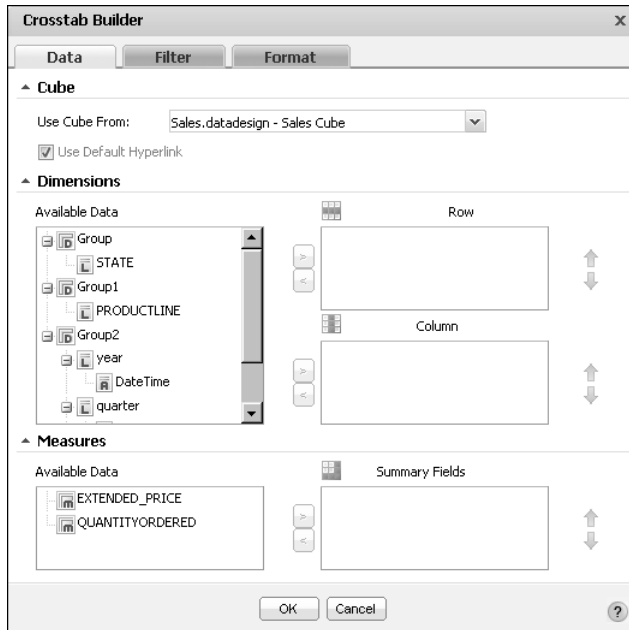
**Figure 7-5** Selecting a data object

The data object appears in Current Data Selection, as shown in Figure 7-6. Choose OK. You can modify your selection by choosing Delete, as shown in Figure 7-6, and selecting a new data object in Available Data.



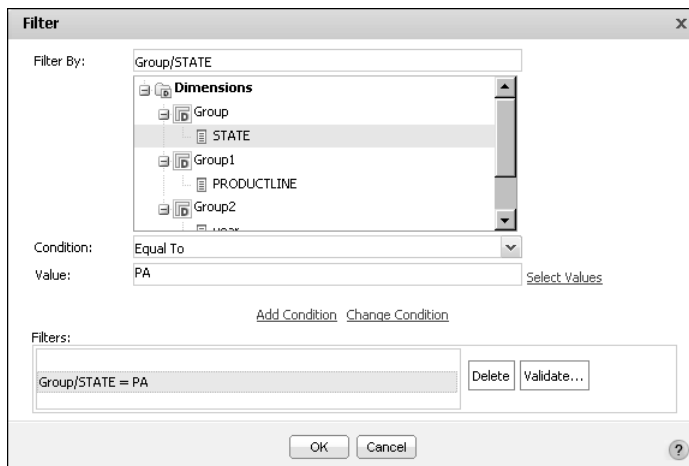
**Figure 7-6** Current data selection displaying the data object

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



**Figure 7-7** 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 7-8.



**Figure 7-8** 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 7-9.

	Classic Cars	Motorcycles	Vintage Cars	Grand Total
	EXTENDED_PRICE	EXTENDED_PRICE	EXTENDED_PRICE	EXTENDED_PRICE
PA	102856.2400000000026	39025.0900000000004	1925.0099999999999	240024.6200000000037
Grand Total	102856.2400000000026	39025.0900000000004	1925.0099999999999	240024.6200000000037

**Figure 7-9** A cross tab displaying filtered data rows

You can also select the Any Of 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 In Crosstab Builder—Data, in Dimensions, in Available Data, select a grouped data field, as shown in Figure 7-10. You can select multiple grouped data fields by pressing Ctrl while you select each field.
- 2 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 7-10, specifies State as a Row, and Product Line as a column.
- 3 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 7-10 sets the Extended Price field as the summary field.

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

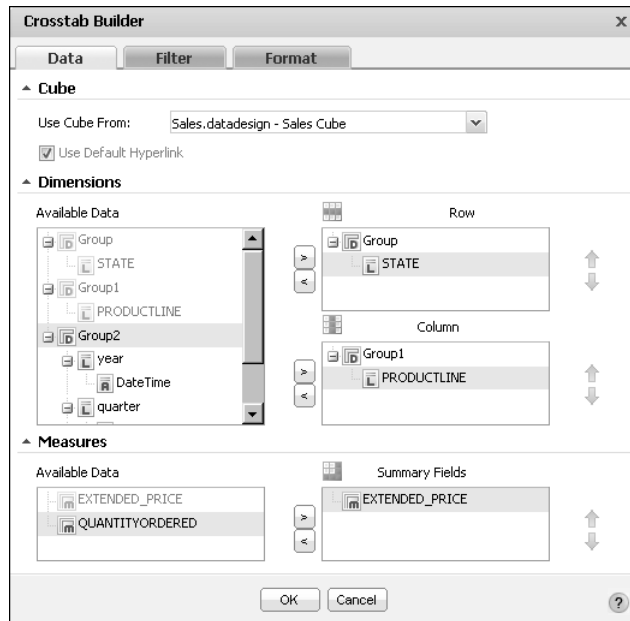
### How to provide data from multiple cubes to a cross tab



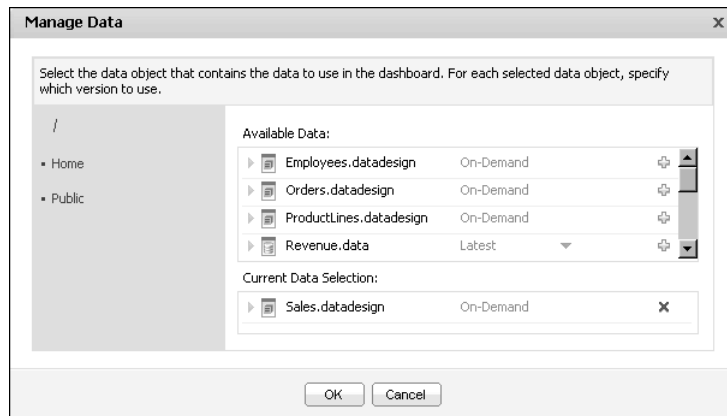
Choose Manage Data to select a different data object data source. Manage Data appears. Complete the following steps:

In Available Data, choose the '+' symbol to add a new data object. Repeat this action to make additional cubes available to the cross tab. Choose OK.

The data object you selected appears in Current Data Selection, as shown in Figure 7-11. The selected cubes now appear in Use Cube From on Table Builder. You can now use any of the cubes as a data source for the cross tab.



**Figure 7-10** Selecting data to display in the cross tab



**Figure 7-11** Adding a cube to Current Data Selection

## Displaying aggregate values

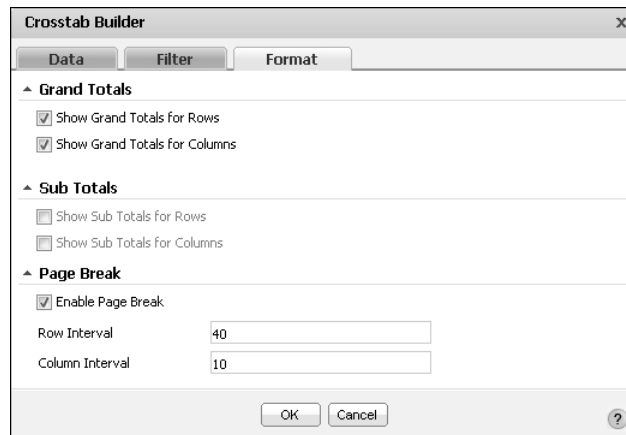
You can display totals for each dimension that you add to a cross tab and for each level within a multilevel dimension. The cross tab in Figure 7-10 displays sales data by product line, and state. The rows and columns that display the subtotals and grand totals are highlighted in the cross tab. In the example cross tab you just

created, no subtotals appear. Subtotals are displayed when you set a multidimensional data field, as a row or column.

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 7-12.
- 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 7-10 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



**Figure 7-12** Specifying display options for the aggregate values

- 4 In Page Break, select Enable Page Break to specify pagination properties for the cross tab, do the following:
  - In Row Interval, type a value such as 40, as shown in Figure 7-12.

- In Column Interval, type a value, such as 10, as shown in Figure 7-12.

Choose OK. Sample data appears in the cross tab, displaying grand totals for rows and columns, as shown in Figure 7-13.

	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 7-13** 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 7-14.

	Vintage Cars		Grand Total
	EXTENDED_PRICE		EXTENDED_PRICE
CA	366355.37000000000039		992815.49
CT	14101.07		119443.84
MA	105384.17999999999988		387238.75
NH	39376.65000000000002		116449.29
NJ	9035.360000000000003		9035.36
NV	21462.090000000000001		80180.98
NY	62342.280000000000003		400958.01
PA	34925.009999999999999		175264.34
Grand Total	652982.01		2281386.06

**Figure 7-14** 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 cube. 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 7-15.

	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	3273280.05000000000085

**Figure 7-15** 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

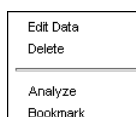
From BIRT Studio, you can launch BIRT Data Analyzer if you have purchased a license for it, to manipulate and further analyze cross-tab data. Using BIRT Data Analyzer, you can answer questions such as:

- 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 7-16.



**Figure 7-16** Launching BIRT Data Analyzer from BIRT Studio



# Presenting data in a chart

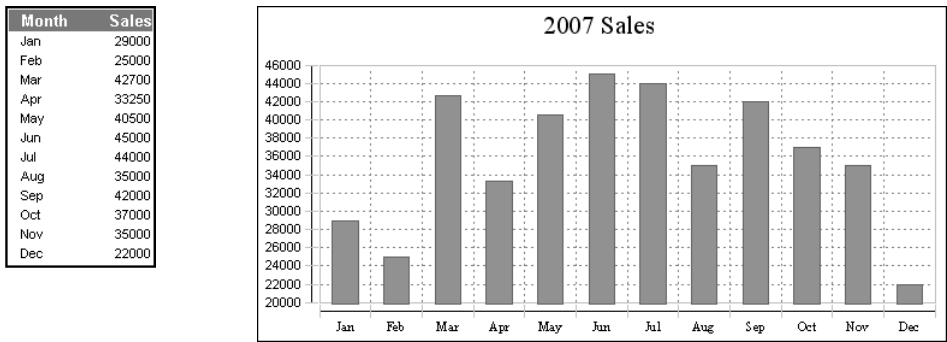
This chapter contains the following topics:

- 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

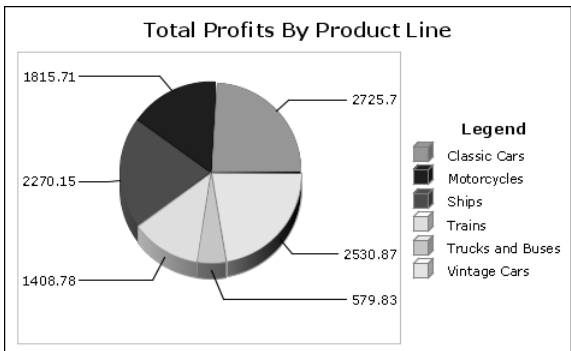
A chart is a graphical representation of data. Charts are particularly useful for summarizing numeric data and showing the relationship between sets of values, called series. For example, a chart can show sales by region, average temperatures by month, or the price of a stock over three months.

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.



**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
	<b>\$5,601.83</b>	<b>\$2,725.70</b>
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
	<b>\$5,460.98</b>	<b>\$1,815.71</b>
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
	<b>\$6,035.74</b>	<b>\$2,270.15</b>
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
	<b>\$4,469.68</b>	<b>\$1,408.78</b>
Product Line	Trucks and Buses	
Product Name	Total	Profit
1996 Peterbilt 379 Stake Bed with Outrigger	\$1,352.86	\$579.83
	<b>\$1,352.86</b>	<b>\$579.83</b>
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
	<b>\$6,722.51</b>	<b>\$2,530.87</b>
	<b>\$29,643.60</b>	<b>\$11,331.04</b>
	<b>\$29,643.60</b>	<b>\$11,331.04</b>

**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:

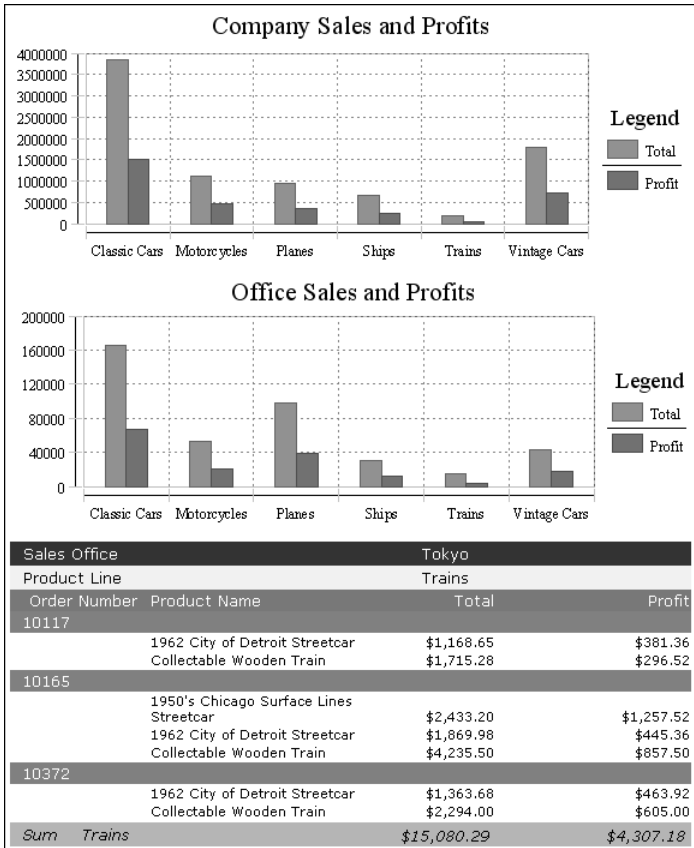
- 1 Insert a chart.
- 2 Choose a chart type.
- 3 Specify the data to present in the chart.
- 4 Format the chart.

# 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.



**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.

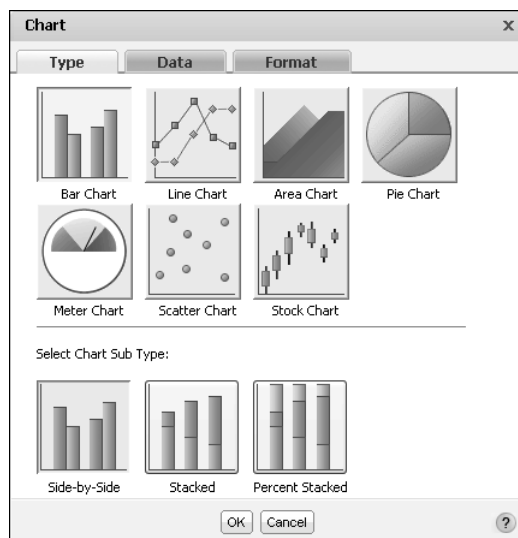
### How to insert a chart

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 choose a chart type, select data for the chart, and format the chart.



**Figure 8-5** The chart editor displaying the Chart–Type page

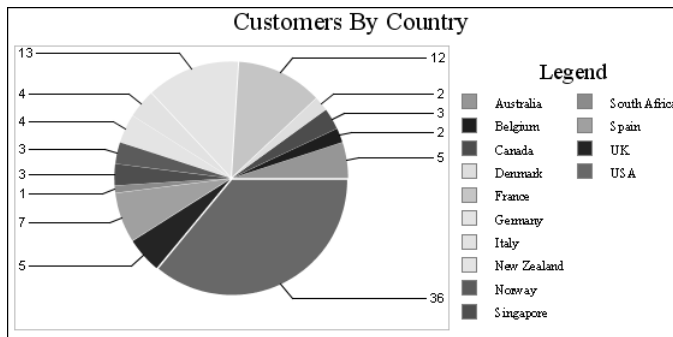
## Choosing a chart type

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

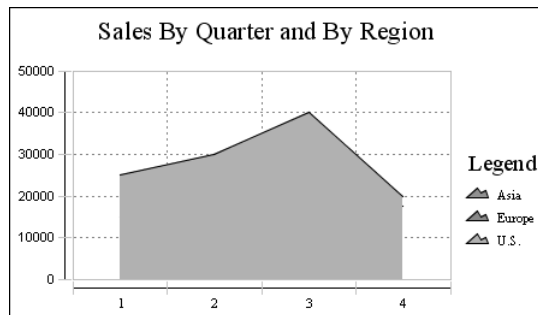
In a stacked area chart, multiple series are stacked vertically, as shown in Figure 8-7. The example shows that the stacked area chart is suitable for the data, because the chart displays totals for all series as well as the proportion that each series contributes to the total. The height of the top 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

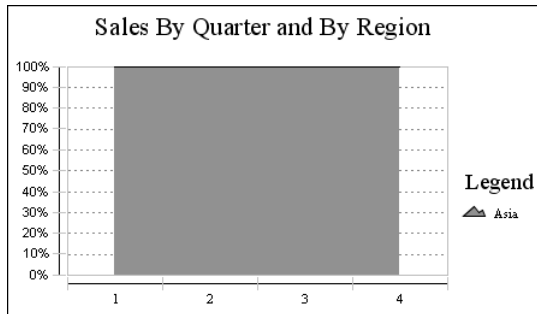
## Percent stacked area chart

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.

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 8-9** A percent stacked area chart displaying three series



**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.

### 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 8-11** A side-by-side bar chart



**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.



**Figure 8-13** A percent stacked bar chart displaying three series

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

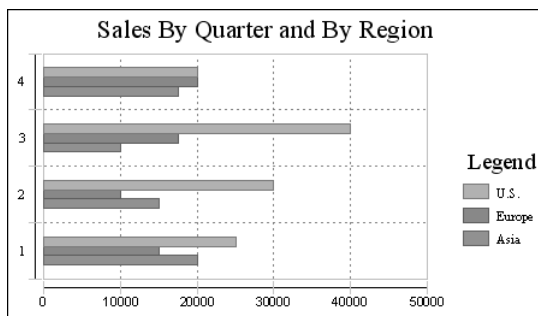
subtype if you are displaying only one series, for example, only sales for Asia. Figure 8-14 shows how a percent stacked bar chart looks when it displays only one region series. All the bars show a value of 100%, because that is the sales percentage of one region when it is compared to itself.



**Figure 8-14** A percent stacked bar chart displaying one series

## Horizontal bar chart

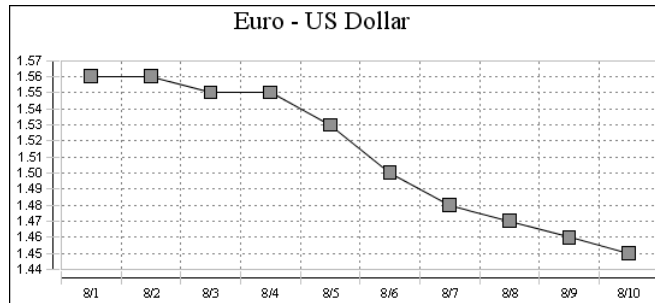
For all the bar chart subtypes, you can transpose, or flip, the axes to display data as horizontal bars. Figure 8-15 shows an example of a side-by-side bar chart with flipped axes. This feature is useful for showing data that contains many categories. For example, if you want to show sales numbers for five regions for twelve months, displaying the data requires sixty 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 ten days.



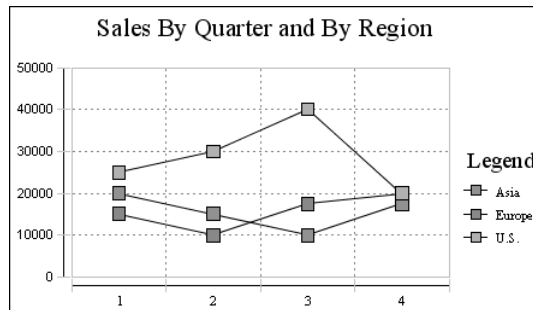
**Figure 8-16** A line chart

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.

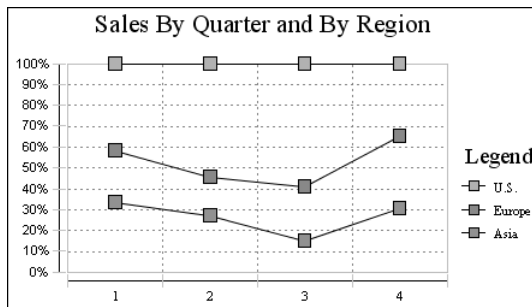


**Figure 8-18** A stacked line chart

## 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.

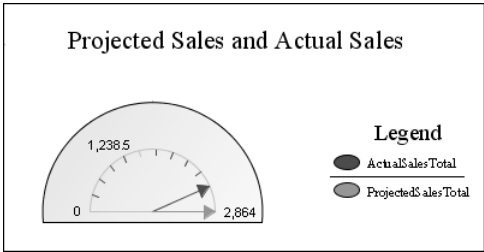
Like the stacked line chart, the percent stacked line chart is not as effective as its counterpart, the percent stacked area chart, for the reasons cited in the previous section.



**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

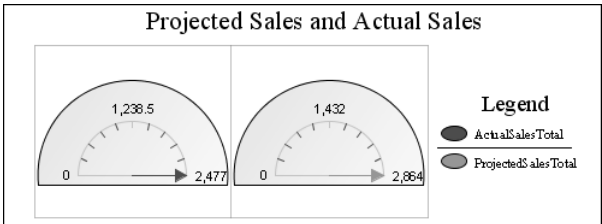
## Superimposed meter chart

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 shown 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.

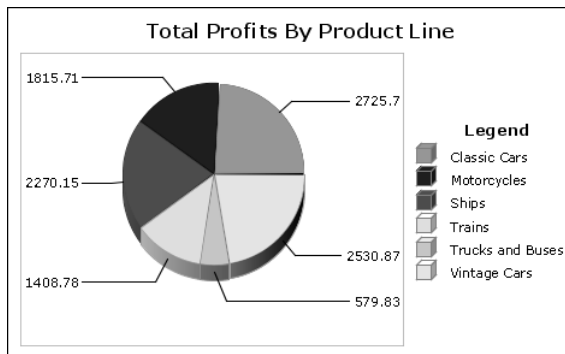


**Figure 8-21** A standard meter chart displaying values in two dials

## 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

when you want 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.

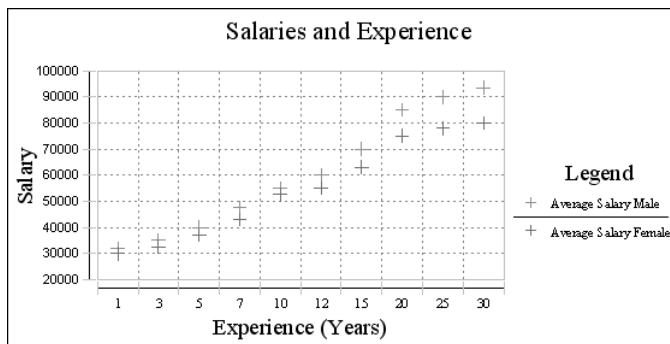


**Figure 8-22** A pie chart

## 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.



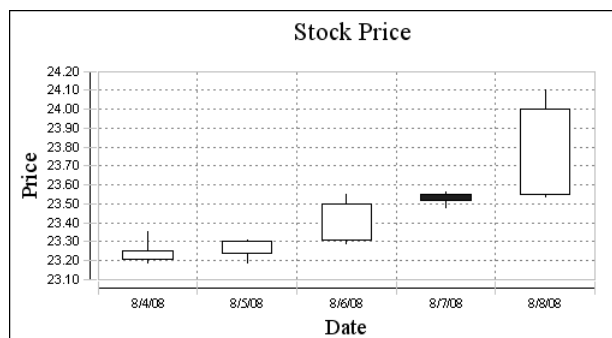
**Figure 8-23** A scatter chart

## 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.

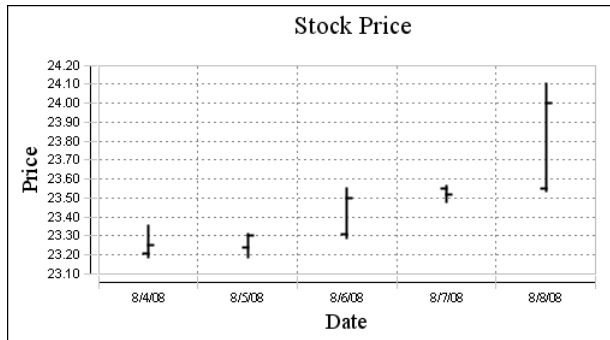


**Figure 8-24** A candlestick stock chart

### 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.

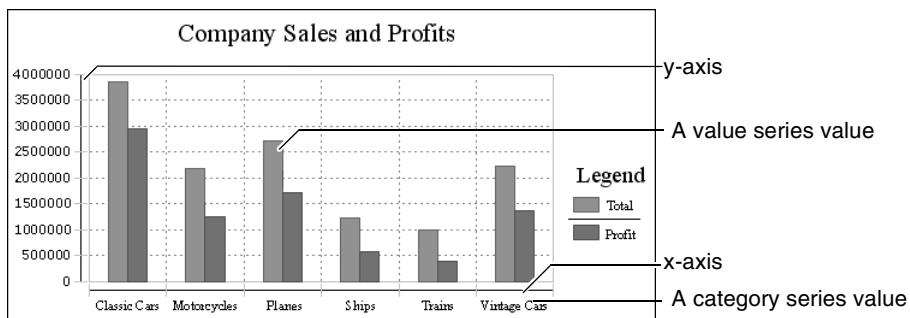


**Figure 8-25** A bar stick stock chart

## 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.



**Figure 8-26** Parts of a chart

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. 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.

Chart

Type

Data

Format

Categories (X Axis)

Product Line

☐ Group Categories

Values (Y Axis)

Data Columns

Aggregate Expression:

Series 1:

Total

Sum

Series 2:

Profit

Sum

Group Legend Items:

None

☒ Tooltip

Default

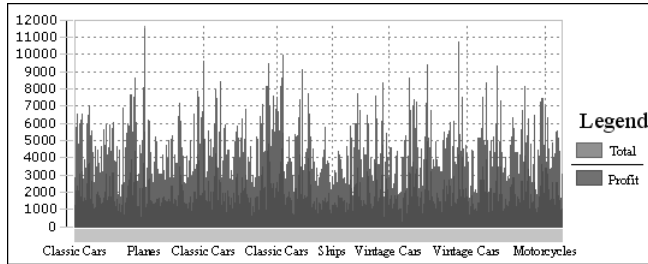
OK

Cancel

?

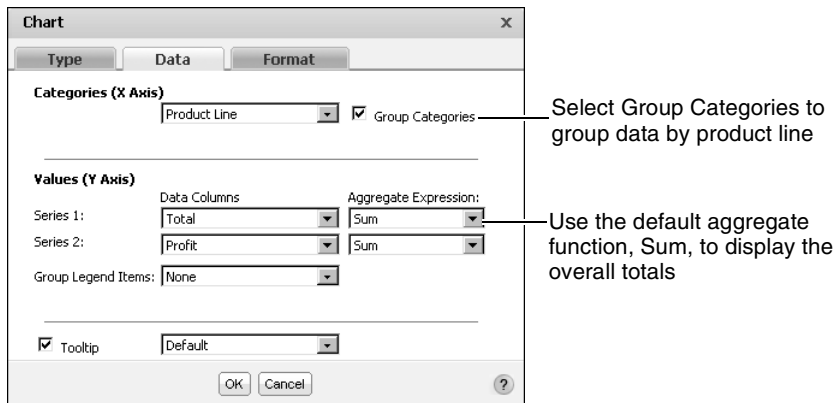
**Figure 8-28** Chart–Data showing columns selected for the category and values series

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 shown 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.



**Figure 8-30** Grouping and aggregating data for a 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.

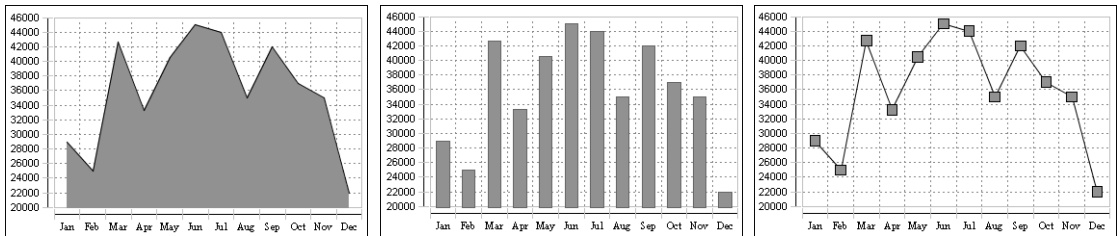
## Selecting data for an area, bar, or line chart

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

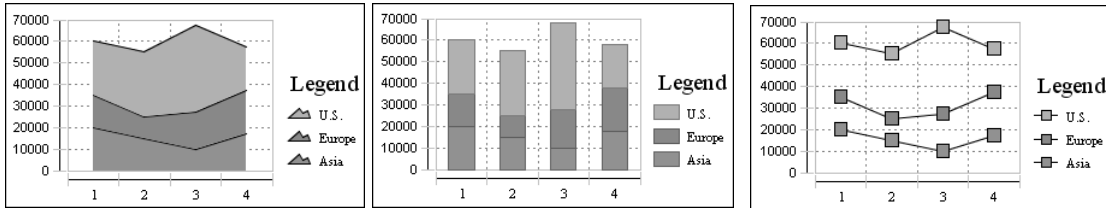
Figure 8-33 shows how the data is selected for the chart. You do not need to group or aggregate the data. As Figure 8-33 shows, all you do is select the Month column as the category series, and the Sales column as the value series.

The screenshot shows the 'Chart' configuration window with the 'Data' tab selected. Under 'Categories (X Axis)', 'Month' is selected in the dropdown, and the 'Group Categories' checkbox is unchecked. Under 'Values (Y Axis)', 'Series 1' has 'Sales' selected in the 'Data Columns' dropdown and 'Sum' in the 'Aggregate Expression' dropdown. 'Series 2' has empty dropdowns. 'Group Legend Items' is set to 'None'. The 'Tooltip' checkbox is checked, and 'Default' is selected in the dropdown. 'OK' and 'Cancel' buttons are at the bottom.

**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.

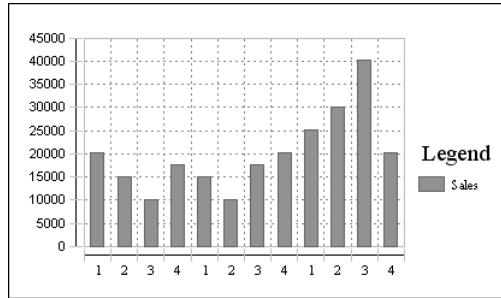
Region	Quarter	Sales
Asia	1	20000
Asia	2	15000
Asia	3	10000
Asia	4	17500
Europe	1	15000
Europe	2	10000
Europe	3	17500
Europe	4	20000
U.S.	1	25000
U.S.	2	30000
U.S.	3	40000
U.S.	4	20000

**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 shows the Chart configuration dialog box. The 'Type' tab is selected. The 'Categories (X Axis)' dropdown is set to 'Quarter'. The 'Values (Y Axis)' section shows 'Series 1' as 'Sales' with an aggregate expression of 'Sum'. 'Series 2' is empty with an aggregate expression of 'Sum'. The 'Group Legend Items' dropdown is set to 'Region'. The 'Tooltip' checkbox is checked, and the tooltip text is set to 'Default'.

**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

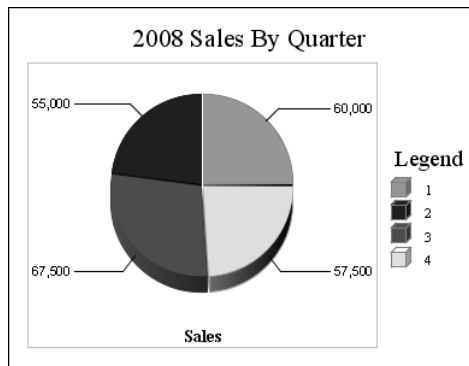
## Selecting data for a pie chart

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.

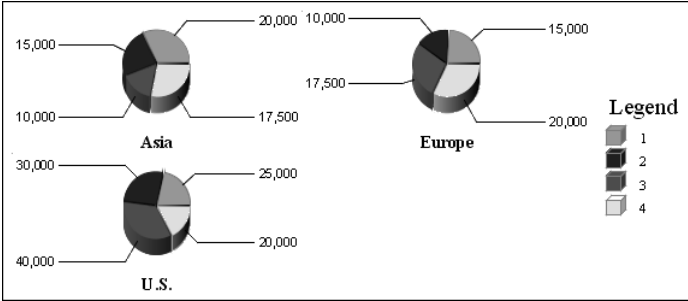
**Figure 8-40** The data selected for the chart in Figure 8-38

## Example 2

Figure 8-41 shows a series of pie charts. Each pie chart shows the sales by quarter for a particular region.

Figure 8-42 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.

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 8-41** A series of pie charts showing sales by quarter and by regions

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

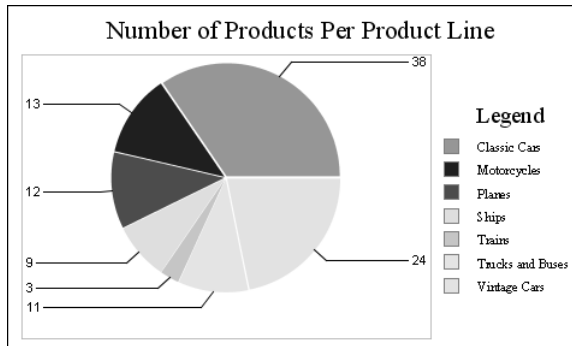
**Figure 8-42** The table data used by the charts in Figure 8-41

Figure 8-43 shows the 'Chart' dialog box, specifically the 'Data' tab. The 'Categories (X Axis)' is set to 'Quarter'. The 'Values (Y Axis)' section shows 'Series 1' with 'Sales' as the Data Column and 'Sum' as the Aggregate Expression. 'Series 2' is also set to 'Sum'. The 'Group Legend Items' are set to 'Region'. The 'Tooltip' checkbox is checked, and the tooltip text is set to 'Default'.

**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	\$69.14	\$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	\$63.51	\$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

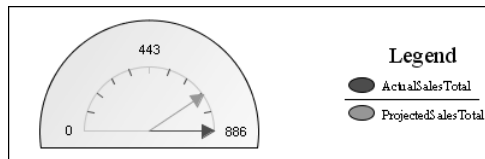
## Selecting data for a meter chart

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.

**Chart**

**Type** **Data** **Format**

**Categories (X Axis)**

Year ☐ Group Categories

**Values (Y Axis)**

Data Columns Aggregate Expression:

Series 1: ActualSalesTotal Sum

Series 2: ProjectedSalesTotal Sum

Group Legend Items: None

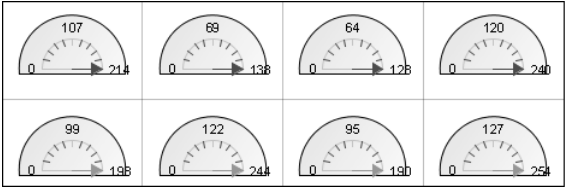
☒ Tooltip Default

OK Cancel ?

**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.

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 shows how the data is selected for the chart.

**Figure 8-52** The data selected for the chart in Figure 8-50

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.

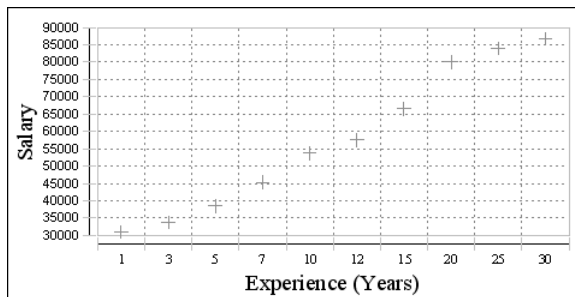
## Selecting data for a scatter chart

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.



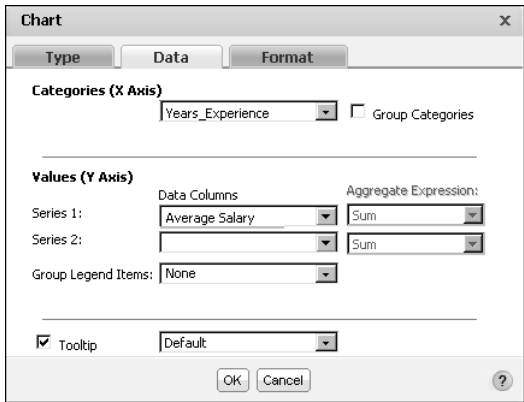
**Figure 8-53** A scatter chart showing the relationship between salaries and years of experience

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. 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.



**Figure 8-55** The data selected for the chart in Figure 8-53

**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.

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.

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 Vlade	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

The screenshot shows the 'Chart' configuration window with the 'Type' tab active. The 'Categories (X Axis)' section has 'Years\_Experience' selected in the dropdown and the 'Group Categories' checkbox checked. The 'Values (Y Axis)' section shows 'Series 1' with 'Salary' selected in the 'Data Columns' dropdown and 'Average' in the 'Aggregate Expression' dropdown. 'Series 2' has an empty 'Data Columns' dropdown and 'Sum' in the 'Aggregate Expression' dropdown. The 'Group Legend Items' dropdown is set to 'Gender'. At the bottom, the 'Tooltip' checkbox is checked and set to 'Default'. 'OK' and 'Cancel' buttons are at the bottom right.

**Figure 8-58** The data selected for the chart in Figure 8-56

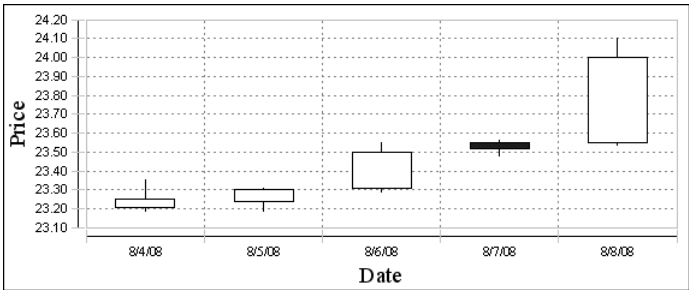
## Selecting data for a stock chart

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.

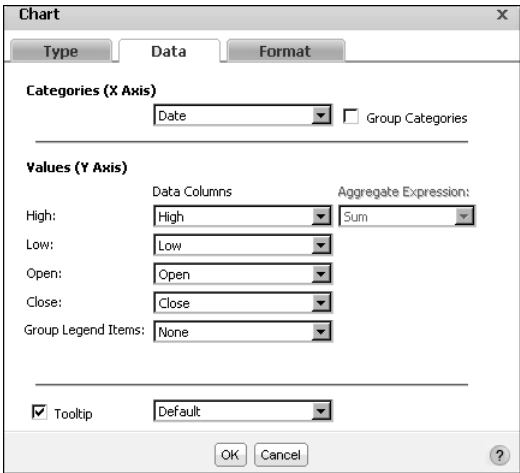


**Figure 8-59** A stock chart showing high, low, open, and close values  
Figure 8-60 shows the table data that the stock chart uses.

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.

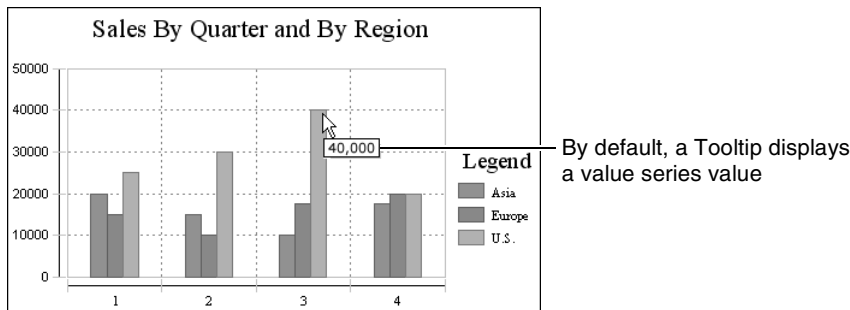


**Figure 8-61** The data selected for the chart in Figure 8-59

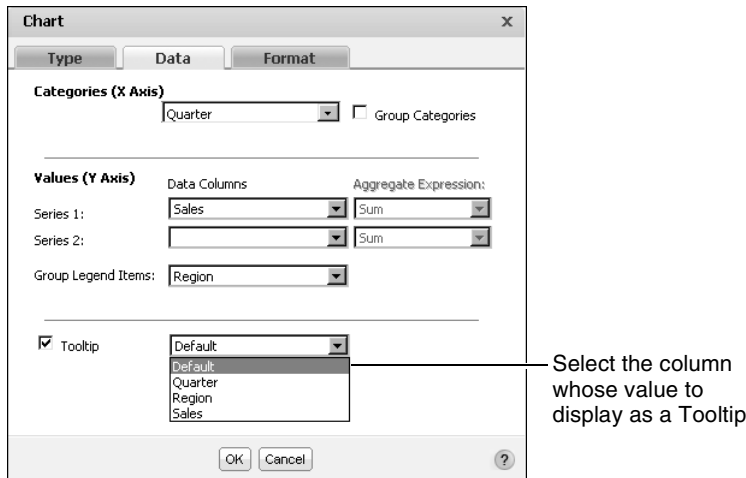
## Displaying additional series information

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 places 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 mouse is placed 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

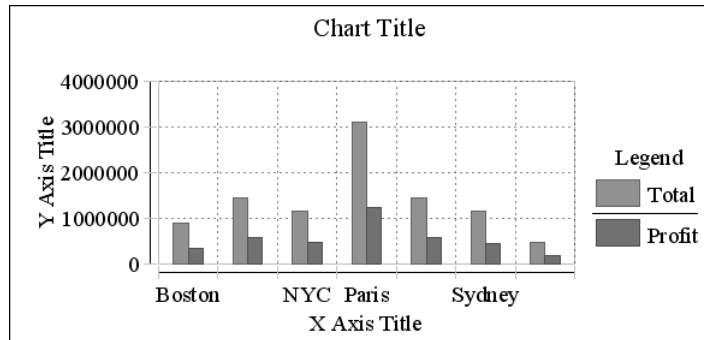


**Figure 8-63** Selecting a value to display as a Tooltip



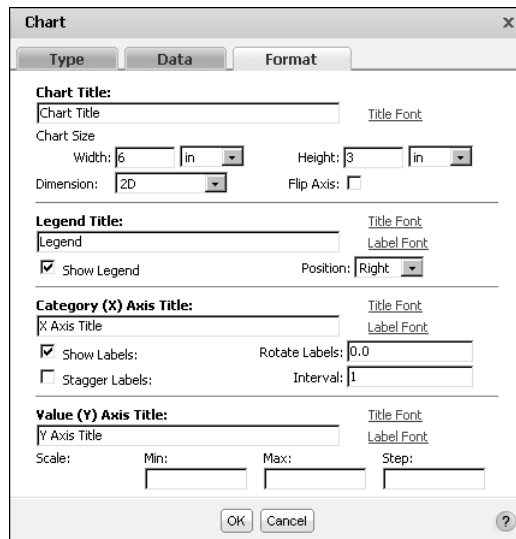
## Formatting a chart

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.



**Figure 8-65** Chart-Format displaying the default options for a bar chart

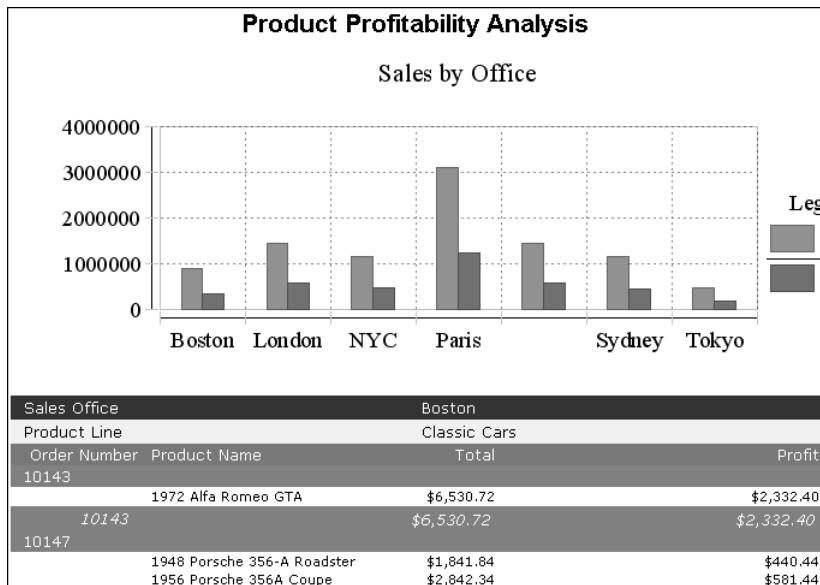
The options that appear differ depending on the chart type. For example, a pie chart does not have axes, so the Category (X) Axis Title and Value (Y) Axis Title options do not appear on Chart-Format for a pie chart.

## 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.

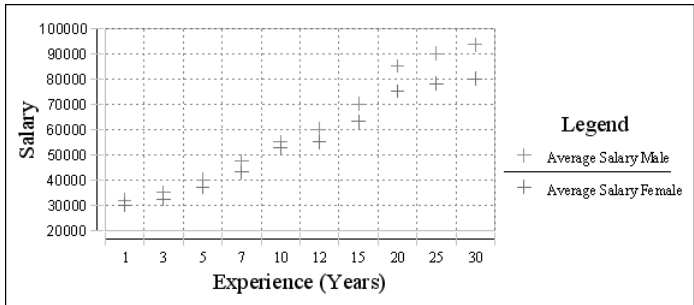


**Figure 8-66** A report where the chart width exceeds the table width

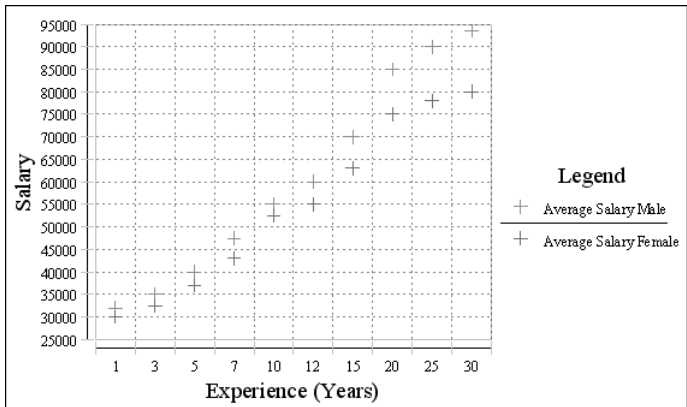
For more information about the Layout Preference option, see “Changing the layout type, page size, and orientation” in Chapter 2, “Editing and formatting report content.”

### 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



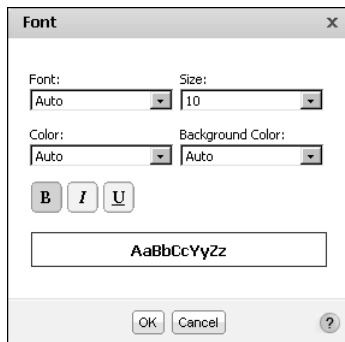
**Figure 8-68** A scatter chart with its height increased

### 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,

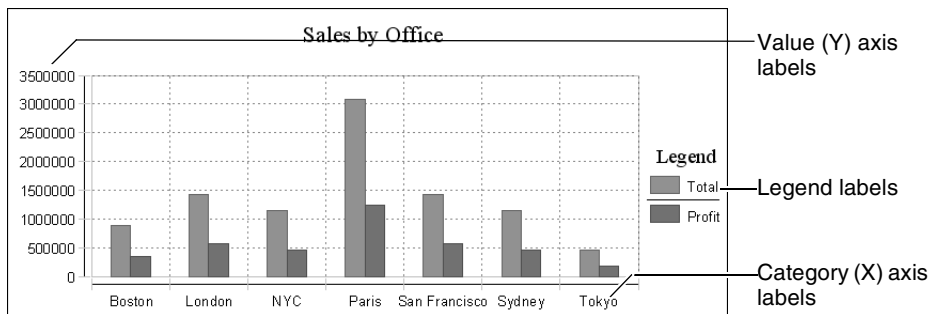
type new titles in the text boxes in Chart Title, Category (X) Axis Title, and Value (Y) Axis Title, respectively. If you do not want to display any titles, delete the text in the text boxes. You also can change the font attributes of every title and label in the chart. To format a title, choose Title 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** Setting Font attributes

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.

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.

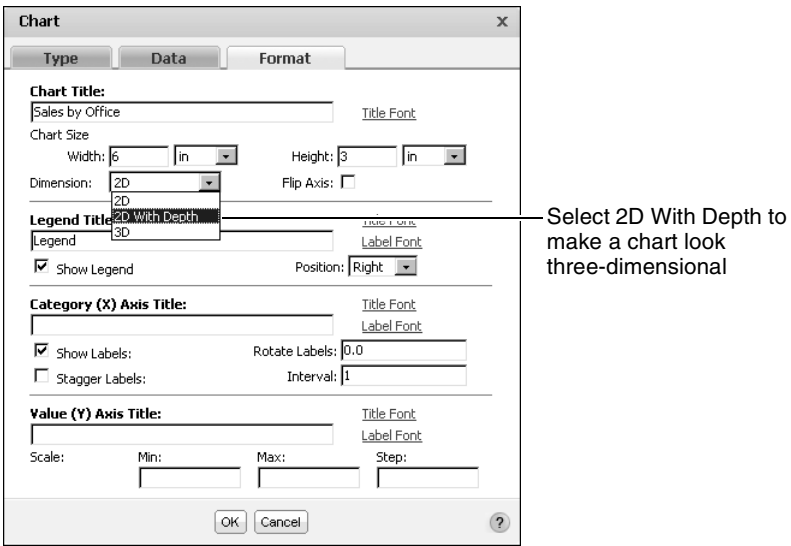


Figure 8-71      Setting the Dimension option

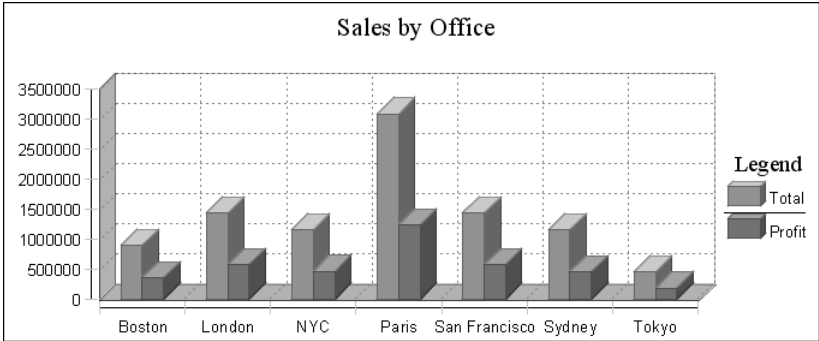


Figure 8-72      A bar chart using the 2D With Depth setting

## 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.

**Chart**

**Type** **Data** **Format**

**Chart Title:**  
Sales by Office [Title Font](#)

**Chart Size**  
Width: 7.05 in Height: 3 in  
Dimension: 2D With Depth Flip Axis: ☐

**Legend Title:**  
Legend [Title Font](#) [Label Font](#)

☒ Show Legend Position: **Right**  
Above  
Below  
Left  
Inside

**Category (X) Axis Title:**

☒ Show Labels: Rotate Labels: ☐ Interval: 1

☐ Stagger Labels:

**Value (Y) Axis Title:** [Title Font](#) [Label Font](#)

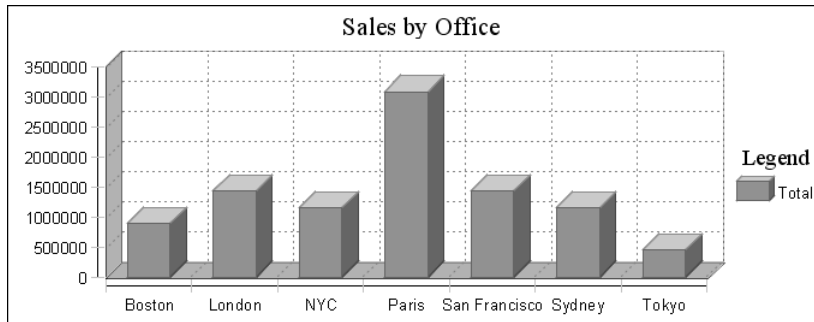
Scale: Min: Max: Step:

OK Cancel ?

Select a value in Position to set the placement of the legend

**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

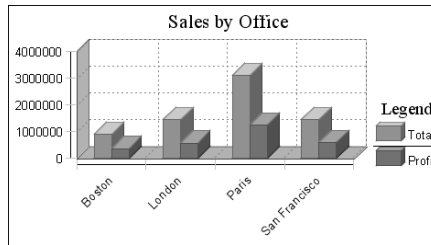
## Formatting labels to fit on the x-axis

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.

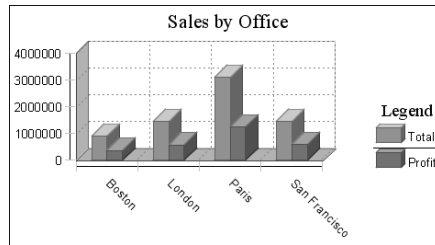
## Rotating x-axis labels

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.

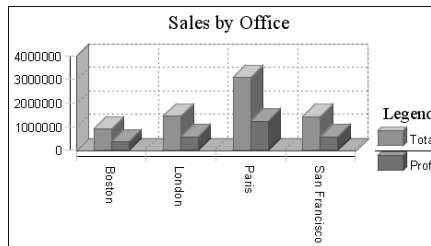
Rotate Labels = 45.0



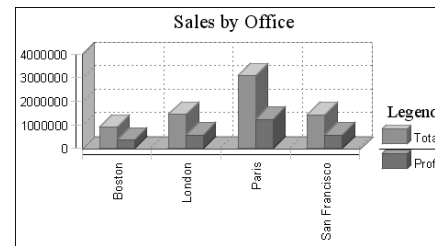
Rotate Label = -45.0



Rotate Labels = 90.0



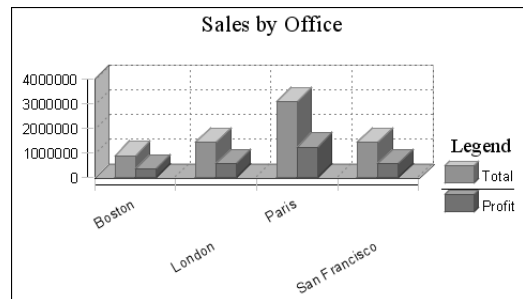
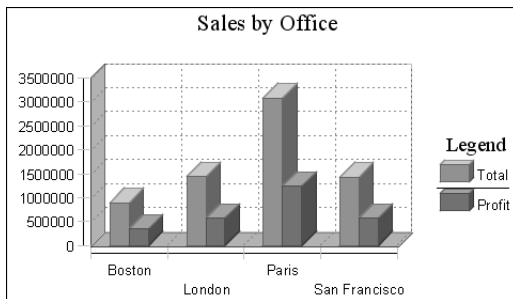
Rotate Label = -90.0



**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.



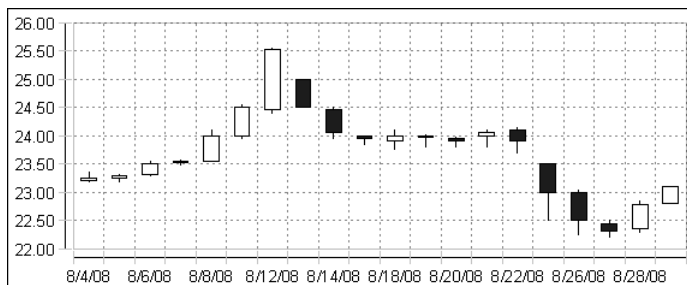
**Figure 8-76** Charts using different values for the Stagger Labels and Rotate Labels options

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.

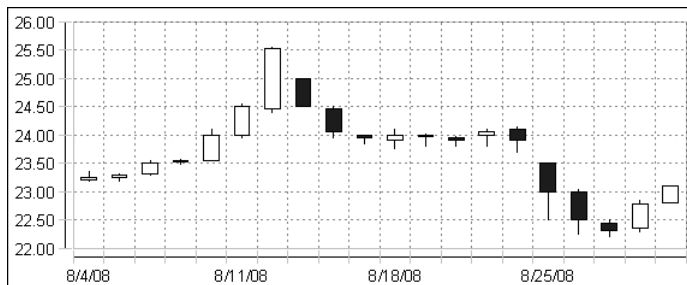
## Displaying *x*-axis labels by interval

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



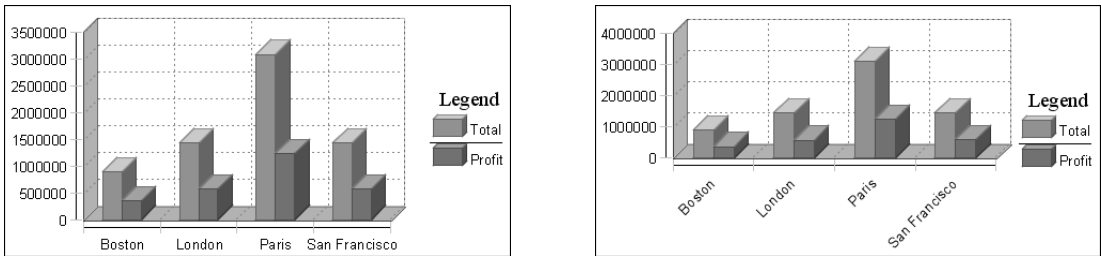
**Figure 8-78** A stock chart displaying the first trading date per week

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

**Chart**

**Type** **Data** **Format**

**Chart Title:** Sales by Office [Title Font](#)

**Chart Size**

Width: 5 in Height: 3 in

Dimension: 2D With Depth Flip Axis: ☐

**Legend Title:** Legend [Title Font](#) [Label Font](#)

☒ Show Legend Position: Right

**Category (X) Axis Title:** [Title Font](#) [Label Font](#)

☒ Show Labels: Rotate Labels: 45.0

☐ Stagger Labels: Interval: 1

**Value (Y) Axis Title:** [Title Font](#) [Label Font](#)

Scale: Min: Max: Step:

0 3500000.0 500000.0

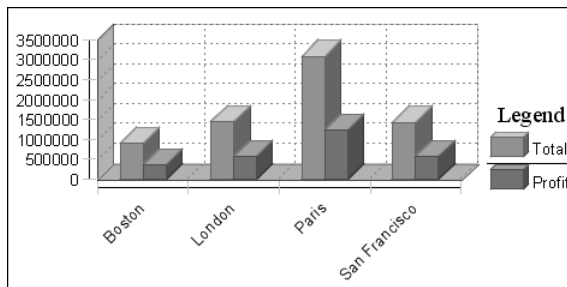
OK Cancel ?

**Figure 8-80** Chart–Format showing modified Scale values

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 8-81** The chart using the modified Scale values

## Working with Flash charts

BIRT Studio supports the creation of Flash charts and standard charts. Flash charts are similar to standard charts, and are designed to be used in a similar manner. Standard charts display 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 to a BIRT report. For example, an animated Flash column chart can progressively draw its columns from bottom to top. The following examples display a standard doughnut chart, and a segmented Flash doughnut chart, viewed in the Actuate BIRT Viewer. The animated Flash doughnut chart slices-off each segment of the doughnut chart, as you select the segment.

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

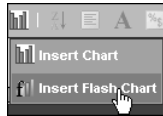


**Figure 8-82** Standard doughnut chart, and segmented Flash doughnut chart

## 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.

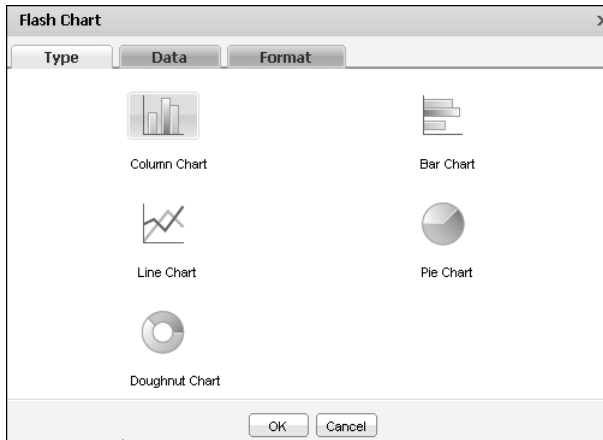


**Figure 8-83** Inserting a Flash chart

Flash Chart appears.

- 2 On Flash Chart—Type, select a type of chart. Figure 8-84 shows the types of Flash charts available in BIRT Studio.
- 3 On 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 formatting options available for a Flash chart are similar to the formatting options available for standard charts.

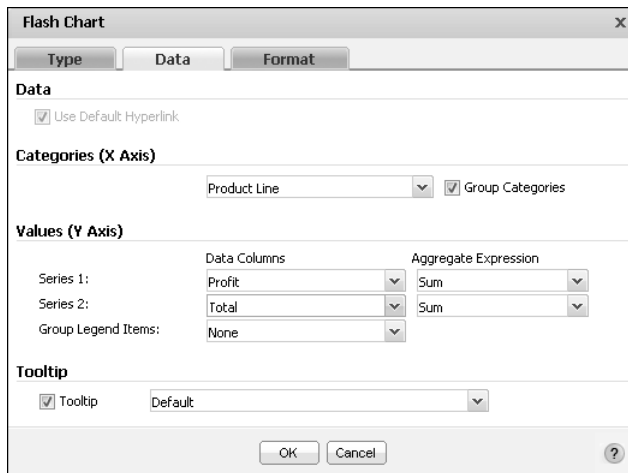


**Figure 8-84** Selecting the type of Flash chart

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 standard charts. Figure 8-85 displays Flash Chart—Data.



**Figure 8-85** Selecting data for a Flash 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.

Specify the value series and the category series, if applicable. Set the grouping options and select a value to display for the Tooltip from the drop-down list.

### **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.

### **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.

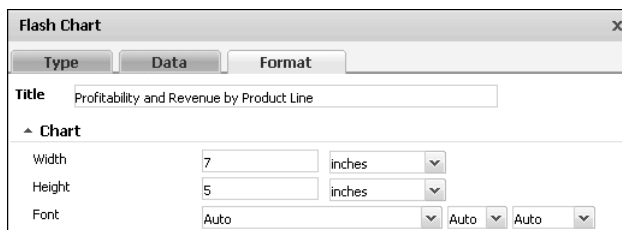
### Formatting a column, line, or bar chart

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.

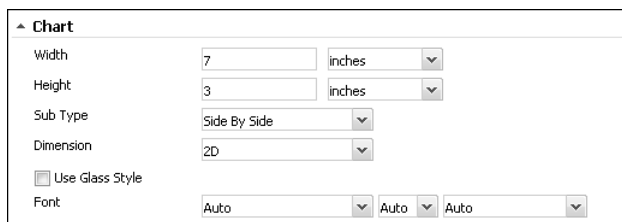
### Specifying chart properties

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.

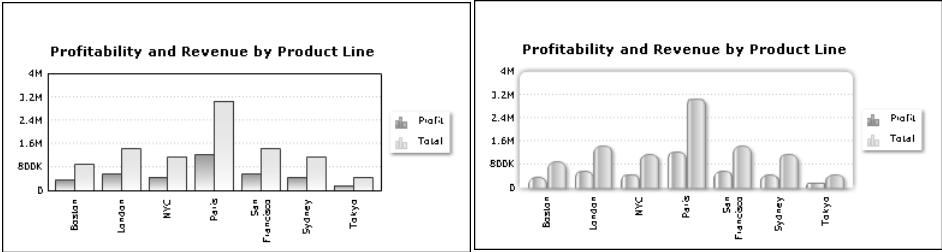


**Figure 8-86** Specifying chart properties

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



**Figure 8-88** Deselecting and selecting glass style in a two-dimensional 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.

### 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 specify whether to display the legend to the right of the chart, or below the chart, as shown in Figure 8-89.

Legend

☒ Show Legend

Title

Position
 

Right

**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.

Categories (X Axis)

Title

☒ Show Labels

☒ Stagger Labels

Labels Interval

Labels Angle
 

45

**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 shows a dialog box titled "Values (Y Axis)". It contains the following fields and controls:

- Title:** A text field containing "Revenue and Profit".
- Minimum Value:** A text field containing "0".
- Maximum Value:** A text field containing "80000".
- Auto Step:** A checked checkbox.
- Step Interval:** A spin box showing the value "1".

**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 shows a dialog box titled "Flash Chart" with three tabs: "Type", "Data", and "Format". The "Format" tab is active. It contains the following sections and controls:

- Title:** A text field containing "Quantity ordered by product line".
- Chart:**
  - Width:** A text field containing "9" and a dropdown menu set to "inches".
  - Height:** A text field containing "5" and a dropdown menu set to "inches".
  - Dimension:** A dropdown menu set to "2D with depth".
  - Rotation:** A slider control set to "180".
  - Auto Radius:** A checked checkbox.
  - Radius:** A spin box showing the value "50" and a unit dropdown set to "pixel".
  - Font:** A dropdown menu set to "Auto", followed by two smaller dropdown menus also set to "Auto".
- Legend:**
  - Show Legend:** A checked checkbox.
  - Title:** A text field containing "Product line".
  - Position:** A dropdown menu set to "Right".

At the bottom of the dialog are "OK" and "Cancel" buttons.

**Figure 8-92** Formatting a pie Flash chart

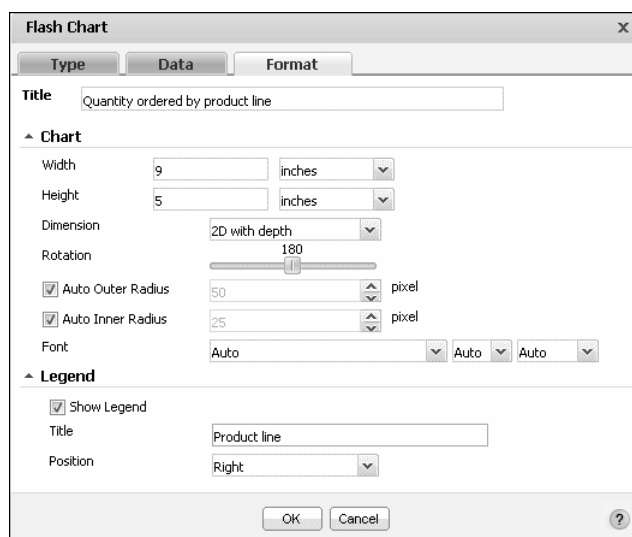
- 2 Specify the size of the pie:
  - Select Auto Radius to enable BIRT Studio to set a radius for the pie, as shown in Figure 8-92.



- 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.

# 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.

## ADD\_MONTH( )

**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 1 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 2 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 2 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)

## BETWEEN( )

**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 5 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 5 days to the orderDate



value. If an orderDate value is in a month other than December, add 3 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.

## DIFF\_DAY( )

**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.

## DIFF\_QUARTER( )

**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 3 months is equal to 1 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.

<b>Syntax</b>	DIFF_SECOND(date1, date2)
<b>Arguments</b>	<p><b>date1</b> 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.</p> <p><b>date2</b> 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.</p>
<b>Returns</b>	The number of seconds between date1 and date2.
<b>Example</b>	<p>The following example calculates the number of seconds between each value in the startTime data field and each value in the finishTime data field:</p> <pre>DIFF_SECOND([startTime], [finishTime])</pre> <p>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:</p> <pre>DIFF_SECOND(NOW(), "12/24/08 12:00 AM")</pre>

---

## DIFF\_WEEK( )

Calculates the number of weeks between two date values.

<b>Syntax</b>	DIFF_WEEK(date1, date2)
<b>Arguments</b>	<p><b>date1</b> The first date or date expression to use in the calculation.</p> <p><b>date2</b> The second date or date expression to use in the calculation.</p>
<b>Returns</b>	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.
<b>Example</b>	<p>The following example calculates the number of weeks between each value in the askByDate data field and each value in the shipByDate data field:</p> <pre>DIFF_WEEK([askByDate], [shipByDate])</pre> <p>The following example calculates the number of weeks from each value in the orderDate data field to the end of the year:</p> <pre>DIFF_WEEK([orderDate], "1/1/09")</pre>

---

## 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, dolfTrue, dolfFalse)

**Arguments** **condition**

The condition to test.

**dolfTrue**

The value to return if condition evaluates to TRUE.

**dolfFalse**

The value to return if condition evaluates to FALSE.

**Returns** Returns the dolfTrue value if condition is TRUE or the dolfFalse 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 5 days to the orderDate

## IN( )

value. If a orderDate value is in a month other than December, add 3 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]))
```

## LEN( )

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.

## MOD( )

**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.

## NOW( )

<b>Syntax</b>	NOTNULL(value)
<b>Argument</b>	<b>value</b> The data field in which to check for non-null values.
<b>Returns</b>	True if a value in the specified data field is not a null value; returns false otherwise.
<b>Example</b>	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.  <pre>IF(NOTNULL([BirthDate]), [BirthDate], "No date specified")</pre>

---

## NOW( )

Returns the current date and time.

<b>Syntax</b>	NOW( )
<b>Returns</b>	The current date and time. For example:  <pre>Sep 23, 2008 11:56 AM</pre>
<b>Example</b>	The following example uses the DIFF_MINUTE( ) and NOW( ) functions to calculate the number of minutes from the current date and time to Christmas:  <pre>DIFF_MINUTE(NOW(), "12/25/08 12:00 AM")</pre>

---

## QUARTER( )

Returns the quarter number for a specified date value.

<b>Syntax</b>	QUARTER(date)
<b>Arguments</b>	<b>date</b> The date or date expression whose quarter number to get.
<b>Returns</b>	A number from 1 to 4 that represents the quarter for a specified date value. Quarter 1 starts in January.
<b>Example</b>	The following example displays the quarter number for each value in the CloseDate data field:  <pre>QUARTER([CloseDate])</pre> The following example displays a string—Q1, Q2, Q3, or Q4—for each value in the CloseDate data field:  <pre>"Q" &amp; QUARTER([CloseDate])</pre>

---

## 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.

## 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)
```

---

## ROUND( )

Rounds a number down to a specified number of digits.

**Syntax** ROUND( )

ROUND( )

**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, ROUND( ) returns 1545. If the PriceEstimate value is 1545.90, ROUND( ) returns 1545.

```
ROUND([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,



ROUNDUP( ) returns 1545.5. If the PriceEstimate value is 1545.23, ROUNDUP( ) returns 1545.2.

```
ROUNDUP([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, ROUNDUP( ) returns 1540. If the PriceEstimate value is 1338.50, ROUNDUP( ) returns 1330.

```
ROUNDUP([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, ROUNDUP( ) 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 travelled 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.

## TRIMLEFT( )

**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.

<b>Syntax</b>	UPPER(str)
<b>Argument</b>	<b>str</b> The string to convert to uppercase.
<b>Returns</b>	The specified string in all uppercase letters.
<b>Example</b>	The following example displays all the string values in the customerName data field in all uppercase:  <code>UPPER ( [customerName] )</code>

## WEEK( )

Returns a number from 1 to 52 that represents the week of the year.

<b>Syntax</b>	WEEK(date)
<b>Argument</b>	<b>date</b> The date or date expression whose week of the year to get.
<b>Returns</b>	A number that represents the week of the year for the specified date value.
<b>Example</b>	The following example gets the week number of the year for each date value in the ShipDate data field:  <code>WEEK ( [ShipDate] )</code>

## WEEKDAY( )

Returns the day of the week for a specified date value.

<b>Syntax</b>	WEEKDAY(date, option)
<b>Arguments</b>	<b>date</b> The date or date expression from which you want to get the day of the week.  <b>option</b> A number that represents the weekday format to return. Use one of the following values: <ul style="list-style-type: none"> <li>■ 1 to get the day as a number from 1 (Sunday) to 7 (Saturday).</li> <li>■ 2 to get the day as a number from 1 (Monday) to 7 (Sunday).</li> <li>■ 3 to get the day as a number from 0 (Monday) to 6 (Sunday).</li> <li>■ 4 to get the full weekday name, for example, Wednesday. The result is locale-specific.</li> <li>■ 5 to get the abbreviated weekday name, for example Wed. The result is locale-specific.</li> </ul>

YEAR( )

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

Table 9-1 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.

**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.
COUNTDISTINCT	Counts the number of unique values in a set of data rows. If a report column contains values 5, 2, 5, 7, and 10, COUNTDISTINCT 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.
<i>(continues)</i>	
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]



**Table 9-2** Operators in computed column expressions

Operator	Use to	Example
*	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")
<>	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 <i>n</i> values.	SalesAmount Bottom N 10
Bottom Percent	Test if a column value is in the bottom <i>n</i> 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
<i>(continues)</i>		
Is False	Test if a column value is False.	LoanApproved Is False
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

**Table 9-3** Operators in conditional formatting and filter condition expressions (continued)

Operator	Use to	Example
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 <i>n</i> values.	SalesAmount Top N 10
Top Percent	Test if a column value is in the top <i>n</i> percent of all values.	SalesAmount Top Percent 5



# Part Two

---

## Customizing and Integrating BIRT Studio



# Introduction to BIRT Studio customization

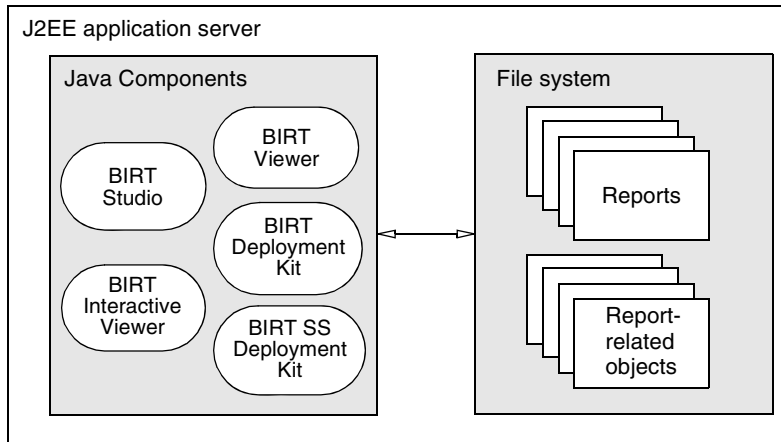
This chapter covers the following topics:

- About BIRT Studio and the Actuate Java Components
- Required Actuate software
- Understanding the context root

---

## About BIRT Studio and the Actuate Java Components

A user accesses BIRT Studio from Actuate Java Components, a web-based application that provides reporting services. Actuate Java Components supports accessing, running, and viewing reports. BIRT Studio is a licensed option of Actuate Java Components. Figure 10-1 shows the relationship between BIRT Studio and Actuate Java Components.



**Figure 10-1** Actuate Java Components architecture

Actuate Java Components is a web application that supports accessing and working with report information using a web browser. You deploy Actuate Java Component on an application server. Actuate Java Component accesses documents in a file system repository.

When accessed the BIRT Studio appears in your web browser as shown in Figure 10-2.

---

## Required Actuate software

To work with BIRT Studio, you must have the following software:

- An Actuate Java Components installation with BIRT Studio license option.
- An application server or JSP or servlet engine such as Actuate embedded servlet engine or IBM WebSphere where Actuate Java Components are deployed.
- Actuate BIRT designer to create report templates.

You can find information about how to install and configure Actuate Java Components in *Installing an Actuate Java Component*.





**Figure 10-2** BIRT Studio

---

## Understanding the context root

Actuate Java Components is packaged and distributed as an web archive file. When deployed to an application server container, the context root is name of the web archive (.war) file without the .war file extension. For example, if your web archive (.war) file were named `DeploymentKit.war`, the URL to access the BIRT Studio is:

```
http://<web server>:<port>/DeploymentKit/wr
```

The context root for Actuate Java Components is the root directory of the web archive (.war) file when it is extracted.

For more information about Actuate Java Components customization refer to *Actuate BIRT Java Components Developer Guide*.



# 11

## **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, organizations typically require 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 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 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 coordination between the template designer and the

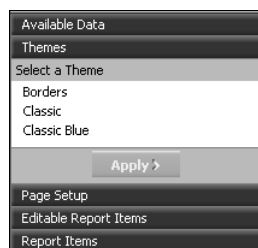
data architect to ensure that the templates and data are suitable for use with each other.

## 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 11-1.



**Figure 11-1** Themes provided with the default templates

When you create a template, consider creating different themes, so 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 need to 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.

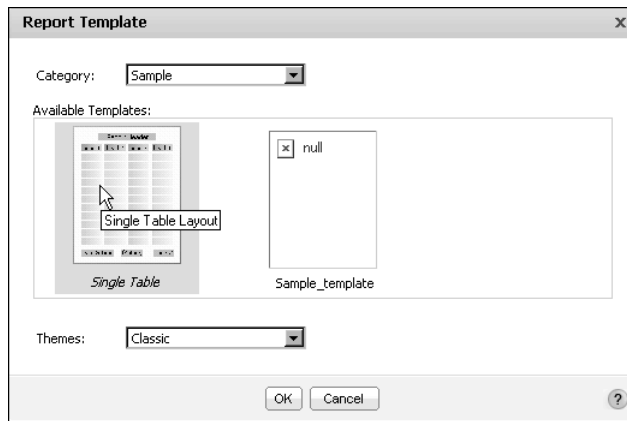
## Improving usability

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:

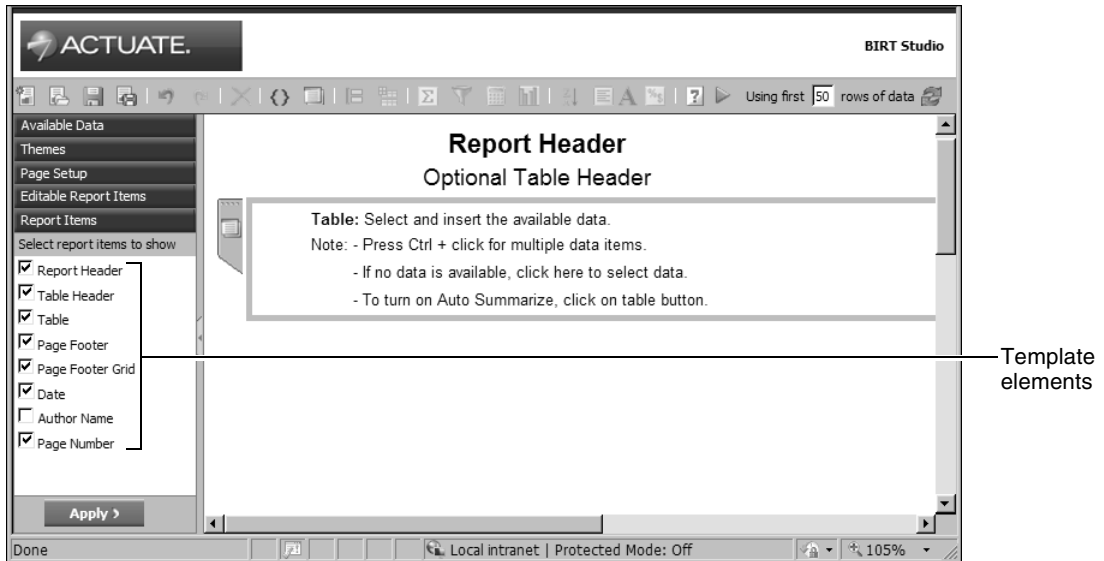
- Set the general properties of the template file:
  - The display name of the template
  - A brief description of the template
  - The image to use as the thumbnail image of the template

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 11-2 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 11-2**      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 11-3 shows one of the default templates with several elements listed under Report Items. The elements have descriptive names.



**Figure 11-3** Report Items listing all the template elements

- 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 11-3 shows a table with detailed instructions.

## Creating a report template

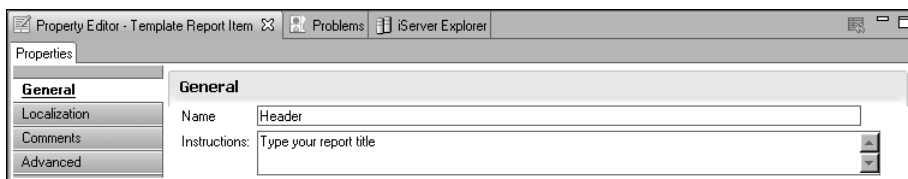
You use Actuate BIRT Designer Professional to create report templates. You design a template in the same way that you design a BIRT report. In fact, you can create a report design then save it as a template. The file-name extension for a template file is .rpttemplate. If you are creating templates for 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.

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:

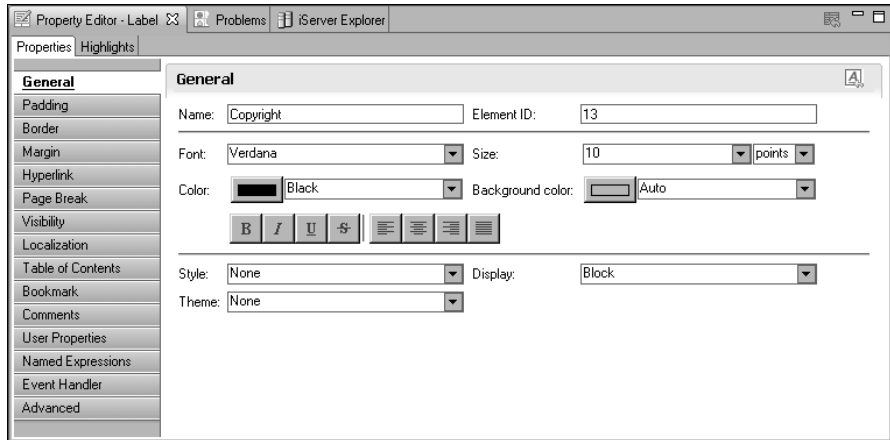
- 1 Choose File→New→Other.
- 2 On New, expand Business Intelligence and Reporting Tools, then select Template. Choose Next.
- 3 In New Template—Template, select the folder in which to create the template file, specify a file name, then choose Next.
- 4 In New Template—Set Template Property:
  - 1 In 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. You must use a path that is relative to the <context root>\resources folder. The following example is a typical example of a relative path:  
`\expense_rpt_template.jpg`  
 The path example means that BIRT Studio expects the image file to be in the following location in the file system:  
`<context root>\resources\expense_rpt_template.jpg`
  - 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. You can only edit labels and tables 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 11-4 shows an example name and instructions for using the element.



**Figure 11-4** Specifying name and instructions for an editable element

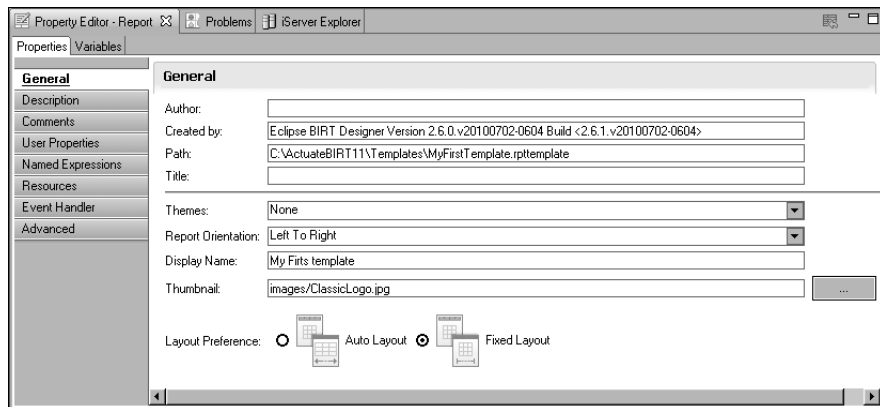


- 4 For elements that you do not want the BIRT Studio user to edit, you should also specify a descriptive name, so that the BIRT Studio user can easily identify the purpose of the element and decide whether to include the element in the report. In Property Editor, type a name for the Name property. Figure 11-5 shows setting a label's name as Copyright.



**Figure 11-5** Specifying a name for a label element that users cannot edit

You can use the BIRT Thumbnail feature to manage the thumbnail image. You can change the thumbnail image property from the General properties in Property Editor as shown on Figure 11-6.



**Figure 11-6** General properties

Selecting the thumbnail button gives you three options for selecting the thumbnail image as shown on Figure 11-7:

- Generate from report option auto-generates the image using the template design view and embeds the image in the template itself. It is a convenient

feature for template developers. In this case you do not deploy the thumbnail image, you deploy only the template.

- Browse from file system option allows you to select the image from the file system. This option uses an absolute path to the image file and you have to ensure this path is accessible when you deploy the template to your Java Components installation.
- Import from shared resources option allows you to select an image file from the BIRT designer resource folder and embed the selected image in the template design. If you use this option you do not need to publish the image file.



**Figure 11-7** Manage Thumbnail

---

## Providing data with a report template

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*.

### 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

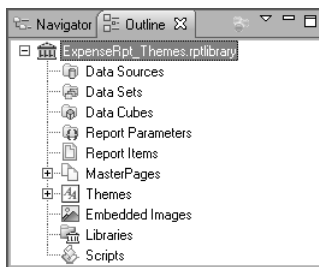
As described earlier in this chapter, you create themes to provide the user with different sets of styles to apply to a report. You use Actuate BIRT Designer Professional to create the themes in a library. After you create the library, you associate the library with the template.

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 11-8, displays the types of report elements you can add to a library.



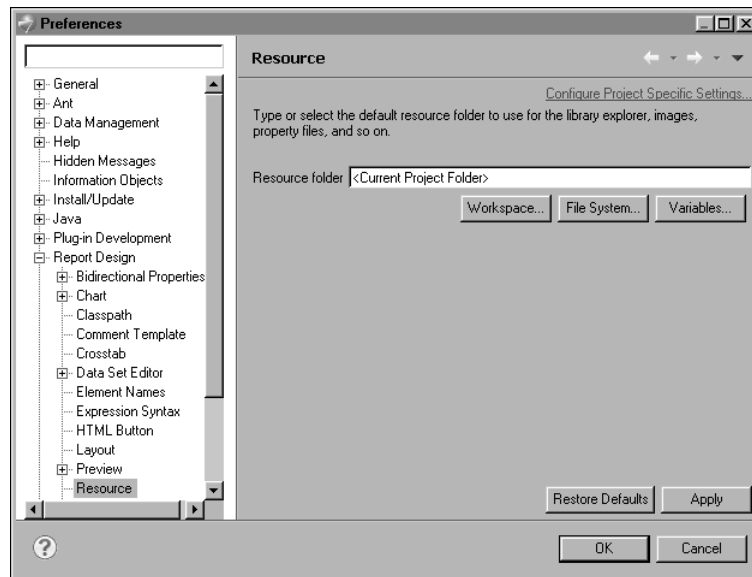
**Figure 11-8** Outline view

- 3 In the Outline view, expand Themes. A default theme, named defaultTheme, has already been created.
- 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

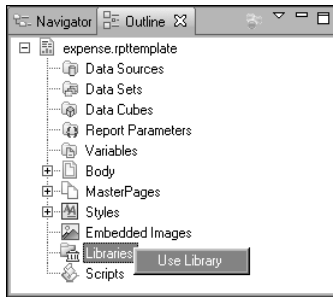
- 1 If the BIRT resource folder is not the current project folder, place the library in the BIRT resource folder, so that it is available to the template. To specify a folder as the resource folder:
  - 1 Choose Window→Preferences.
  - 2 In Preferences, expand Report Design, then choose Resource, as shown in Figure 11-9.



**Figure 11-9** Specifying the location of the resource folder in Preferences

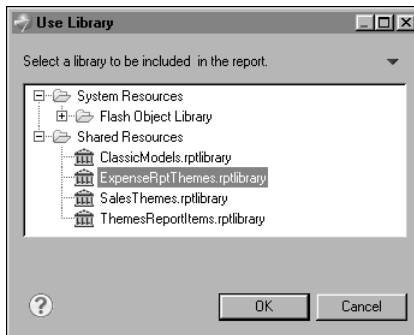
- 3 Choose File System to select a folder to use as the resource folder.
- 4 On Directory Selection, navigate to a folder on your computer or on the network, or choose Make New Folder to create a new folder.
- 5 Choose OK to confirm your folder selection. Preferences displays the path to the resource 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→Place Library in Resource Folder. Publish 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 11-10.



**Figure 11-10** 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 11-11 shows an example of Use Library.



**Figure 11-11** Displaying libraries in the resource folder

- 5 Select the library that contains the themes you want to use with the template, then choose OK.

---

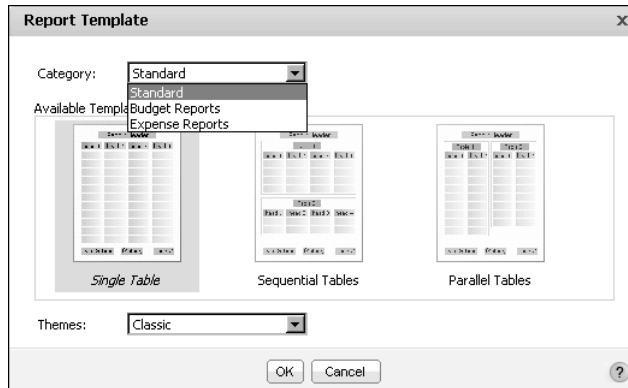
## Publishing a template

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 two categories named Standard and Sample. 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 11-12 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.



**Figure 11-12** Report Template displays three template categories

Template files must be copied to the appropriate folders under the BizRDRptTemplates folder. You must have direct access to the machine on which Actuate BIRT Java Components run, because you need to manually place the template files in the following location on the file system:

```
<context root>\resources\BizRDRptTemplates
```

### How to publish a report template

- 1 Create the proper folder structure on the machine where Actuate BIRT Java Components runs. If you want to organize the templates in categories, create one folder under the BizRDRptTemplates folder for each category. For

example, to display templates in categories named Sales and Marketing, you create the following folders:

```
<context root>\resources\BizRDRptTemplates\Sales
<context root>\resources\BizRDRptTemplates\Marketing
```

- 2 Place the template files into their category folders.
- 3 Restart your Java Component application every time you create a new category for templates.

---

## Setting the default template category

By default, the Report Template dialog displays the templates in the predefined category, Standard, as shown earlier in Figure 11-12. 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 the following location:

```
<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 your Java Component application 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.

All resources used by a template must be copied to the Actuate Java Components resource folder in the following location:

```
<context root>\resources
```

You do not need to restart Actuate Java Components for the resource files to take effect. All you need to do is to log out of Actuate BIRT Java Components, then log in again.



# 12

## Providing data

This chapter covers the following topics:

- Types of data sources
- Creating a custom data source

---

## Types of data sources

To create a new report, the first step the BIRT Studio user performs is selecting 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 either of the following ways:

- 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 the second method. The first method is described in Chapter 11, “Creating and publishing report templates.”

---

## Creating a custom data source

You can provide users access to data by creating a custom user interface to a data source and integrating it 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 12-1.

**Listing 12-1** 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.JdbcS
electDataSet</datasetExtensionId>
  <datasetDisplayName>CLASSICMODELS.PAYMENTS</datasetDisplayName>

  <odaDriverClass>org.eclipse.birt.report.data.oda.sampledb.Drive
r</odaDriverClass>
  <odaURL>jdbc:classicmodels:sampled</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>true</enabledInWorkgroupMode>
  <enabledInEnterpriseMode>>false</enabledInEnterpriseMode>
  <entryPoint></entryPoint>
</odaconfig>

```

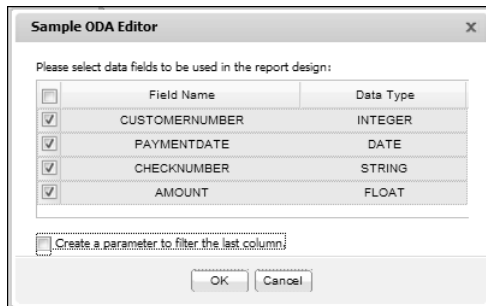
The Sample ODA data source with BIRT Studio is enabled by default. The setting of the data source's `<enabledInWorkgroupMode>` attribute in `erni_config.xml` is set to `true`. The `erni_config.xml` file is stored in the following location:

```
<context root>\WEB-INF
```

The context root for Actuate Java Components is the root directory of the web archive (.war) file when it is extracted.

If you create a new data source you must always enable the sample data source, and then restart the application server, and open BIRT Studio.

You test this custom data source by starting BIRT Studio. When prompted, select one of the standard report templates. 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 12-1. The check boxes appearing next to the column names allow the user to select the data for the report. The editor gives the user the ability to create a parameter for filtering the data in the last column.



**Figure 12-1** Selecting data in sample ODA editor

## Configuring a sample ODA data source

Table 12-1 lists the configuration attributes and their descriptions.

**Table 12-1** Attributes of a sample ODA data source

Attribute	Description
<code>&lt;name&gt;</code>	The unique name of the data source. This name follows certain naming conventions. There can be no spaces, for example.

**Table 12-1** Attributes of a sample ODA data source

Attribute	Description
<displayName>	The data source name that appears in the Data Source dialog box, shown in Figure 12-2.
<description>	The data source description that appears in the user interface.
<datasetExtensionId>	Identifies the ID of the data set extension of the data driver.
<datasourceExtensionId>	Identifies the ID of the data source extension of the data driver.
<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>	Indicates whether this data source is available to users.
<enabledInEnterpriseMode>	Not used.
<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 12-2 shows portions of the report XML that contain the configuration attributes and their values.

#### Listing 12-2 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. When you use BIRT drivers you skip this step.
- 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 `<enabledInWorkgroupMode>` attribute in `erni_config.xml` to true. The `erni_config.xml` file is stored in the following location:

```
<context root>\WEB-INF
```

Listing 12-3 shows in bold the `<enabledInWorkgroupMode>` attribute whose value you change from false to true. Look for the ODA data source named OdaSampleExt.

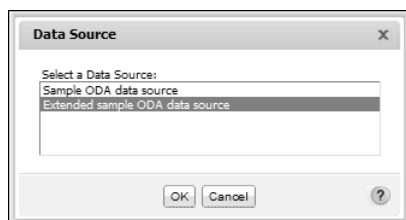


### Listing 12-3     Configuring the extended sample data source in erni\_config.xml

```
<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>

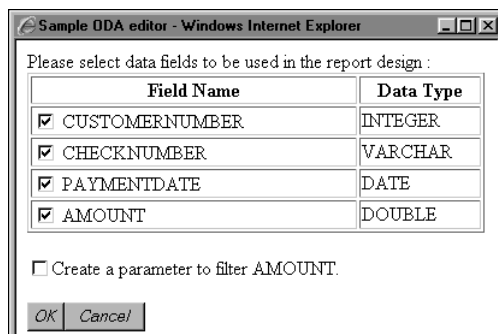
  <datasetExtensionId>org.eclipse.birt.report.data.oda.jdbc.JdbcS
electDataSet</datasetExtensionId>
  <enabledInWorkgroupMode>true</enabledInWorkgroupMode>
  <enabledInEnterpriseMode>true</enabledInEnterpriseMode>
  <entryPoint>/OdaSample</entryPoint>
</odaconfig>
```

After you enable the extended sample data source, restart the application server, and open BIRT Studio. When prompted, select one of the standard report templates. The Data Source dialog box, shown in Figure 12-2, prompts you to select a data source. Select Extended sample ODA data source.



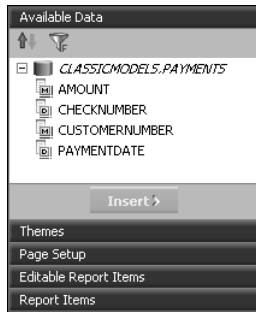
**Figure 12-2**     Selecting data source

A query builder page, shown in Figure 12-3, prompts you to select the data to use in the report.



**Figure 12-3**     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 12-4.



**Figure 12-4** 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](http://www.eclipse.org/datatools/project_connectivity/connectivity_doc/OdaOverview.htm)

Another resource is the Addison-Wesley's *Integrating and Extending BIRT* book.

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 folder:

```
<context root>\WEB-INF\platform\dropins
```

## 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 Actuate Java Component. 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 12-3. 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 12-4.

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 12-2 describes the supported parameters.

**Table 12-2** 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 12-3 describes the actions taken with the different inedit and state values.

**Table 12-3** 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 the data fields. 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 application server you use. For example, if you use Apache Tomcat 6.0 as an application server, you can use `servlet-api.jar` from the following location:

```
Tomcat 6.0\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 `portal.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 12-3 shows the definition of the extended data source. To add your custom data source, create a new `<odaconfig>` element in `erni_config.xml`. Table 12-4 describes each attribute, within the `<odaconfig>` element, that you configure.

**Table 12-4** Attributes of an extended custom data source

Attribute	Description
<code>&lt;name&gt;</code>	The unique name of the data source. This name follows certain naming conventions. There can be no spaces, for example.
<code>&lt;displayName&gt;</code>	The data source name that appears in the Data Source dialog box, shown in Figure 12-2.
<code>&lt;description&gt;</code>	The data source description that appears in the user interface.
<code>&lt;datasetExtensionId&gt;</code>	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.
<code>&lt;datasourceExtensionId&gt;</code>	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.
<code>&lt;enabledInWorkgroupMode&gt;</code>	Indicates whether this data source is available to users.
<code>&lt;enabledInEnterpriseMode&gt;</code>	Not used.
<code>&lt;entryPoint&gt;</code>	A URL or servlet that points to the first web page of your custom query builder.

# Configuring BIRT Studio

This chapter covers the following topics:

- Enabling or disabling functionality
- Configuring the application environment

---

## Enabling or disabling functionality

BIRT Studio provides a full range of tools to support the report design process. 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, and aggregating data. 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 the following location:

```
<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 13-1 shows the hierarchy of elements.

**Listing 13-1** 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>
    <action>
        <name>Parameter</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 web application. 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 13-1 lists the 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 13-1** User actions 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

*(continues)*

**Table 13-1** User actions you can enable or disable through erni\_config.xml (continued)

Action Set	Action	Description
Calculations (continued)	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
	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	Deletes a column from a table in the report
EditText	EditText	Enables editing of the selected static text

**Table 13-1**      User actions you can enable or disable through  
                         erni\_config.xml (continued)

Action Set	Action	Description
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 the 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
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

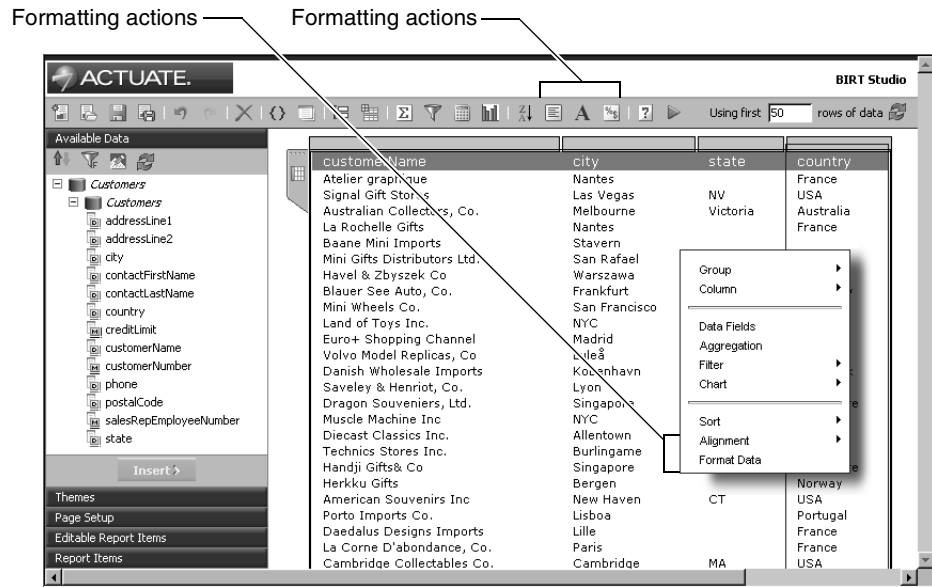
(continues)

**Table 13-1** User actions you can enable or disable through erni\_config.xml (continued)

Action Set	Action	Description
Grouping (continued)	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 13-1 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.



**Figure 13-1** Default BIRT Studio page

Listing 13-2 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 13-2** Visibility of the Formatting action set, changed to false

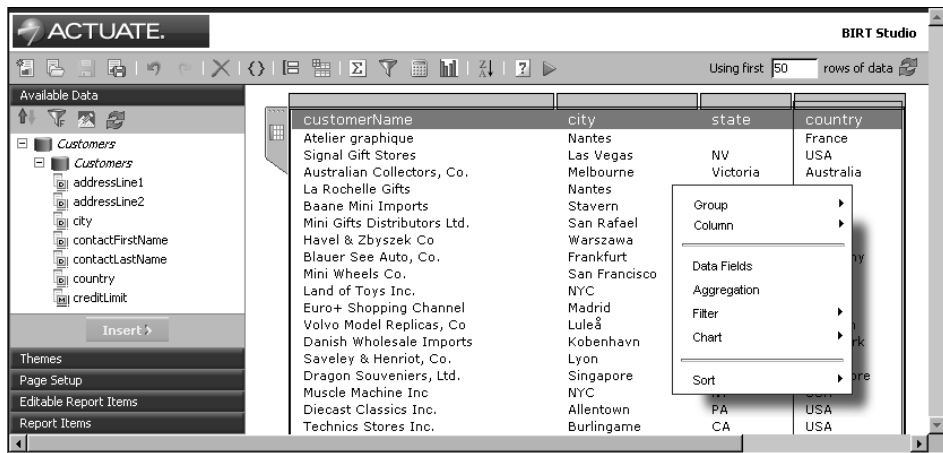
```
<actionSet>
  <name>Formatting</name>
  <visible>false</visible>
  <action>
    <name>AlignLeft</name>
    <enabled>true</enabled>
  </action>
  <action>
    <name>AlignCentert</name>
    <enabled>true</enabled>
  </action>
  <action>
    <name>AlignRight</name>
    <enabled>true</enabled>
  </action>
```

```

<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 13-2 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 13-2** Updated toolbar and context menu, without any formatting functions

Listing 13-3 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 13-3** 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 13-3 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.

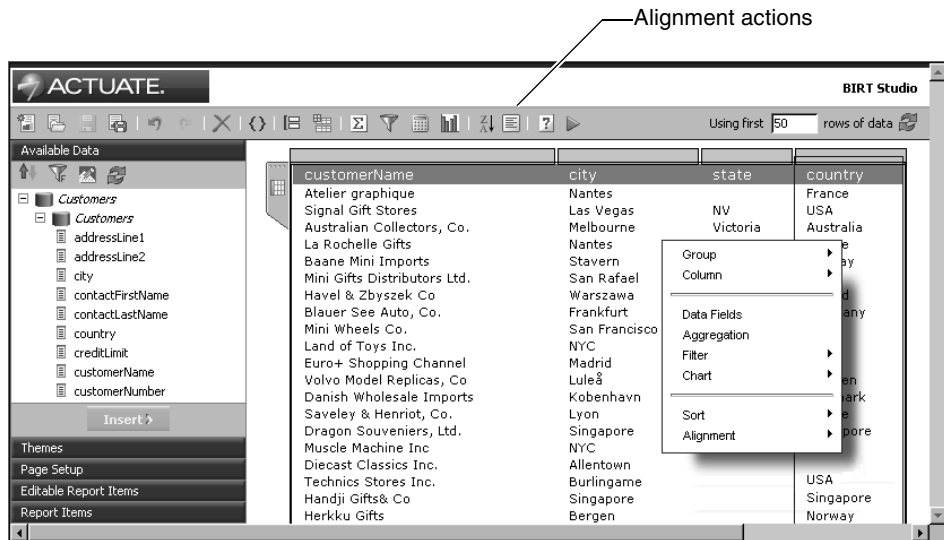
## Specifying the default position of aggregate values

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>

```



**Figure 13-3** Updated toolbar and context menu, without the font and data formatting actions

## 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 any Java components packaged with it in a WAR file. Table 13-2 describes the configuration parameters for BIRT Studio.

**Table 13-2** BIRT Studio web.xml parameters

Parameter name	Description
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.
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.



**Table 13-2** BIRT Studio web.xml parameters (continued)

Parameter name	Description
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).
DEFAULT_WORKGROUP_FUNCTIONALITY_ROLE	Specifies the functionality role for all users of BIRT Studio, and the BIRT Viewer. The default value is Active Portal Intermediate.
ENABLE_CASE_SENSITIVE_STRING_FILTER	Specifies whether the filter dialog considers case. TRUE specifies case-sensitivity for example CA matches CA only, not ca or Ca. The default value is false
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	<p>The number of values shown in the parameter dialog box for a dynamic value parameter in BIRT Studio:</p> <ul style="list-style-type: none"> <li>■ A positive number value N means only the first N values appear in the parameter dialog box.</li> <li>■ A value of 0 means all values from the data source appear in the parameter dialog. 0 is the default value.</li> <li>■ 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.</li> </ul>
	MAX_NUMBER_OF_VALUES_FOR_DYNAMIC_PARAMETER only applies to a dynamic value parameter. 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.

*(continues)*

**Table 13-2** BIRT Studio web.xml parameters (continued)

Parameter name	Description
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.
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.

# 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
- BIRT Studio file manager URIs overview
- BIRT Studio file manager URIs quick reference
- Common URI parameters
- BIRT Studio file manager URIs reference
- Actuate BIRT Viewer URIs reference

---

## 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 <context root>\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 a repository parameter in order to operate. Table 14-1 lists and describes the URI parameter for the BIRT Studio servlet.

**Table 14-1** BIRT Studio URI parameters

URI parameter	Description
repositoryType	The repository type. Use workgroup for the local file system.

---

## Using the BIRT Studio URLs

You can log in to BIRT Studio by typing a URL in a web browser's address field. In addition to the initial BIRT Studio page, you can open BIRT Studio with a specific report design.

In the example URLs in the following topics, special characters are represented by codes, as shown in Table 14-2.

**Table 14-2** Codes for special characters in URLs

Character	Code
Colon (:)	%3a
Slash (/)	%2f
Period (.)	%2e
Space ( )	%20

### 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://localhost:8080/DeploymentKit/wr?
__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.

---

## BIRT Studio file manager URLs overview

The following sections describe the Java Component BIRT Studio JSPs that manage files and folders. For a repository managed with iServer, the Information Console provides this functionality. For information about Information Console URLs, see *Information Console Developer Guide*.

The following sections provide quick reference tables and detailed reference information about Actuate BIRT Studio URIs. An Actuate BIRT Studio URI is a directive to Actuate BIRT Studio to perform an action, such as showing a list of files, rather than change the appearance of the application.

File manager pages use the .do extension for the Struts action mapping to a page. The complete page name appears as part of the reference material. BIRT Studio page and folder names are case sensitive.

---

## BIRT Studio file manager URIs quick reference

Table 14-3 lists the BIRT Studio file manager URIs.

**Table 14-3** Actuate BIRT Studio URI pages

BIRT Studio page	Description
banner page	Displays a banner at the top of each BIRT Studio file manager page.
browse file page	Provides file and folder browsing functionality for the submit request pages.
browse page	See browse file page.
delete file status page	Displays whether a file was successfully deleted.
detail page	Supports error handling and presenting object details.
drop page	Supports deleting files or cancelling running jobs.
error page	Retrieves an error message from the exception or the request and displays it.
execute report page	Submits a run report job request to the server.
executereport page	See execute report page.
getfiledetails page	See file or folder detail page.
getfolderitems page	See file and folder index page.
list page	Supports listing files and folders.
page not found page	Displays an error message when a JSP is unavailable in BIRT Studio file manager.
parameters page	Presents a list of the request parameters.

---

## Common URI parameters

All Actuate BIRT Studio file manager URIs have the parameters shown in Table 14-4. String values that are too long are truncated for all parameters. The web browser that you use determines the length of parameters.

**Table 14-4** Common Actuate BIRT Studio URI parameters

URI parameter	Description
serverURL	Contains the URI that accesses the Actuate web application, such as <code>http://Services:8000.&gt;</code>
timezone	The current user's time zone.
volume	The volume to which the user is connected.

---

## BIRT Studio file manager URIs reference

This section provides the detailed reference for BIRT Studio file manager URIs. In the definitions, `<context root>` represents the name of your Actuate BIRT Studio context root.

Table 14-5 lists the topics this chapter covers and the file names discussed in each topic. All pages are under the BIRT Studio context root.

**Table 14-5** Actuate BIRT Studio pages

Topic	BIRT Studio file
banner page	<code>iportal\activePortal\private\common\banner.jsp</code>
browse file page	<code>browsefile.do</code> <code>iportal\activePortal\private\query\browse.jsp</code>
delete file status page	<code>iportal\activePortal\private\filesfolders\deletefilestatus.jsp</code>
detail page	
■ error detail page	<code>iportal\activePortal\errors\detail.jsp</code> <code>getfiledetails.do</code>
■ file or folder detail page	<code>iportal\activePortal\private\filesfolders\filedetail.jsp</code>
error page	<code>errors\error.jsp</code> <code>iportal\activePortal\private\common\errors\error.jsp</code>

*(continues)*

**Table 14-5** Actuate BIRT Studio pages (continued)

Topic	BIRT Studio file
execute report page	executereport.do
file or folder drop page	deletefile.do
file and folder index page	getfolderitems.do iportal\activePortal\private\filesfolders\filefolderlist.jsp
file and folder list page	getfolderitems.do iportal\activePortal\private\filesfolders\filefolderlist.jsp
page not found page	iportal\activePortal\errors\pagenotfound.jsp
parameters page	iportal\activePortal\private\newrequest\parameters.jsp
viewer page for Actuate BIRT Reports and Business Reports	IVServlet

## banner page

Provides the banner that appears across the top of all Actuate BIRT Studio web pages. The default banner displays the Actuate logo, user name, cluster name, and volume name, and provides links for Logout, Options, and Help. The banner page obtains the user name, cluster name, and volume name from variables maintained by the authenticate page.

<b>Name</b>	<context root>\iportal\activePortal\private\common\banner.jsp
<b>Used by</b>	iportal\activePortal\private\login.jsp iportal\activePortal\private\channels\channelnoticelist.jsp iportal\activePortal\private\channels\channeloperationstatus.jsp iportal\activePortal\private\filesfolders\deletefilestatus.jsp iportal\activePortal\private\filesfolders\filedetail.jsp iportal\activePortal\private\filesfolders\filefolderlist.jsp iportal\activePortal\private\jobs\getjobdetails.jsp iportal\activePortal\private\jobs\joboperationstatus.jsp iportal\activePortal\private\jobs\selectjobs.jsp iportal\activePortal\private\newrequest\newrequest.jsp iportal\activePortal\private\newrequest\newrequest2.jsp iportal\activePortal\private\newrequest\submitjobstatus.jsp iportal\activePortal\private\options\options.jsp



iportal\activePortal\private\query\create.jsp  
 iportal\activePortal\private\query\execute.jsp

---

## browse file page

Contains file and folder browsing functionality used by other submit request pages.

**Name** <context root>\browsefile.do

<context root>\iportal\activePortal\private\query\browse.jsp

**Parameters** workingFolder is the name of the folder for which to display contents in the folder browser window. The browse file page also uses the common URI parameters.

**Used by** iportal\activePortal\private\newrequest\browse.jsp  
 iportal\activePortal\private\query\browse.jsp

---

## delete file status page

Summarizes the result of a deletion performed by the drop page and indicates whether a file was successfully deleted. The delete file status page includes authenticate to obtain user session data. BIRT Studio performs the deletion as part of an action and then forwards to the delete file status page.

**Name** <context root>\iportal\activePortal\private\filesfolders\deletefilestatus.jsp

**Used by** Not applicable

---

## detail page

Displays detailed information about Repository objects. There are two detail pages:

<context root>\iportal\activePortal\errors

<context root>\iportal\activePortal\filesfolders

### error detail page

Provides a template error page that can be embedded in another page.

**Name** <context root>\iportal\activePortal\errors\detail.jsp

**Used by** iportal\activePortal\private\common\errors\error.jsp

drop page

```
iportal\activePortal\viewer\print.jsp
iportal\activePortal\viewer\saveas.jsp
iportal\activePortal\viewer\searchframe.jsp
iportal\activePortal\viewer\viewdefault.jsp
iportal\activePortal\viewer\viewtoc.jsp
```

file or folder detail page

Displays detailed information about the selected viewable folder or file. Users request file details by choosing the magnifying glass icon to the right of files listed on the folder page, or folder details by choosing the magnifying glass icon to the right of the folder name in the breadcrumb. Users can request another viewable document or delete the current file or folder from the file or folder detail page. filedetail.jsp uses the HTML code in <context root>\iportal\activePortal\private\filesfolders\filedetailcontent.jsp to display the information.

**Name** <context root>\getfiledetails.do

<context root>\iportal\activePortal\private\filesfolders\filedetail.jsp

**Parameters** Table 14-6 describes the parameters for the file or folder detail page. The file or folder detail page also uses the common URI parameters.

**Table 14-6** File or folder detail URI parameters

URI parameter	Description
name	The full path name of the repository object for which to show details. This parameter is ignored if objectID is also specified.
objectID	The repository object’s unique identifier.
version	The repository object’s version number. The default is the latest version.

**Used by** Not applicable

drop page

Deletes one or more files or folders

file or folder drop page

Deletes the specified file or folder. The file or folder drop page includes the banner page to obtain user session data.

**Name** <context root>\deletefile.do

**Parameters** Table 14-7 describes the parameters for the file or folder drop page. The file or folder drop page also uses the common URI parameters.

**Table 14-7** File or folder drop URI parameters

URI parameter	Description
ID	The unique identifier of the repository object to delete.
name	The full path name of the repository object to delete. Multiple name parameters, to delete more than one file or folder at a time, are allowed. This parameter is ignored if ID is also specified.
redirect	URI to which to redirect the job deletion page. The default redirect page is processedaction_status.

**Used by** Not applicable

---

## error page

Displays the specified error message. BIRT Studio uses two pages. All BIRT Studio code uses <context root>\iportal\activePortal\private\common\errors\error.jsp.

**Name** <context root>\iportal\activePortal\errors\error.jsp  
<context root>\iportal\activePortal\private\common\errors\error.jsp

**Used by** iportal\activePortal\private\login.jsp  
iportal\activePortal\private\common\closewindow.jsp  
iportal\activePortal\private\common\sidebar.jsp  
iportal\activePortal\private\common\errors\errorpage.jsp  
iportal\activePortal\private\options\options.jsp  
iportal\activePortal\private\query\create.jsp  
iportal\activePortal\private\query\execute.jsp  
iportal\activePortal\private\templates\template.jsp  
iportal\activePortal\viewer\closewindow.jsp  
iportal\activePortal\viewer\print.jsp  
iportal\activePortal\viewer\saveas.jsp  
iportal\activePortal\viewer\searchframe.jsp  
iportal\activePortal\viewer\searchreport.jsp  
iportal\activePortal\viewer\viewframeset.jsp

---

# execute report page

Submits a run report job request.

When executing a report job or query, a Cancel button appears after a specified wait time passes. You can change the time by setting the EXECUTE\_REPORT\_WAIT\_TIME configuration parameter in the appropriate BIRT Studio configuration file.

For reports that accept run-time parameters, you can set the parameter in the URL by adding an ampersand (&), the parameter name, and an equal(=) sign, followed by the parameter value in quotes.

**Name** <context root>\executereport.do

**Parameters** Table 14-8 describes the parameters for the execute report page. The execute report page also uses the common URI parameters.

**Table 14-8**      Execute Report URI parameters

URI parameter	Description
__ageDays	Use with __ageHours to determine how long output objects exist before they are automatically deleted. Use only if __archivePolicy is set to Age. __ageDays can be any positive number.
__ageHours	Use with __ageDays to determine how long output objects exist before they are automatically deleted. Use only if __archivePolicy is set to Age. __ageHours can be any positive number.
__executableName	The name of the executable file for this request.
invokeSubmit	Controls whether the browser is redirected to the parameter screen or whether the report job is run immediately. If True, the report job is executed without displaying the parameters. If False, the parameters are displayed. False is the default.
__outputDocName	The name and path of the resulting BIRT document. This parameter is only usable for BIRT reports when the BIT_SAVE_REPORT_DOCUMENT_ENABLED parameter is set to TRUE in web.xml.  If the given path is absolute, then executereport saves the report to that path. If the given path is relative, then executereport saves the report to the path set in the BIT_SAVE_REPORT_DOCUMENT_PATH web.xml parameter.

**Table 14-8**      Execute Report URI parameters

URI parameter	Description
__priority	Specifies the job submission priority. Values are High, Medium, and Low.
__priorityValue	Specifies a number ranging from 1 to 1000 and corresponding to the job submission priority. You can only specify values allowed by your functionality level.
__progressive	Indicates whether to display the report document after it generates. If False, the report document displays after it generates. If True, the report document displays progressively, as it generates.
__serverURL	Contains the URI that accesses the JSP engine, such as http://Services:8900.
__wait	If "wait", BIRT Studio waits for the report generation to be completed before displaying it. If "nowait", BIRT Studio displays the first page right away even if the report job is not completed.

For example, the following URL executes the Sales By Territory.rptdesign report immediately with the Territory run-time parameter set to EMEA:

```
http://localhost:8080/ActuateJavaComponent/executereport.do?
__requesttype=immediate&__executableName=/Public
/BIRT%20and%20BIRT%20Report%20Studio%20Examples
/Sales%20by%20Territory.rptdesign&userid=anonymous&
__saveOutput=false&Territory=%22EMEA%22&invokeSubmit=True
```

The following parameter names are reserved for internal use only by the execute report page:

- doframe
- inputfile
- jobType
- name
- selectTab

**Used by**    Not applicable.

---

# index page

Provides the entry point and structure for the parts of BIRT Studio generated from multiple files.

## file and folder index page

The default entry point to the BIRT Studio web application. The file and folder index page provides the entry point and structure to support the Files and Folders functionality. The structure is a table that BIRT Studio uses to format and present files and folders data. Page content varies depending on the BIRT Studio directive.

The file and folder index page uses the banner page to provide the reporting web page banner. filefolderlist.jsp uses the HTML code in <context root>\iportal\activePortal\private\filesfolders\filefolderlistcontent.jsp to display files and folders data.

<b>Name</b>	<context root>\getfolderitems.do  <context root>\iportal\activePortal\private\filesfolders\filefolderlist.jsp
<b>Parameters</b>	Table 14-9 describes the parameters for files and folders index page. The files and folders index page also uses the common URI parameters.

**Table 14-9** File and folders index URI parameters

URI parameter	Description
startUpMessage	Specifies a message to appear when BIRT Studio calls this page.
subpage	Specifies the content of the page. Possible values are: <ul style="list-style-type: none"><li>■ _list: include list</li><li>■ _detail: include detail</li></ul> Specifying any other value for subpage invokes the page not found page.

---

# list page

Lists files in a container, such as a folder.

## file and folder list page

Presents a list of objects that reside in the current working repository folder. Users request folder listings by choosing links on the reporting web page. The file and

folder list page includes a filter section where users specify criteria for viewing report documents.

When users access a repository for the first time, BIRT Studio displays their home folder, if they have one, or the top folder in the repository. All files and folders in that folder that they have permission to view appear in the BIRT Studio listing page. Users can specify a filter to choose the types of files to view.

**Name** <context root>\getfolderitems.do

<context root>\portal\activePortal\private\filesfolders\filefolderlist.jsp

**Parameters** Table 14-10 describes the parameters for the file and folder list page. The file and folder list page also uses the common URI parameters.

**Table 14-10** File and folder list URI parameters

URI parameter	Description
applyFilter	If True, apply filter. If False, filter not applied.
filter	The filter specifying the file and folder names to list. Filter is a string. The default is "".
folder	The folder for which to list the contents. Folder name is a string. If no folder is specified, List uses the last working folder known for the session if cookies are enabled. If cookies are not enabled, List uses the user's home folder as specified in the user settings.
onlyLatest	If True, show only the latest version of a file if multiple versions exist. If False, show all versions of a file if multiple versions exist. The default is False.
resetFilter	Any non-null value for resetFilter causes the filter to return to its original state. Users can reset the filter by choosing the Default button on the listing page.
showDocument	If True, show all viewable documents. If False, do not show viewable documents. The default is True.
showExecutables	If True, show all report executables. If False, do not show report executables. The default is True.
showFolders	If True, show all folders. If False, do not show folders. The default is True.

**Used by** Not applicable.

---

# page not found page

Displays an error message when BIRT Studio cannot find the page that a user specifies. This page is a BIRT Studio page only.

- Name** <context root>\portal\activePortal\errors\pagenotfound.jsp
- Used by** Not applicable.

---

# parameters page

Displays job parameters. Users access the parameters list by choosing Save and View for a .rptdesign that contains parameters.

Parameters looks like Figure 14-1.

- Name** <context root>\portal\activePortal\private\newrequest\parameters.jsp
- Used by** portal\activePortal\private\newrequest\newrequestpage.jsp



**Figure 14-1** Parameters page

---

# Actuate BIRT Viewer URIs reference

To view and interact with Actuate BIRT reports, you use the Actuate BIRT servlet. All BIRT Viewer options and varieties use the same URL. For detailed information about the BIRT servlet URL, see *Working with Actuate BIRT Viewers*.



# Index

## Symbols

- ^ operator 221
- \_ (underscore) character 85, 207
- : (colon) character 273
- ! (exclamation point) character 28
- ? (question mark) character 52, 214
- . (period) character 273
- ' (single quotation mark) character 52
- " (double quotation mark) character 52, 206, 214
- [ ] (brackets) characters 47, 52
- \* (asterisk) character 214
- \* operator 221
- / (forward slash) character 273
- / operator 221
- \ (backslash) character 85, 207
- & operator 221
- % (percent) character 85, 206
- % operator 221
- + operator 220
- < operator 221
- <= operator 221
- <> operator 221
- = operator 221
- > operator 221
- >= operator 221
- operator 220

## Numerics

- 2D With Depth setting 177
- 3-dimensional charts 176

## A

- ABS function 192
- absolute paths 238, 280
- absolute values 192
- access privileges. *See* privileges
- accessing
  - BIRT Studio 4
  - BIRT Studio functionality 18, 228, 260
  - calendar 84
  - chart editor 145

- data 238, 248
- Data Analyzer 140
- ODA sample data source 252
- report items 18
- report templates 6
- reports 228
- repository items 282
- resources 244
- web applications 282
- aciportal.jar 257
- action element 260
- action mapping 274
- action sets 261, 266
- actions 256, 260, 261, 273
- actionSet element 260
- actionSets element 260
- Actuate Java Components. *See* Java Components application
- Add Data Field link 50
- Add Group button 60, 61
- Add Section button 65
- ADD\_DAY function 52, 192
- ADD\_HOUR function 193
- ADD\_MINUTE function 193
- ADD\_MONTH function 194
- ADD\_QUARTER function 194
- ADD\_SECOND function 195
- ADD\_WEEK function 195
- ADD\_YEAR function 195
- adding
  - aggregate data 69
  - borders 40
  - charts 64, 119, 144, 145, 183
  - column headers 38, 48
  - computed columns 47
  - cross tabs 131
  - data fields 12, 19, 41, 49
  - data groups 58, 60, 61, 62
  - data rows 39
  - filter conditions 79, 80, 86, 90
  - filter parameters 94, 97, 109
  - filters 79, 86, 92
  - formatting rules 29, 30, 33

- adding (*continued*)
  - images 234
  - multiple fields 50
  - page breaks 74, 264
  - report elements to libraries 240
  - report elements to templates 232, 234, 236
  - report items 18, 236
  - report sections 63, 65
  - styles 233, 240
  - summary tables 112, 117
  - themes to templates 19, 233, 239
  - Tooltips 98, 99, 172
- addition 219
- addresses 36
- administrators 4
- Advanced Filter dialog box
  - changing conditions and 91
  - comparing values and 85
  - creating filters and 92
  - defining multiple conditions and 87
  - removing conditions and 91
- Advanced Sort command 57
- Advanced Sort dialog box 57, 58
- AdvancedSort action 264
- After property 75
- ageDays parameter 280
- ageHours parameter 280
- aggregate data
  - building cross tabs for 131, 132
  - building data cubes for 131
  - calculating values for 49, 69, 218
  - creating charts and 144
  - creating summary reports and 72
  - creating summary tables and 112, 120
  - displaying 68, 69, 112, 137
  - specifying placement of 267
- aggregate functions
  - See also* specific function
  - data types and 69
  - listed 218
  - selecting 70
  - summary tables and 112, 125
- aggregate values. *See* aggregate data
- aggregating data 68–70, 72
- Aggregation action 261
- Aggregation button 69
- Aggregation dialog box 69
- AlignCenter action 263
- aligning text 22, 34
- AlignLeft action 263
- alignment options 22, 126
- AlignRight action 263
- analyzing data 140, 142
- AND operator 221
- And operator 89
- animation properties 190
- application servers 229
- applications 4, 254, 257, 268, 282
- applyFilter parameter 283
- area chart subtypes 146, 147
- area charts 146, 158
  - See also* charts
- ascending sort order 57
- asterisk (\*) character 214
- attribute columns 112
  - See also* summary tables
- author names 19
- Auto Expand Width setting 43, 174
- auto suggest option 96
- AutoSummarizeOn action 264
- Available Data pane 12, 134
- AVERAGE function 69, 218
- axes values. *See* category series; value series

## B

- backslash (\) character 85, 207
- banner page 274, 276
- banner.jsp 275
- bar chart subtypes 150, 155
- bar charts 146, 148, 158
  - See also* charts
- Before property 75
- BETWEEN function 196
- Between operator 222
- binary files 272
- BIRT Designer Professional 232
- BIRT Interactive Viewer
  - See also* BIRT Viewer
- BIRT reports 284
  - See also* reports
- BIRT servlet 284
- BIRT Studio
  - accessing functionality 18, 228, 260

- adding data sources to 238
  - configuring 268
  - creating custom data sources for 252, 258
  - customizing 4, 260, 264
  - defining data sets for 238
  - designing reports and 4, 232
  - disabling actions for 261, 266
  - enabling ODA drivers for 254
  - logging in to 273
  - running 228
  - running sample ODA data source for 252
  - starting 4, 272
  - BIRT Studio file manager 273, 274, 275
  - BIRT Studio function reference 192
  - BIRT Studio servlet 272
  - BIRT Viewer 15, 284
    - See also* BIRT Interactive Viewer
  - BIRT\_RESOURCE\_PATH parameter 268
  - BIRT\_SCRIPT\_LIB\_PATH parameter 268
  - BIT\_SAVE\_REPORT\_DOCUMENT\_ENABLED parameter 280
  - BizRDRptTemplates folder 243
  - blank cells 132
  - blank characters 47, 215, 216
  - Boolean expressions 203, 207, 209
  - Border action 263
  - Border command 40
  - Border dialog box 40
  - borders 40, 119
  - Bottom N operator 83, 222
  - Bottom Percent operator 83, 222
  - brackets ( [ ] ) characters 47, 52
  - branding 233
  - browse file page 274, 277
  - browse page. *See* browse file page
  - browse.jsp 275
  - browsers. *See* web browsers
  - browsing 277
- ## C
- cache 269
  - calculated columns. *See* computed columns
  - calculated data 46, 49, 192
  - Calculation action 261
  - Calculation dialog box 48
  - calculations 46, 47, 48, 69, 132
  - calculations action set 261
  - calendar 31, 84
  - candlestick chart subtypes 155
    - See also* stock charts
  - capitalization 27, 216
  - case 27, 207, 216
  - case sensitivity
    - enabling or disabling 269
    - field names 47
    - folder names 274
    - page names 274
    - pattern matching 206
    - servlet names 272
  - categories (templates) 6, 7, 243, 244, 269
  - Category (X) Axis Title property 176
  - category series (charts) 156, 178, 180
    - See also* x-axis values
  - category values (data cubes) 131
  - CEILING function 197
  - cells (cross tabs) 132
  - ChangeSubtotal action 261
  - changing
    - actions 261
    - chart titles 175
    - column headers 19
    - data 29, 94
    - filter conditions 91
    - font properties 21
    - grouping order 62
    - page layouts 42
    - report designs 13, 34, 269
    - reports 18, 34, 94
    - servlets 272
    - styles 233
    - summary tables 117–120
    - text 19, 27, 173
    - text alignment 22
  - character formats. *See* string formats
  - character strings. *See* strings; substrings
  - characters
    - calculating data values and 52
    - counting 206
    - enabling auto suggest for 96
    - entering URLs and 273
    - finding matching 82, 84, 206, 207, 214
    - formatting string values and 27, 28
    - naming report files and 14

- characters (*continued*)
  - referencing data fields and 47, 53
  - removing leading or trailing 47, 215, 216
- Chart action 261
- chart builder 173
- chart editor 145
- Chart icon 145
- chart reports 143, 190
- Chart Size property 174, 175
- chart subtypes. *See* specific subtype
- Chart Title property 175
- chart titles 173, 175
- chart types 145, 156
- charts
  - adding 64, 119, 144, 145
  - controlling range of values for 181
  - creating 142, 143, 156, 183
  - displaying small sets of values and 153
  - displaying values in 146, 156, 172, 180
  - exporting 190
  - flipping axes values in 150
  - formatting 173, 175, 178
  - overlapping series in 147, 151
  - previewing 174
  - removing labels in 176
  - removing legends in 178
  - repositioning legends in 177
  - resizing 174–175
  - selecting data for 156, 184
  - setting animation effects for 190
  - setting dimensions for 176
  - showing percentages in 147, 149, 152
  - showing relationships in 153
  - showing scientific or statistical data 154
  - showing side-by-side series in 148
  - stacking series in 146, 148, 151
  - tracking stock data and 155
  - viewing 144, 190
- class files 257
- classes 255, 256, 257
- ClassicModelQueryBuilder.jsp 254
- ClassicModels database 252
- classpaths 256
- closing values 155
  - See also* stock charts
- code 254, 255
- colon (:) character 273
- colors 19, 33
- column areas (cross tabs) 131, 132
- column handles 20
- column header rows 38
- column headers
  - See also* column names
  - adding borders to 40
  - adding to tables 38
  - aligning text in 22
  - changing font properties for 21
  - creating 13, 48
  - editing 19
  - formatting 20
  - merging 37, 40
  - selecting 20
  - summary tables and 120
- column headings 270
- column headings (cross tabs) 131
- column names 65
  - See also* field names
- Column Properties dialog box 42
- Column Width command 42
- ColumnHeaderOperations action set 262
- ColumnOperations action set 262
- columns
  - See also* computed columns; fields
  - adjusting widths automatically 43
  - aggregating values in 68
  - aligning text in 22
  - building cross tabs and 132
  - building summary tables and 112, 120, 124
  - changing grouping order and 62
  - comparing values between 32, 85
  - counting number of values in 69
  - creating charts and 156
  - creating report sections and 65
  - creating reports and 13
  - filtering on 80, 84, 86
  - formatting data in 23, 25, 27, 29
  - formatting with conditions 29, 31, 34
  - grouping data and 60, 61, 62
  - inserting data fields and 41
  - merging 36, 37
  - removing report sections and 66
  - reordering 35, 119
  - resizing 41, 42, 44
  - selecting 20

- sorting on 57
  - spanning text across multiple 39, 40
  - viewing format type 23
- ColumnWidth action 262
- combo boxes 96
- comma-separated values files 238
- comments 235
- comparison operators 31, 82
- comparisons
  - conditional formats and 31, 32
  - date-and-time values 84
  - filter conditions and 80, 81
  - grouping data and 58
  - plotting data and 146
  - string patterns 84
  - values in multiple columns 85
- compilers 256
- compiling servlet classes 256
- computed columns
  - adding data fields for 50
  - calculating values for 192, 220
  - creating 46, 47
  - naming 48
  - summary tables and 121, 125
- concatenating string values 51, 221
- concatenation operator 221
- conditional expressions 52, 203
- conditional formats 29–34, 222
- Conditional Formatting command 29
- Conditional Formatting dialog box
  - comparing values and 31, 32
  - formatting data and 29, 30
- ConditionalFormat action 263
- conditions
  - calculating values and 52, 203
  - filtering data and. *See* filter conditions
  - formatting data and 30, 31, 32, 34
  - testing 221
- confidentiality statements 18
- configuration parameters. *See* parameters
- configurations
  - accessing data and 238, 239
  - changing servlets and 272
  - changing user actions and 261
  - customizing BIRT Studio and 260, 264
  - customizing data sources and 250, 252, 258
  - registering servlets and 257
  - running BIRT Studio and 268
  - setting default template category and 244
- connecting to
  - data sources 255
- context menus 20, 117, 260, 261
- context roots 229
- copyright statements 18, 237
- COUNT function 69, 218
- Count Value function 69
- COUNTDISTINCT function 218
- Create Template Report Item command 236
- CreateSection action 263
- creating
  - aggregate data 69
  - charts 142, 143, 156, 183
  - column headers 13, 38, 48
  - computed columns 46, 47
  - cross tabs 131
  - custom data sources 248, 248–250, 258
  - custom query builders 254–257
  - data filters 79
  - data groups 58, 60, 61, 62
  - expressions 51–53
  - filter conditions 79, 80, 86, 90
  - filter parameters 94, 97, 109
  - filters 79, 86, 92
  - formatting rules 29, 30, 33
  - libraries 240
  - list of values 97, 99, 105
  - multi-line data fields 36
  - ODA drivers 254
  - report items 236
  - report sections 63, 65
  - report templates 232, 235–238
  - reports 4, 5, 72, 248
  - resource folders 241
  - servlets 255
  - styles 233, 240
  - summary tables 112, 117
  - template categories 243, 244
  - themes 233, 239–242
  - Tooltips 98, 99, 172
- criteria. *See* conditions
- cross tab templates 133
- cross tabs
  - analyzing data in 140

- cross tabs (*continued*)
  - creating 131
  - designing 136
  - displaying 130, 139
  - formatting data in 139
  - previewing 139
  - retrieving data for 8
  - saving 139
  - selecting data for 131, 132
  - viewing aggregate values in 137, 138
  - viewing data in 131, 135
- Crosstab Builder 133, 138
- CSV files 238
- Currency format 24
- current date and time 210, 215
- customizing
  - BIRT Studio 4, 260, 264
  - data formats 24, 26, 28
  - data sources 248, 248–250, 258
  - query builder 254–257
  - reports 13, 18
  - styles 233, 240, 241

## D

- dashboards 153
- data
  - See also* values
  - accessing 238, 248
  - aggregating 68–70, 72
  - analyzing 140, 142
  - changing 29, 94
  - controlling access to 239
  - displaying 15, 46
  - filtering 8, 78, 94
  - formatting 20, 22, 25, 27, 29
  - grouping 58–63, 64, 112
  - hiding 72, 73, 190
  - including in templates 232, 238, 239
  - plotting 142, 143, 144, 156
  - previewing 4
  - redisplaying hidden 74
  - restoring presorted state 58
  - retrieving 8, 112, 255
  - selecting 20, 21
  - sorting 56, 57, 59
  - summarizing 112, 130
- Data action 263
- Data Analyzer 130, 140
- data cubes
  - building cross tabs and 131
  - structure of 131
- data drivers. *See* drivers
- data elements 232
- data fields
  - See also* columns
  - adding 12, 19, 41, 49
  - calculating values in 46, 48
  - changing headers for 19
  - creating multi-line 36
  - creating report sections and 64, 65
  - displaying 50
  - naming. *See* field names
  - referencing 47, 53
  - returning first value in 83
  - returning last value in 83
  - testing for null values in 205
- Data Fields action 262
- Data Fields command 49
- Data Fields dialog box 49
- data filters
  - applying 80, 283
  - building cross tabs and 135
  - comparing date-and-time values and 84
  - comparing string patterns and 84
  - comparing values in multiple columns and 85
  - creating 79, 86, 92
  - displaying top or bottom values and 82
  - enabling case sensitivity for 269
  - excluding null or blank values and 82
  - excluding sets of values and 82, 89
  - prompting for 94, 95
  - resetting 283
  - selecting multiple values for 79, 81, 95
  - setting evaluation order for 90
  - specifying conditions for. *See* filter conditions
  - specifying parameters as 94, 97, 104
  - testing 80
- data groups. *See* groups
- data objects
  - selecting 134
- data points 154

- data rows
  - adding 39
  - conditionally formatting data and 33
  - counting 218
  - displaying 13, 268, 269
  - grouping data and 58
  - merging columns and 37, 39
  - removing duplicate values from 59
  - removing report sections and 66
  - retrieving 78, 87, 89, 104, 270
  - returning first value in 218
  - returning last value in 219
  - returning most frequent value in 219
  - returning null values in 82
  - returning unique values in 60, 65, 218
  - selecting 21, 78
- data sections. *See* sections
- data series
  - adding 156
  - changing range of 181
  - displaying in multiple dials 153
  - displaying percentage of whole 147, 149, 152
  - displaying relationship between 154
  - displaying relationship to whole 153
  - displaying values 146, 156, 172, 180
  - omitting 180
  - overlapping multiple 147, 151
  - rotating 179
  - scaling 181
  - showing distinct values for 153
  - showing high and low values 155
  - showing open and close values 155
  - showing side-by-side comparisons of 148
  - stacking multiple 146, 148, 151
  - staggering 179
  - tracking single values and 153
- Data Set dialog box 50
- data set extensions 251, 255, 258
- data sets
  - building templates and 238, 239
  - creating list of values and 100, 106
  - displaying all fields in 50
  - displaying charts and 146
  - displaying values in 46, 49
  - filtering data and 78, 92
  - retrieving from text files 238
  - selecting 8
  - sorting data and 56
  - viewing 112
- Data Source dialog box 251, 258
- data source extensions 251, 255, 258
- data sources
  - accessing sample ODA 252
  - configuring 258
  - connecting to 255
  - creating reports and 248
  - creating templates and 238
  - customizing 248, 248–250, 258
  - naming 250, 258
  - prompting for input and 97, 104
  - restoring formats in 34
  - retrieving data from 8, 112, 255
  - selecting 8, 238
- data sparsity 132
- Data Tools Platform components 254
- data types 23, 69
- database drivers. *See* drivers
- databases 252
- datasetExtensionId element 251, 258
- datasourceExtensionId element 251, 258
- data-sources element 238
- date values
  - adding days to 52, 192
  - adding months to 194
  - adding quarters to 194
  - adding time values to 193, 195
  - adding weeks to 195
  - adding years to 195
  - aggregating 69
  - building data cubes and 131
  - calculating days between 51, 198
  - calculating months between 199
  - calculating quarters between 200
  - calculating time values between 198, 199, 200
  - calculating weeks between 201
  - calculating years between 202
  - comparing 31
  - entering in expressions 52
  - filtering 84
  - formatting 25, 26
  - grouping 61, 121
  - returning current 210, 215

- date values (*continued*)
  - returning month for 208
  - returning quarter in 210
  - returning weekdays for 197, 217
  - returning weeks for 217
  - returning year for 218
  - selecting 84
- Date/Time Column Format dialog box 25
- date-and-time formats 25, 26, 34, 121
- day formats 217
- DAY function 197
- days
  - adding to date values 52, 192
  - calculating number of 51, 198
  - returning number in month 197
  - returning specific 217
- default banner 276
- default locale 268
- default sort order 56
- default template category 243
- default themes 240
- Default Value property 99
- default values
  - overriding 95
  - prompting for input and 98, 100, 106
- DEFAULT\_DATA\_CACHE\_ROW\_COUNT
  - parameter 268
- DEFAULT\_LOCALE parameter 268
- DEFAULT\_PAGE\_BREAK\_INTERVAL
  - parameter 269
- DEFAULT\_REPORT\_TEMPLATE\_
  - CATEGORY\_NAME parameter 244, 269
- DEFAULT\_TIMEZONE parameter 269
- DEFAULT\_WORKGROUP\_
  - FUNCTIONALITY\_ROLE parameter 269
- delete file status page 274, 277
- Delete Group button 63
- Delete Section button 66
- DeleteColumn action 262
- DeleteColumn action set 262
- deletefilestatus.jsp 275
- DeleteRow action 262
- DeleteSection action 263
- deleting
  - blank characters 47, 215, 216
  - chart titles 176
  - data groups 63
  - duplicate values 59
  - files 277, 278
  - filter conditions 91
  - Flash charts 189
  - folders 278
  - legends 178
  - report items 18
  - report sections 66
- deploying
  - query builder servlet 257
- deploying Actuate Java Components 229
- descending sort order 57
- description element 251, 258
- Description property 236
- design files 14
- design tools 260
- designing
  - cross tabs 136
  - report templates 232, 233, 235
  - reports 18, 232, 260
  - summary tables 112
- designs
  - See also* page layouts
  - adding sections to 63, 64, 65
  - aggregating data and 68, 112
  - applying themes to 20
  - changing 13, 34, 269
  - displaying charts and 143, 144, 145, 173
  - displaying data rows in 13
  - displaying detail tables and 128
  - displaying summary tables and 112, 120, 124, 128
  - editing 18
  - filtering data and 78, 94
  - grouping data in 59, 64
  - hiding detail data in 72, 73
  - including calculations in 46
  - including report items in 18
  - opening 273
  - redisplaying detail data in 74
  - removing groups from 63
  - removing sections from 66
  - saving as templates 235
  - selecting report elements for 234
  - selecting report items in 18, 20
  - selecting templates for 6
  - setting locations for 15



- viewing changes to 75
  - viewing detail data in 112
  - detail areas (cross tabs) 131
  - detail data 72
  - detail pages 274, 277
  - detail rows 63, 72
  - detail tables 112, 128
  - detail.jsp 275
  - developing
    - custom data sources 252
    - custom query builders 254–257
    - ODA data drivers 254
  - dials 153
    - See also* meter charts
  - DIFF\_DAY function 51, 198
  - DIFF\_HOUR function 198
  - DIFF\_MINUTE function 199
  - DIFF\_MONTH function 199
  - DIFF\_QUARTER function 200
  - DIFF\_SECOND function 200
  - DIFF\_WEEK function 201
  - DIFF\_YEAR function 202
  - dimension columns 112, 120
    - See also* summary tables
  - Dimension property 176
  - dimensions 131, 132
  - directories 238
  - directory paths
    - image files 238
    - report documents 280
    - report templates 236
    - script libraries 268
    - shared resources 268
  - Display Name property 236
  - display names 234, 251, 258
  - Display Type property 98, 99, 105
  - display types 96
  - displaying
    - aggregate data 68, 69, 112, 137
    - banners 276
    - calculated data values 46, 49, 192
    - charts 144, 190
    - cross tabs 130, 139
    - data 4, 15, 46
    - data fields 50
    - data rows 13, 268, 269
    - data sets 112
    - date-and-time values 25, 26, 61
    - detail tables 128
    - distinct values 69, 153, 218
    - error messages 279
    - filter conditions 87
    - folders 283
    - legends 177
    - libraries 242
    - numeric values 22, 23
    - prompts 101, 103, 108
    - report elements 234, 240
    - report files 282
    - report items 18
    - report parameters 269, 284
    - report templates 7, 234, 243, 244
    - reports 43, 74, 232, 281
    - string values 27
    - summary tables 128
    - summary values 144
    - themes 20, 233
    - unique values 60, 65, 218
  - displayName element 251, 258
  - distinct values 69, 153, 218
    - See also* unique values
  - do directive 274
  - document files 278, 280, 283
  - documentation ix, 4
  - double quotation mark (") character 52, 206, 214
  - doughnut charts
    - See also* Flash charts
  - drag-and-drop operations 35
  - drivers 254
  - drop pages 274, 278
  - DTP project components 254
  - duplicate values 59, 69, 153
  - dynamic filter parameters 94, 96, 104
  - dynamic filters 78
  - dynamic lists 97, 105
  - dynamic value parameters 269
- ## E
- Eclipse project components 254
  - Edit Parameter dialog box 98, 99, 105
  - editable fields 19
  - editable report elements 234, 236

- editable report items 18, 19, 20
- EditCalculation action 262
- editing. *See* changing
- EditText action 262
- EditText action set 262
- empty cells 132
- empty values 82, 95
  - See also* null values
- ENABLE\_CASE\_SENSITIVE\_STRING\_FILTER parameter 269
- enabled element 261
- enabledInEnterpriseMode element 250, 251, 258
- enabledInWorkgroupMode element 251, 258
- EnableNewAggregationStyle parameter 267
- Encyclopedia volumes 275
  - See also* repositories
- entryPoint element 251, 258
- Equal to operator 222
- erni\_config.xml 258, 260
- error detail page 277
- error messages 279, 284
- error page 274, 279
- error.jsp 275
- exclamation point (!) character 28
- executable files 283
- executableName parameter 280
- execute report page 274, 280
- EXECUTE\_REPORT\_WAIT\_TIME parameter 280
- executing reports 15, 280
- exporting
  - Flash charts 190
  - reports 15, 42
- expressions
  - adding data fields and 49
  - adding functions to 48
  - adding reserved characters to 52
  - aggregating data and 70
  - calculating data values and 46, 48, 50, 192, 220
  - creating 51–53
  - filtering data and. *See* filter conditions
  - formatting data and 30, 31, 32, 34, 222
  - negating Boolean 209
  - validating 48
- extension IDs 251, 255, 258

## F

- features 18, 260
- field names 47, 53
  - See also* column headers; column names
- fields
  - See also* columns
  - adding 12, 19, 41, 49
  - calculating values in 46, 48
  - changing headers for 19
  - creating multi-line 36
  - creating report sections and 64, 65
  - displaying 50
  - referencing 47, 53
  - returning first value in 83
  - returning last value in 83
  - testing for null values in 205
- file and folder index page 282
- file and folder list page 282
- file manager 273, 274
- file manager URI reference 275
- file names 14
- file or folder detail page 278
- file or folder drop page 278
- file paths. *See* paths
- file system repository 228
  - See also* repositories
- filedetail.jsp 275
- filefolderlist.jsp 276
- FileOperations action set 263
- files
  - See also* specific type
  - accessing 282
  - browsing 277
  - changing location of 15
  - deleting 277, 278
  - displaying 282
  - filtering 283
  - getting information about 278
  - managing binary 272
- Filter action 262
- Filter button 80, 92
- filter conditions
  - adding multiple values to 79, 81, 95
  - adding parameters to 94
  - changing 91
  - creating 79, 80

- defining multiple 86, 87, 89, 108
- deleting 91
- grouping 90
- validating 88
- viewing 87
- Filter dialog box
  - data filters and 79, 80
  - filter parameters and 97, 105, 109
  - optional values and 95
- filter expressions 222
- Filter Pane 92
- filter parameter (URIs) 283
- filter parameters
  - creating 94, 97, 104
  - defining as optional 95
  - defining list of values and 99
  - defining multiple 108, 109
  - prompting for values and 94
  - setting default values for 100
  - setting display types for 96, 104
- Filter Text property 80, 88
- filtering
  - data 78, 94
  - empty or blank values 82
  - items in lists 32
  - range of values 82
  - report files 283
- filtering data 8
- filtering options 78
- filters
  - applying 80, 283
  - building cross tabs and 135
  - comparing date-and-time values and 84
  - comparing string patterns and 84
  - comparing values in multiple columns and 85
  - creating 79, 86, 92
  - displaying top or bottom values and 82
  - enabling case sensitivity for 269
  - excluding null or blank values and 82
  - excluding sets of values and 82, 89
  - prompting for 94, 95
  - resetting 283
  - selecting multiple values for 79, 81, 95
  - setting evaluation order for 90
  - specifying conditions for. *See* filter conditions
  - specifying parameters as 94, 97, 104
  - testing 80
- FIND function 202
- FIRST function 218
- Fixed format 24
- Fixed Width setting 43, 174
- Flash charts 183–190
- folder detail page 278
- folder drop page 278
- folder index page 282
- folder list page 282
- folder names 274
- folder parameter 283
- folders
  - accessing 282
  - browsing contents 277
  - copying ODA driver files to 254
  - creating resource 241
  - deleting 278
  - deploying class files to 257
  - displaying 283
  - getting information about 278
  - publishing resource files to 245
  - publishing templates to 243
  - saving reports to 14
  - selecting images and 236
  - viewing library files in 242
  - viewing objects in 282
- Font action 263
- font attributes. *See* font properties
- Font button 21
- Font dialog box 21
- font properties 21, 126, 176
- fonts 19
- footers 6, 69, 70
- Format Chart page
  - formatting chart titles and 175
  - formatting charts and 173
  - repositioning legends and 177
  - resizing charts and 174, 175
  - rotating labels and 179
  - scaling y-axis values and 181
  - setting dimension options and 176
  - setting x-axis label intervals and 180
  - staggering x-axis labels and 179
- Format Data button 23, 25, 27
- Format page (Crosstab Builder) 138

- format patterns 24, 26, 28
- format symbols 24, 26, 28
- formats
  - adding data rows and 40
  - applying with conditions 29–34
  - changing page layouts and 42
  - customizing 24, 26, 28
  - defining styles and 233
  - displaying data and 23, 25, 27
  - displaying months and 209
  - displaying weekdays and 217
  - resizing charts and 174
  - reverting to default 34
  - selecting 23, 25, 27
- formatting
  - charts 173, 175, 178
  - cross tabs 139
  - data 20, 22, 25, 27, 29
  - report items 20, 233
  - summary tables 117, 121
  - text 27
- Formatting action set 263
- formatting options 23, 29, 173, 233
- formatting rules 29, 30, 33
- forward slash (/) character 273
- functionality roles 269
- functions 47, 48, 192, 254
  - See also* specific function

## G

- General Number format 23
- GeneralOperations action set 263
- generating images 237
- generating reports 281
- getfiledetails page. *See* file or folder detail page
- getfolderitems page. *See* file and folder index page
- grand totals 137, 138
- graphical user interfaces 4
  - See also* BIRT Studio
- graphs. *See* charts
- Greater Than operator 222
- Greater Than or Equal to operator 222
- Group Details dialog box 61
- group footers 69, 70

- group headers 70
- GroupBy action 263
- grouped dimension columns 120
- grouping
  - data 58–63, 64, 112
  - report templates 6
- Grouping action set 263
- grouping intervals 61
- grouping options 61
- grouping order 62
- groups
  - adding report sections and 63, 64
  - aggregating data and 68, 70, 112
  - building charts and 144
  - building cross tabs and 131
  - building summary tables and 112, 120, 125
  - changing order of 62
  - creating 58, 60, 61, 62
  - deleting 63
  - filtering on 83
  - hiding detail data in 72
  - redisplaying detail data in 74
  - starting on new page 74
- GUIs. *See* graphical user interfaces

## H

- header rows 38, 145
- headers. *See* column headers; group headers
- headings. *See* section headings
- Help action 264
- Help action set 264
- Help Text property 98, 99, 105
- Hide Detail command 72, 74
- Hide Table command 190
- HideColumn action 262
- HideDetail action 263
- hiding data 72, 73, 190
- home folders 14, 283
- HOME property 239
- horizontal bar charts 150
- hours
  - See also* time values
  - adding to date values 193
  - calculating number of 198
- HTTPServlet class 255

## I

- ID parameter 279
- IF function 52, 203
- image files 236, 238, 244
- images 234, 236, 237
- IN function 204
- In operator 82, 95, 222
- index pages 282
- inedit parameter 256
- Information Console 273
  - integrating query builder with 254
- information object files 12
- information objects
  - building summary tables and 117
  - building templates and 232
  - selecting 12
- input 94, 95, 96, 97
- Insert Flash Chart command 183
- Insert Row Above command 39
- Insert Row Below command 39
- InsertRow action 262
- installing Actuate Java Components 228
- Interactive Viewer 139
- Interval setting 180
- invokeSubmit parameter 280
- Is Blank operator 222
- Is False operator 222
- Is Not Blank operator 222
- Is Not Null operator 82, 222
- Is Null operator 82, 222
- Is True operator 222
- iServer 273
- ISNULL function 205

## J

- Jakarta Struts action mapping 274
- JAR files 256, 257
- Java Components application 228, 229, 238
- Java Server Pages (JSPs) 254
- Java servlets. *See* servlets
- javac compiler 256
- JavaScript functions 254
- job priority 281
- jobs 280, 284
- JSP engine 281
- JSPs 273

## L

- Label Font property 176
- labels
  - adding report sections and 65
  - aligning text in 22, 34
  - changing font properties for 21, 176
  - deleting chart 176
  - displaying aggregate data and 70
  - displaying by interval 180
  - editing 19, 173
  - rotating 179
  - selecting 20
  - staggering 179
- LAST function 219
- laying out data
  - See also* page layouts
- Layout Preference property 43, 174
- leading characters 47, 215, 216
- LEFT function 205
- legends 176, 177
- LEN function 206
- Less Than operator 222
- Less Than or Equal to operator 223
- libraries 233, 240, 241, 268
- Library command 240
- library files 244
- LIKE function 206
- Like operator 84, 223
- line chart subtypes 151, 152
- line charts 146, 150, 158
  - See also* charts
- links 276, 282
- list boxes 96
- list pages 274, 282
- listing reports 68
- lists
  - adding empty values to 95
  - creating 97, 99, 105
  - displaying data and 58, 78, 97
  - filtering values in 32
  - prompting for values and 96, 97
  - selecting values in 31, 82
  - setting sort order for 100, 107
  - testing for values in 204
- literal characters 85, 207
- literal values 31, 32, 51, 52

- locales
  - calculating data values and 52
  - filtering date-and-time values and 84
  - formatting data and 22, 23, 25, 26
  - selecting templates and 235, 245
  - setting default 268
- logging in to BIRT Studio 273
- logical operators 89
- login information 4
- LOWER function 207
- lowercase characters 27, 28, 207
- Lowercase format 27

## M

- Manage Thumbnail dialog box 237
- margins 44
- MATCH function 207
- matching string patterns 82, 84, 206, 207, 214
- mathematical operations 47, 51, 69, 218
- MAX function 69, 219
- MAX\_BRSTUDIO\_DESIGN\_SESSION
  - parameter 269
- MAX\_BRSTUDIO\_USER\_SESSION
  - parameter 269
- MAX\_DATA\_CACHE\_ROW\_COUNT
  - parameter 269
- MAX\_NUMBER\_OF\_VALUES\_FOR\_DYNAMIC\_PARAMETER parameter 269
- measure columns 112, 124
  - See also* summary tables
- measures 131
- MEDIAN function 219
- MEMORY\_DATA\_CACHE\_ROW\_COUNT
  - parameter 269
- menus 20, 117, 260, 261
- Merge action 262
- Merge Columns command 37
- Merge Right command 40
- Merge Up command 37
- MergeColumns action 262
- merging
  - column headers 37, 40
  - columns 36, 37
- messages 235
- meter charts 153, 165
  - See also* charts

- methods 255
  - See also* functions
- Microsoft Word documents 42
- MIN function 69, 219
- minutes
  - See also* time values
  - adding to date values 193
  - calculating number of 199
- MOD function 208
- MODE function 219
- modulus 208
- month formats 209
- MONTH function 208
- months
  - adding to date values 194
  - calculating number of 199
  - returning 208
- MORE\_VALUE\_ROW\_COUNT
  - parameter 270
- MoveToDetail action 262
- MoveToGroup action 262
- moving through reports 59
- multi-column groups 62
- multi-line column headers 38
- multi-line data fields 36
- multi-page reports 59, 72, 74

## N

- name element 250, 258
- name parameter 278, 279
- Name property 237
- names
  - predefining styles and 240
  - referencing field 47, 53
  - reserved characters in 53
  - servlets 257
- naming
  - aggregate data 70
  - computed columns 48
  - data sources 250, 258
  - JSPs 274
  - report design files 14
  - report elements 234, 236, 237
  - report templates 236
  - servlets 272
- naming restrictions. *See* case sensitivity

- navigating reports 59
- negating Boolean expressions 209
- New action 263
- New Computed Column button 47
- New Library wizard 240
- New Style command 240
- New Style wizard 240
- New Template wizard 236
- New Theme wizard 241
- No Value field 95
- non-editable report elements 234, 237
- non-editable report items 18
- non-null values 209
  - See also* values
- NoRepeat action 262
- Not Between operator 82, 223
- Not Equal To operator 82
- Not Equal to operator 223
- NOT function 209
- Not Like operator 82, 223
- Not operator 89
- NOTNULL function 209
- NOW function 210
- null values 82, 205
  - See also* empty values
- Number Column Format dialog box 23
- number formats 22, 23, 24, 34
- numeric values
  - aggregating 69, 126, 218
  - averaging 218
  - calculating square root of 214
  - displaying 22, 23
  - entering in expressions 52
  - filtering 82
  - returning absolute 192
  - returning modulo 208
  - rounding 197, 211, 212, 213

## O

- objectID parameter 278
- ODA data sources 252, 255, 258
- ODA drivers 254
- ODA framework 254
- ODA session IDs 255
- odaconfig element 258
- OdaOverview.htm 254

- OdaSample data source 252, 258
- online documentation ix, 4
- online help. *See* online documentation
- online reports 43
- onlyLatest parameter 283
- Open action 263
- open data access. *See* ODA data sources; ODA drivers
- opening
  - BIRT Studio 4
  - calendar 84
  - chart editor 145
  - Data Analyzer 140
  - report designs 273
  - report templates 6
- opening values 155
  - See also* stock charts
- Operator property 105
- operators
  - See also* specific type
  - conditional formats 31
  - filter conditions 80, 94, 105
  - listed 220
- optional parameters 94, 95, 98, 99
- optional values 95
- OR operator 221
- Or operator 89
- orientation (page layouts) 44
- Outline view 240
- output
  - changing page layouts and 42, 43
  - exporting charts and 190
  - resizing charts and 174
  - testing data filters and 80
- output formats 42, 43, 190
- output objects 280
- outputDocName parameter 280
- overlapping values 34
- overlay chart subtypes 147, 151

## P

- page-based output formats 42, 43
- Page Break command 74
- page breaks 74, 264
- page footers 6
- page layout types 43

- page layouts
  - See also* designs
  - adding page breaks to 74, 264
  - adding report items to 18
  - changing 42
  - defining styles for 233, 240
  - displaying chart labels and 178, 181
  - fitting data in 43
  - grouping data for 58–63, 64
  - merging data and 36
  - repositioning chart legends and 177
  - resizing charts and 174, 175
  - selecting templates and 18
  - selecting themes for 233
- page names 274
- page not found messages 282, 284
- page not found page 274, 284
- page orientation 44
- Page Setup options 43
- page size options 43, 44
- PageBreak action 264
- PageLayout action set 264
- PageLayoutInToolbar action 264
- pagenotfound.jsp 276
- paper size settings 44
- Parameter action 263
- Parameter button 101, 103, 108
- parameters
  - adding custom data sources and 255
  - customizing BIRT Studio and 268
  - displaying 269, 284
  - executing reports and 281
  - file manager URIs 275
  - filtering data with 94, 95, 97, 104
  - prompting for input and 94, 95, 96, 97
  - running BIRT Studio servlet and 272
  - running reports and 280
- Parameters dialog box 101, 103, 108
- parameters page 274, 284
- parameters.jsp 276
- paths
  - image files 238
  - report documents 280
  - report templates 236
  - script libraries 268
  - shared resources 268
- pattern matching 84, 206, 207, 214
- PDF documents 42, 174
- percent (%) character 85, 206
- Percent format 24
- percent stacked chart subtypes 147, 149, 152
- percentages
  - charts and 146, 147
  - computed columns and 221
  - numeric data and 51, 83
- performance 78, 95, 104, 112
- period (.) character 273
- pie charts 146, 153, 161
  - See also* charts
- Place Library in Resource Folder
  - command 242
- pop-up menus. *See* context menus
- Position property 177
- predefined style names 240
- predefined styles 240
- predefined templates 6, 232
- predefined themes 233
- Preview action set 264
- PreviewHTML action 264
- previewing
  - charts 174
  - cross tabs 139
  - data 4
  - reports 13, 42
- printing reports 15, 42
- priorities (jobs) 281
- priority parameter 281
- priorityValue parameter 281
- privileges 6, 14
- progressive parameter 281
- progressive viewing 281
- Prompt Text property 95, 98, 105
- prompting for input 94, 95, 96, 97
- prompts, testing 101, 107, 108
- properties
  - applying styles and 240, 241
  - changing font 21
  - creating templates and 234, 236
  - formatting aggregate data and 126
  - formatting charts and 176
  - preserving animation 190
  - reverting to default 34
  - setting filter parameter 96, 104
  - setting page layout 43



- publishing
  - image files 238
  - resources 244–245
  - templates 243–244

## Q

- QUARTER function 210
- quarters 194, 200, 210
- QUARTILE function 219
- queries 104, 255
- query builder 251, 254–257, 258
- question mark (?) character 52, 214

## R

- radio buttons 96
- range of values 196
- redirect parameter 279
- redirection 280
- Redo action 264
- referencing data fields 47, 53
- registering servlets 257
- relative paths 236, 280
- removing. *See* deleting
- renaming themes 240
- Reorder Columns command 35, 62
- Re-order Columns dialog box 36
- Reorder Columns dialog box 62
- ReorderColumns action 262
- repeated values. *See* duplicate values
- RepeatValues action 262
- report design files 14
- report design tools 260
- report designs
  - See also* page layouts
  - adding sections to 63, 64, 65
  - aggregating data and 68, 112
  - applying themes to 20
  - changing 13, 34, 269
  - displaying charts and 143, 144, 145, 173
  - displaying data rows in 13
  - displaying detail tables and 128
  - displaying summary tables and 112, 120, 124, 128
  - editing 18
  - filtering data and 78, 94
  - grouping data in 59, 64
  - hiding detail data in 72, 73
  - including calculations in 46
  - including report items in 18
  - opening 273
  - redisplaying detail data in 74
  - removing groups from 63
  - removing sections from 66
  - saving as templates 235
  - selecting report elements for 234
  - selecting report items in 18, 20
  - selecting templates for 6
  - setting locations for 15
  - viewing changes to 75
  - viewing detail data in 112
- report document files 278, 280, 283
- report elements
  - adding to libraries 240
  - adding to templates 232, 234, 236
  - applying styles to 240
  - displaying 234, 240
  - naming 234, 236, 237
  - selecting 234
- report executable files 283
- report files
  - accessing 282
  - browsing 277
  - changing location of 15
  - deleting 277, 278
  - displaying 282
  - filtering 283
  - getting information about 278
- report items
  - See also* report elements
  - adding 18
  - deleting 18
  - displaying 18
  - formatting 20, 233
  - including in templates 236
  - selecting 18, 20
- Report Items list 18, 234
- report parameters 94, 284
  - See also* parameters
- report section headings 64, 65, 66, 144
- report sections
  - adding charts to 144, 145
  - adding page breaks to 264
  - building summary tables and 125

- report sections (*continued*)
  - creating 63, 65
  - deleting 66
  - displaying summary data in 68
  - hiding detail data in 72, 73
  - organizing data and 63, 64
  - redisplaying detail data in 74
  - starting on new page 74
- report template categories 6, 243, 244, 269
- Report Template dialog box 7, 234, 244
- report template files 235, 243
- Report Template page. *See* Report Template dialog box
- report templates
  - accessing 6
  - accessing resources for 244
  - adding images to 237
  - adding report elements to 232, 234, 236
  - adding themes to 19, 233, 239
  - associating libraries with 241
  - changing styles in 233
  - creating 232, 235–238
  - designing reports and 6, 18, 232, 233
  - displaying 7, 234, 243, 244
  - grouping 6
  - localizing 235, 245
  - naming 236
  - overriding formats in 34
  - providing data with 232, 238, 239
  - providing descriptions for 234, 236
  - publishing 243–244
  - selecting 6, 7, 244
  - setting properties for 234, 236
  - viewing descriptions of 7
- report titles 19
- report viewers 15, 174
- reports
  - accessing 228
  - changing 18, 34, 94
  - controlling content of 94, 95, 112
  - creating 4, 5, 72, 248
  - customizing 13, 18
  - designing 18
  - designing. *See* report designs
  - developing custom data sources for 248–258
  - displaying 43, 74, 232, 281
  - entering descriptions for 14
  - generating 281
  - generating table of contents for 59
  - moving through 59
  - organizing data in 56, 58, 63
  - paginating 74
  - previewing 13, 42
  - running 15, 280
  - saving 14, 15, 280
  - selecting data sources for 8
  - standardizing 233
  - submitting requests for 280
  - viewing changes to 75
  - viewing charts in 143, 144, 174, 190
- repositories
  - accessing items in 228, 272, 282
  - getting information about 277
  - managing files and folders in 273
  - saving reports to 14, 15
- repositoryType parameter 272
- requests 255
- required parameters 98, 99
- reserved characters 52
- reserved parameters 281
- resetFilter parameter 283
- resizing
  - charts 174–175
  - columns 41, 42, 44
- resource folders 236, 241, 245
- resources 244–245, 268
- RIGHT function 211
- right-click menus. *See* context menus
- Rotate Labels setting 179
- ROUND function 211
- ROUNDDOWN function 212
- rounding 197, 211, 212, 213
- ROUNDUP function 213
- row areas (cross tabs) 131, 132
- row headings 131
- rows
  - adding 39
  - building cross tabs and 131, 132
  - conditionally formatting data and 33
  - counting 218
  - displaying 13, 268, 269
  - grouping data and 58
  - merging columns and 37, 39

- removing duplicate values from 59
- removing report sections and 66
- retrieving 78, 87, 89, 104, 270
- returning first value in 218
- returning last value in 219
- returning most frequent value in 219
- returning null values in 82
- returning unique values in 60, 65, 218
- selecting 21, 78
- showing quartile value for 219
- .rptdesign files 14
- .rpttemplate files 235
- rules 29, 30, 33
- running reports 15, 280

## S

- sample data 4
- sample ODA data source 252, 258
- sample query builder 254
- sample servlet 255, 257
- sample templates 6, 232
- sample themes 233
- SampleServlet.class servlet 257
- SampleServlet.java 255
- Save action 263
- Save Report Design dialog box 14, 15
- SaveAs action 263
- saving
  - cross tabs 139
- saving reports 14, 15, 280
- scaling options (charts) 181
- scatter charts 146, 154, 168
  - See also* charts
- Scientific format 24
- script libraries 268
- SEARCH function 214
- search results 270
- SEARCH\_ENABLE\_COLUMN\_HEADERS
  - parameter 270
- SEARCH\_USE\_QUOTE\_DELIMITER
  - parameter 270
- searching text strings 202, 214
- seconds
  - See also* time values
  - adding to date values 195
  - calculating number of 200
- Section Heading command 66
- section headings 64, 65, 66, 144
- SectionHeading action 264
- SectionOperations action set 264
- sections
  - adding charts to 144, 145
  - adding page breaks to 264
  - building summary tables and 125
  - creating 63, 65
  - deleting 66
  - displaying summary data in 68
  - hiding detail data in 72, 73
  - organizing data and 63, 64
  - redisplaying detail data in 74
  - starting on new page 74
- sectors 153
  - See also* pie charts
- selection parameter 256
- series
  - adding 156
  - changing range of 181
  - displaying in multiple dials 153
  - displaying percentage of whole 147, 149, 152
  - displaying relationship between 154
  - displaying relationship to whole 153
  - displaying values 146, 156, 172, 180
  - omitting 180
  - overlapping multiple 147, 151
  - rotating 179
  - scaling 181
  - showing distinct values for 153
  - showing high and low values 155
  - showing open and close values 155
  - showing side-by-side comparisons of 148
  - stacking multiple 146, 148, 151
  - staggering 179
  - tracking single values and 153
- servers 229
- serverURL parameter 275, 281
- servlet classes 255, 256, 257
- servlet element 257
- servlet names 257, 272
- servlet.jar 257
- servlet-mapping element 257
- servlets
  - creating query builder and 255–257

- servlets (*continued*)
  - running BIRT Studio and 272
- session IDs 255
- sessionId parameter 256
- sessions 269
- shared resources 268
- shortcut menus. *See* context menus
- Show All command 190
- Show Detail command 74
- Show Legend setting 178
- ShowColumns action 262
- ShowDetail action 264
- showDocument parameter 283
- showExecutables parameter 283
- showFolders parameter 283
- side-by-side chart subtypes 148
- single quotation mark (') character 52
- software 228
- Sort button 57
- sort order 56, 57, 100, 107
- SortAscending action 264
- SortDescending action 264
- Sorting action set 264
- sorting data 56, 57, 59
- sorting options 57
- source code 254, 255
- space characters 273
- special characters 84, 206, 214
- Split action 262
- SQRT function 214
- square root 214
- stacked chart subtypes 146, 148, 151
- Stagger Labels setting 179
- standard chart subtypes 153
- standard deviation 219
- standard formats 23, 25, 27
- starting
  - Data Analyzer 140
- starting BIRT Studio 4, 272
- startUpMessage parameter 282
- state parameter 256
- static filter parameters 94, 96, 97, 99
- static filters 78
- static lists 97, 105
- static text 19
- STDDEV function 219
- stick stock charts 155
- stock chart subtypes 155
- stock charts 155, 170
  - See also* charts
- String Column Format dialog box 27
- string formats 27, 28, 34
- string functions 69
- string patterns 82, 84, 206, 207, 214
- string values 275
- strings
  - calculating values and 51, 69
  - comparing 31
  - concatenating 51, 221
  - converting to lowercase 27, 28, 207
  - converting to uppercase 28, 216
  - counting characters in 206
  - displaying values 27
  - filtering 82, 84
  - finding substrings in 202, 214
  - matching characters in 206, 207
  - referencing data fields and 47, 53
  - removing blank characters in 47, 215, 216
  - returning length of 206
  - returning substrings in 205, 211
- Struts action mapping 274
- styles 18, 173, 233, 240
- subpage parameter 282
- substrings
  - extracting 205, 211
  - finding location of 202, 214
- subtotals 137, 138
- SUM function 69, 219
- summarizing data 112, 130
- summary data 68, 120, 137
  - See also* aggregate data
- summary reports 72
- summary tables
  - adding columns to 112, 120, 124
  - changing 117–120
  - converting to detail tables 128
  - creating 112, 117
  - designing 112
  - displaying 128
- superimposed chart subtypes 153
- SwitchSummaryMode action 263
- system administrators 4

## T

- table footers 69
- table handles 119
- table headers 145
- table of contents 59
- tables
  - adding data fields to 12, 19, 41, 49
  - adding rows to 38, 39
  - creating charts and 143, 144, 145, 174, 190
  - creating computed columns in 47
  - deleting data groups in 63
  - designing summary 112
  - displaying aggregate data and 112
  - displaying aggregate data in 69
  - filtering data in 78, 79, 80, 83
  - formatting items in 20
  - grouping data for 60, 61, 62
  - merging columns in 36, 37
  - reordering columns in 35
  - resizing columns in 41, 42, 44
  - restoring presorted state 58
  - sorting data in 56, 57, 59
  - switching types 128
- template categories 6, 7, 243, 244, 269
- Template command 236
- template error pages 277
- template files 235, 243
- Template Image property 236
- template selection dialog box. *See* Report Template dialog box
- templates
  - accessing 6
  - accessing resources for 244
  - adding images to 237
  - adding report elements to 232, 234, 236
  - adding themes to 19, 233, 239
  - associating libraries with 241
  - changing styles in 233
  - creating 232, 235–238
  - designing reports and 6, 18, 232, 233
  - displaying 7, 234, 243, 244
  - grouping 6
  - localizing 235, 245
  - naming 236
  - overriding formats in 34
  - providing data with 232, 238, 239
  - providing descriptions for 234, 236
  - publishing 243–244
  - selecting 6, 7, 244
  - setting properties for 234, 236
  - viewing descriptions of 7
- TemplateTableOperations action set 264
- testing
  - custom data sources 250, 252
  - data filters 80
  - prompts 101, 107, 108
- text
  - aggregating values for 69
  - aligning 22, 34
  - changing 19, 27, 173
  - formatting 27
  - selecting templates and 235
  - setting font properties for 21, 176
  - spanning multiple columns 39, 40
  - spanning multiple lines 36
- text boxes 96
- text files 238, 270
- text formats. *See* string formats
- text strings. *See* strings
- themes
  - adding styles to 233, 240
  - adding to templates 19, 233, 239
  - applying 20
  - changing reports and 18
  - creating 233, 239–242
  - displaying 233
  - re-applying default formats and 34
  - selecting 20
- Themes list 20, 240
- 3-dimensional charts 176
- thumbnail images 236, 237
- Thumbnail property 237
- time continuums 150
- time values
  - adding to date values 193, 195
  - aggregating 69
  - building data cubes and 131
  - calculating number of 198, 199, 200
  - filtering 84
  - formatting 25, 26
  - grouping 61, 121
  - returning current 210, 215
- time zones 269, 275

- timezone parameter 275
- Title Font property 176
- titles 19, 173, 175
- TODAY function 215
- toolbars 20, 260, 261
- Tooltips 98, 99, 172, 236
- Top N operator 83, 223
- Top Percent operator 83, 223
- totals 46, 137, 138
  - See also* summary data
- trailing characters 47, 215, 216
- trends 142, 146
- TRIM function 47, 215
- TRIMLEFT function 216
- TRIMRIGHT function 216
- truncated strings 275
- Turn off Auto Summarize button 128
- Turn on Auto Summarize button 117, 128
- types. *See* data types

## U

- underscore (\_) character 85, 207
- Undo action 264
- UndoRedo action set 264
- UngroupBy action 264
- unique values 60, 65, 218
- UPPER function 216
- uppercase characters 28, 216
- Uppercase format 28
- URI parameters 272, 275
- URIs
  - BIRT Studio file manager 273, 274, 275
  - BIRT Studio servlet 272
  - JSP engine 281
- URLs
  - BIRT Studio 4, 273
  - BIRT Viewer 284
  - Java Components deployment 229
  - query builder 251, 257, 258
- UsageInBRS property 239
- Use Library dialog box 242
- user actions. *See* actions
- user interfaces 4
  - See also* BIRT Studio
- users
  - adding Tooltips for 98, 99, 236

- prompting 94, 95, 96, 97

## V

- Value (Y) Axis Title property 176
- value series (charts) 156, 172, 181
  - See also* y-axis values
- values
  - See also* data
  - applying custom formats to 24
  - averaging 218
  - calculating. *See* calculations
  - comparing. *See* comparisons
  - conditionally formatting 29, 33, 34
  - counting 69
  - defining Tooltips for 98, 99, 172
  - determining standard deviation
    - among 219
  - determining variance in 220
  - determining weighted average for 220
  - displaying distinct 69, 153, 218
  - displaying top or bottom 83
  - displaying unique 60, 65, 218
  - entering at runtime 78
  - filtering empty or blank 82
  - filtering on multiple 79, 81, 95
  - negating Boolean 209
  - prompting for 94, 95, 97
  - removing duplicate 59
  - returning largest 219
  - returning median 219
  - returning most frequent 219
  - returning smallest 219
  - selecting 21, 31, 82
  - setting default 98, 100, 106
  - showing over time 150
  - showing relationship between 154
  - showing relationship to whole 153
  - showing trends in 142, 146
  - testing for non-null 209
  - testing if null 205
  - testing range of 196
  - viewing at run time 96, 97, 105
- VARIANCE function 220
- version parameter 278
- viewers 15, 174

- viewing
  - aggregate data 68, 69, 112, 137
  - banners 276
  - calculated data values 46, 49, 192
  - charts 144, 190
  - cross tabs 130, 139
  - data 4, 15, 46
  - data fields 50
  - data rows 13, 268, 269
  - data sets 112
  - date-and-time values 25, 26, 61
  - detail tables 128
  - distinct values 69, 153, 218
  - error messages 279
  - filter conditions 87
  - folders 283
  - legends 177
  - libraries 242
  - numeric values 22, 23
  - prompts 101, 103, 108
  - report elements 234, 240
  - report files 282
  - report items 18
  - report parameters 269, 284
  - report templates 7, 234, 243, 244
  - reports 43, 74, 232, 281
  - string values 27
  - summary tables 128
  - summary values 144
  - themes 20, 233
  - unique values 60, 65, 218
- visible attribute 266
- visible element 261
- visual report elements 232
- volume parameter 275

**W**

- wait parameter 281
- WAR files 229
- web applications 257, 268
  - See also* applications
- web archive files 229
- web browsers
  - accessing BIRT servlets and 272
  - accessing reporting services and 228
  - displaying reports and 15

- issuing URIs and 275
- redirecting 280
- starting BIRT Studio and 4, 273
- web page banners 276
- web resources 268
- web.xml 257
- WEEK function 217
- weekday formats 217
- WEEKDAY function 217
- weekdays
  - adding to date values 52, 192
  - calculating number of 198
  - returning number in month 197
  - returning specific 217
- weeks
  - adding to date values 195
  - calculating number of 201
  - returning 217
- WEIGHTEDAVE function 220
- whitespace characters 273
- wildcard characters 84, 206, 214
- Word documents 42
- workingFolder parameter 277

## X

- X Axis Title property 175
- x-axis labels 173, 176, 179, 180
- x-axis titles 173, 175
- x-axis values 156, 178, 180
- x-axis values
  - See also* category series
- XML files 238

## Y

- Y Axis Title property 175
- y-axis labels 176
- y-axis titles 173, 175
- y-axis values 156, 172, 181
- y-axis values
  - See also* value series
- YEAR function 218
- years
  - adding to date values 195
  - calculating number of 202
  - returning 218

