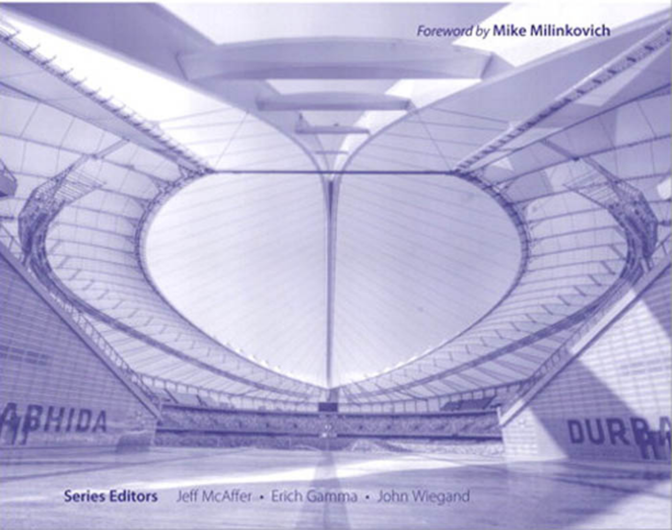
the eclipse series

Integrating and Extending BIRT

Third Edition

Jason Weathersby • Tom Bondur • Iana Chatalbasheva



Foreword by **Mike Milinkovich**

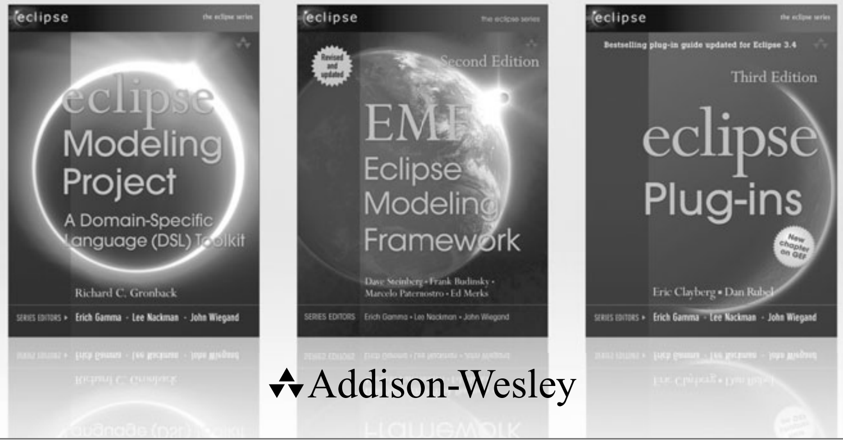
Series Editors Jeff McAffer • Erich Gamma • John Wiegand

Integrating and Extending BIRT

Third Edition

The Eclipse Series

Eric McAffer, Erich Gamma, John Wiegand, Series Editors


eclipse Modeling Project
A Domain-Specific Language (DSL) Toolkit
Richard C. Gronback
Second Edition
EMF Eclipse Modeling Framework
Dave Steinberg • Frank Budinsky • Marcelo Paternostro • Ed Merks
Bestselling plug-in guide updated for Eclipse 3.4
Third Edition
eclipse Plug-ins
Eric Clayberg • Dan Rubel
SERIES EDITORS ▸ Erich Gamma • Lee Nackman • John Wiegand

Addison-Wesley

Visit **informit.com/series/eclipse** for a complete list of available publications.

Eclipse is a universal, multilanguage software development environment—an open, extensible, integrated development environment (IDE)—that can be used for anything. Eclipse represents one of the most exciting initiatives to come from the world of application development, and it has the support of leading companies and organizations in the technology sector. Eclipse is gaining widespread acceptance in both commercial and academic arenas.

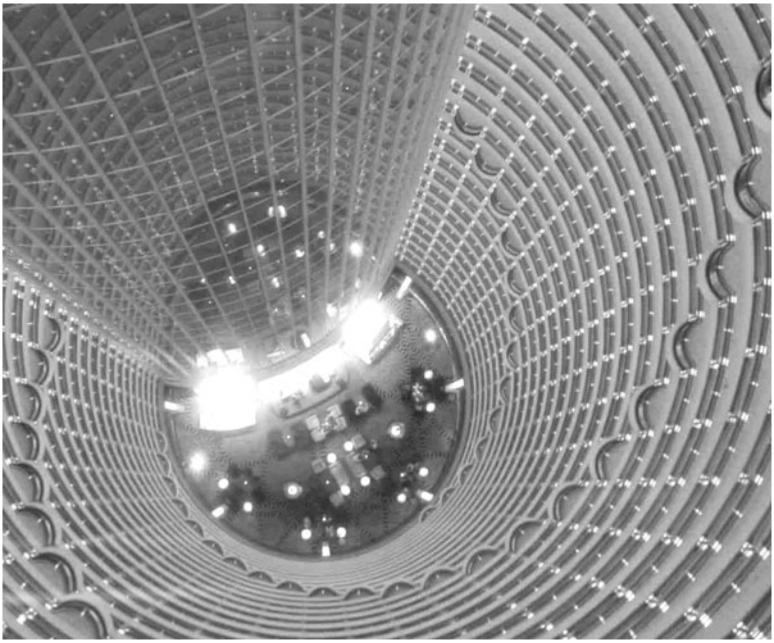
The Eclipse Series is the definitive collection of publications dedicated to the Eclipse platform. Books in this series bring you key technical information, critical insight, and the practical advice you need to build tools to support this revolutionary open-source platform.

PEARSON

Addison-Wesley Cisco Press EXAM/CRAM IBM Press QUE PRENTICE HALL SAMS | Safari Books Online

Integrating and Extending BIRT

Third Edition



Jason Weathersby • Tom Bondur • Iana Chatalbasheva

Addison-Wesley

Upper Saddle River, NJ • Boston • Indianapolis • San Francisco
New York • Toronto • Montreal • London • Munich • Paris • Madrid
Capetown • Sydney • Tokyo • Singapore • Mexico City

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and the publisher was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals.

The authors and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

The publisher offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales, which may include electronic versions and/or custom covers and content particular to your business, training goals, marketing focus, and branding interests. For more information, please contact:

U.S. Corporate and Government Sales
(800) 382-3419
corpsales@pearsontechgroup.com

For sales outside the United States please contact:

International Sales
international@pearsoned.com

Visit us on the Web: informit.com/aw

Library of Congress Control Number: 2011932838

Copyright© 2012 by Actuate Corporation

All rights reserved. Printed in the United States of America. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. To obtain permission to use material from this work, please submit a written request to Pearson Education, Inc., Permissions Department, One Lake Street, Upper Saddle River, New Jersey 07458, or you may fax your request to (201) 236-3290.

ISBN-13: 978-0-321-77282-4
ISBN-10: 0-321-77282-2

Text printed on recycled paper in the United States at Courier in Westford, Massachusetts.
First printing, August 2011

Foreword. xix

Preface xxi

About this book xxi
Who should read this book xxii
Contents of this book xxii
Typographical conventions xxvii
Syntax conventions xxvii

Acknowledgments xxix

Part 1 Installing and Deploying BIRT 1

Chapter 1 Introducing BIRT Report Designers 3

Understanding BIRT components 3
Understanding Eclipse BIRT packages 4
About types of BIRT builds 5

Chapter 2 Installing a BIRT Report Designer 7

Installing BIRT Report Designer Full Eclipse Install 7
Installing BIRT RCP Report Designer 8
Troubleshooting installation problems 9
  Avoiding cache conflicts after you install a BIRT report designer 9
  Specifying a Java Virtual Machine when starting BIRT report designer 10
Installing a language pack 10
Updating a BIRT Report Designer installation 11
Updating BIRT RCP Report Designer installation 12

Chapter 3 Installing Other BIRT Packages 15

Installing Chart Engine 15
Installing BIRT Data Tools Platform Integration 17
Installing BIRT Demo Database 17
Installing Report Engine 19
Installing BIRT Samples 21
Installing BIRT Source Code 21

Installing BIRT Web Tools Integration 22

Chapter 4 Deploying a BIRT Report to an Application Server 25

About application servers 25
About deploying to Tomcat 25
About deploying to other application servers 26
Placing the BIRT report viewer on an application server 26
Installing the BIRT report viewer as a web application 26
Testing the BIRT report viewer installation 28
Changing the BIRT report viewer context root 28
Changing the BIRT report viewer location 28
Understanding the BIRT report viewer context parameters 29
Verifying that Apache Tomcat is running BIRT report viewer 30
Placing fonts on the application server 31
Viewing a report using a browser 31
Using connection pooling on Tomcat 32
Setting up a report to use connection pooling 32
Using a jndi.properties file 32
Configuring a JNDI connection object on Tomcat 33

Part 2 Understanding the BIRT Framework 37

Chapter 5 Understanding the BIRT Architecture 39

Understanding the BIRT integration 39
About the BIRT applications 43
About BIRT Report Designer and BIRT RCP Report Designer 43
About the BIRT Viewer 44
About the BIRT engines and services 44
About the design engine 44
About the report engine 45
About the generation services 45
About the presentation services 45
About the chart engine 45
About the data engine and services 46
About data services 46
About the ODA framework 46
About the types of BIRT report items 46
About standard report items 46
About custom report items 47
About the chart report item 47
About the Report Object Model (ROM) 47
About the types of BIRT files 47
About report design files 48
About report document files 48
About report library files 48
About report template files 48

About custom Java applications 49
About a custom report designer 49
About a custom Java report generator 49
About extensions to BIRT 50

Chapter 6 Understanding the Report Object Model 51

About the ROM specification 51
ROM methods 52
ROM properties 52
ROM slots 52
ROM styles 52
About the ROM schema 53
About the rom.def file 53
Understanding ROM elements 58
About the primary ROM elements 58
About report item elements 59
About report items 59
Understanding report item element properties 59
About data report elements 60

Part 3 Scripting in a Report Design 61

Chapter 7 Using Scripting in a Report Design 63

Overview of BIRT scripting 63
Choosing between JavaScript and Java 63
Using both JavaScript and Java 64
Events overview 64
Engine task processes 64
BIRT Web Viewer 65
BIRT processing phases 65
BIRT event types 66
Parameter events 66
Report design events 67
Types of data source and data set events 68
ReportItem events 71
Event order sequence 71
Preparation phase operation 71
Generation phase operation 73
About data source and data set events 74
About data binding 75
About page break events 76
About chart event order 77
About table and list event order 77
Completion of the generation phase 80
Presentation phase operation 80
Event order summary 80

Chapter 8 Using JavaScript to Write an Event Handler 83
Using BIRT Report Designer to enter a JavaScript event handler 83
Creating and using a global variable 84
Creating and using page variables and scripts 85
Understanding execution phases and processes 86
Using the reportContext object 86
Using getOutputFormat 88
Using reportContext to retrieve the report design handle 89
Passing a variable between processes 90
Using getAppContext 91
Getting information from an HTTP request object 93
Using the this object 93
Using this object methods 93
Using the this object to set a report item property 94
Using the row object 96
Getting column information 97
Getting and altering the query string 98
Changing data source connection properties 99
Getting a parameter value 100
Determining script execution sequence 101
Providing the ReportDesign.initialize code 101
Providing code for the scripts you want to track 102
Providing the ReportDesign.afterFactory code 103
Tutorial 1: Writing an event handler in JavaScript 103
Task 1: Create the report design 103
Task 2: Create a counter in Table.onCreate 104
Task 3: Conditionally increment the counter 105
Task 4: Display the result 106
JavaScript event handler examples 107
JavaScript onPrepare example 107
JavaScript onCreate examples 108
JavaScript onRender examples 109
Cross-tab script examples 109
Calling external JavaScript functions 111
Calling Java from JavaScript 112
Understanding the Packages object 112
Understanding the importPackage method 112
Using a Java class 113
Placing Java classes 113
Issues with using Java in JavaScript code 114
Calling the method of a class in a plug-in 114

Chapter 9 Using Java to Write an Event Handler 117
Writing a Java event handler class 117
Locating the JAR files for a BIRT event handler 117
Extending an adapter class 118
Making the Java class visible to BIRT 121

Associating a Java event handler class with a report element 121
BIRT Java interface and class naming conventions 123
Writing a Java event handler 123
Using event handler adapter classes 124
Using event handler interfaces 124
About the Java event handlers for report items 125
Using Java event handlers for a data source element 125
Using Java event handlers for a data set element 126
Using Java event handlers for a scripted data source element 126
Using Java event handlers for a scripted data set element 127
Using Java event handlers for a report design 128
Understanding the BIRT interfaces 128
About the element design interfaces 129
About the methods for each report element 129
About the IReportElement interface 129
About the element instance interfaces 130
Using the IReportContext interface 131
Using the IColumnMetaData interface 133
Using the IDataSetInstance interface 133
Using the IDataSetRow interface 134
Using the IRowData interface 135
Java event handler example 135
Report level events 135
Report item events 137
Debugging a Java event handler 142

Chapter 10 Working with Chart Event Handlers 143
Chart events overview 143
Understanding the Chart script context 145
Using the Chart instance 146
Chart instance getter methods 146
Chart instance setter methods 147
Chart instance methods 148
Accessing the Chart instance in an event handler 148
Understanding the external context 149
Understanding when chart events fire 150
Prepare phase 150
Data binding phase 150
About the beforeDataSetFilled event 151
About the afterDataSetFilled event 152
Building phase 153
About the beforeGeneration event 153
About the afterGeneration event 154
Rendering phase 155
Rendering phase script events 156
Rendering blocks 158
Rendering series 158
Rendering data points and data point labels 159
Rendering axes 161

Rendering legend items 163
Writing a Java chart event handler 164
Setting up the chart event handler project 164
Java chart event handler examples 165
Writing a JavaScript chart event handler 167
Using the simplified charting API 169
Getting an instance of a chart item 170
Understanding the sub-interfaces of IChart 170

Chapter 11 Using Scripting to Access Data 173
Using a Scripted Data Source 173
Tutorial 2: Creating a scripted data source 175
Task 1: Create a new report design 175
Task 2: Create a scripted data source. 175
Task 3: Create a scripted data set 176
Task 4: Write the open( ) and close( ) methods of the data source 178
Task 5: Write the open( ) method of the data set 178
Task 6: Write the fetch( ) method of the data set 178
Task 7: Place the columns on the report layout 179
Writing the scripted data set in Java 180
Using a Java object to access a data source 181
Performing initialization in the data set open( ) method 181
Getting a new row of data in the data set fetch( ) method 182
Cleaning up in the data set close( ) method 182
Deciding where to place your Java class 182
Deploying your Java class 183
Using input and output parameters with a scripted data set 183
Creating a web services data source using a custom connection class 184

Chapter 12 Debugging Event Handlers 187
Checking the syntax of JavaScript expressions 188
Debugging JavaScript event handlers code 190
Debugging Java event handler code 194
Debugging report execution exceptions 194
Creating a debug configuration 196
Tutorial 3: Debugging a report that contains Java and JavaScript code 198
Task 1: Preparing the report for debugging 199
Task 2: Setting a JavaScript breakpoint 203
Task 3: Setting a Java breakpoint 204
Task 4: Create the debug configuration. 206
Task 5: Debugging the report 207

Part 4 Integrating BIRT into Applications 215

Chapter 13 Understanding the BIRT APIs 217
Package hierarchy diagrams 218
About the BIRT Report Engine API 219

Creating the BIRT ReportEngine instance 220
Using the BIRT Report Engine API 220
EngineConfig class 220
ReportEngine class 221
IReportRunnable interface 221
IReportDocument interface 221
IEngineTask interface 222
IGetParameterDefinitionTask interface 222
IDataExtractionTask interface 222
IRunTask interface 222
IRenderTask interface 222
IRunAndRenderTask interface 223
Report engine class hierarchy 223
Report engine interface hierarchy 225
About the Design Engine API 226
Using the BIRT Design Engine API 227
DesignConfig class 227
DesignEngine class 227
SessionHandle class 228
ModuleHandle class 228
ReportDesignHandle class 229
LibraryHandle class 229
DesignElementHandle class 230
Individual element handle classes 230
Design engine class hierarchy 230
DesignElementHandle hierarchy 232
ReportElementHandle hierarchy 233
ReportItemHandle hierarchy 235
ElementDetailHandle hierarchy 236
StructureHandle hierarchy 236
Design engine interface hierarchy 238
About the BIRT Chart Engine API 239
Using the BIRT Chart Engine API 239
Chart engine class hierarchy 240
chart.aggregate class and interface hierarchy 240
chart.datafeed class and interface hierarchy 241
chart.device class and interface hierarchy 241
chart.event class and interface hierarchy 242
chart.exception class hierarchy 244
chart.factory class and interface hierarchy 244
chart.log class and interface hierarchy 245
chart.model interface hierarchy 245
chart.model.attribute class and interface hierarchy 246
chart.model.component interface hierarchy 250
chart.model.data interface hierarchy 251
chart.model.layout interface hierarchy 253
chart.model.type interface hierarchy 253
chart.render class and interface hierarchy 255
chart.script class and interface hierarchy 255

chart.util class hierarchy 256

Chapter 14 Programming Using the BIRT Reporting APIs 257

Building a reporting application 258
About the development environment 259
About plug-ins in BIRT home 259
About libraries in BIRT home 259
About required JDBC drivers 261
Setting up the build path and accessing Javadoc 261
Modifying a report design using the API 262
About the deployment environment 262
Generating reports from an application 262
Setting up the report engine 263
Configuring the BIRT home 263
Configuring the report engine 264
Setting up a stand-alone or WAR file environment 265
Starting the platform 267
Creating the report engine 267
Using the logging environment to debug an application 269
How to use BIRT logging 270
Opening a source for report generation 270
Understanding an IReportRunnable object 270
Understanding an IReportDocument object 271
Accessing a report parameter 272
Preparing to generate the report 278
Setting the parameter values for running a report design 279
Adding to the report engine's class path 279
Providing an external object to a report design 280
Generating a binary report document 280
Preparing to render a formatted report 281
Setting up the rendering options 281
Rendering formatted output 286
Accessing the formatted report 288
Checking the status of a running report task 288
Cancelling a running report task 288
Programming the structure of a report design 289
About BIRT model API capabilities 290
Opening a report design for editing 291
Configuring the design engine to access a design handle 291
Using an IReportRunnable object to access a design handle 292
Using a report item in a report design 292
Accessing a report item by iterating through a slot 293
Accessing a report item by name 293
Examining a report item 294
Accessing the properties of a report item 294
Modifying a report item in a report design 296
Accessing and setting complex properties 297
Understanding property structure objects 298
Adding a report item to a report design 302

Accessing a data source and data set with the API .303
About data source classes .303
About data set classes .304
Using a data set programmatically .304
Saving a report design .306
Creating a report design .306

Chapter 15 Programming Using the BIRT Charting API .307

About the chart engine contents .307
About the environment for a charting application .308
Configuring the chart engine run-time environment .308
Verifying the environment for a charting application .309
About the charting API and the chart structure .310
About chart visual components .310
About chart data .310
Understanding static data .310
Understanding dynamic data .311
Using the charting API to create a new chart .312
Modifying chart properties .312
Understanding simple and complex properties .313
Setting plot properties .313
Setting legend properties .314
Setting axes properties .315
Using series .315
Adding a series to a chart .316
Creating a category series .316
Creating an orthogonal series .317
Setting series properties .317
Associating data with a series .318
Adding a series definition to a chart .319
Setting up the default aggregation for the chart .319
Changing the aggregation for secondary value series .319
Adding a chart event handler .320
Adding a Java chart event handler .320
Adding a JavaScript chart event handler .320
Using a chart item in a report design .321
Accessing an existing chart item .321
Creating a new chart item .322
Getting a design engine element factory object .323
Setting the chart type and subtype .323
Creating sample data .325
Getting an extended item handle object .325
Setting up the report item as a chart .325
Preparing a data set and data columns .326
Binding the chart to the data set .326
Set any other report item properties .327
Adding the new chart to the report design .327
Saving the report design after adding the chart .327

Putting it all together 327
Using the BIRT charting API in a Java Swing application 334
Understanding the chart programming examples 340
api.data examples 340
DataCharts example 340
GroupOnXSeries example 341
GroupOnYAxis example 341
api.data.autobinding example 341
api.format example 341
api.interactivity examples 342
api.pdf example 342
api.preference example 342
api.processor example 342
api.script examples 343
api.viewer examples 343
Chart3DViewer example 343
CurveFittingViewer example 343
DialChartViewer example 344
SwingChartViewerSelector example 344
SwingLiveChartViewer example 344
SWTChartViewerSelector example 344
builder example 345
radar.ui example 345
report.api examples 345
MeterChartExample example 345
SalesReport example 345
StockReport example 345
report.design examples 346
report.design.script examples 346
view example 346

Part 5 Working with the Extension Framework 347

Chapter 16 Building the BIRT Project 349
About building the BIRT project 349
Installing a working version of BIRT 350
Configuring Eclipse to compile BIRT and build the viewer JAR files 350
Downloading and extracting the correct version of the BIRT source code 353
Importing, building, and testing the BIRT project 354
Building new JAR files to display BIRT output 356
Building the viewservlets.jar file 356
Building the chart-viewer.jar file 358

Chapter 17 Extending BIRT 359
Overview of the extension framework 359
Understanding the structure of a BIRT plug-in 359
Understanding an extension point schema definition file 360

Understanding a plug-in manifest file .362
Understanding a plug-in run-time class .364
Working with the Eclipse PDE .366
Understanding plug-in project properties .367
Understanding the Eclipse PDE Workbench .368
Creating the structure of a plug-in extension .370
Creating the plug-in extension content .373
Building a plug-in extension .377
Generating an Ant build script .380
Testing a plug-in extension .380
Deploying the extension plug-in .381
Creating an update site project .382
Installing available software .384
Downloading the code for the extension examples .384

Chapter 18 Developing a Report Item Extension .387

Understanding a report item extension .387
Developing the sample report item extension .389
Downloading BIRT source code from the CVS repository .390
Creating a rotated label report item plug-in project .390
Defining the dependencies for the rotated label report item extension .393
Specifying the run-time package for the rotated label report item extension .394
Declaring the report item extension points .395
Creating the plug-in extension content .399
Understanding the rotated label report item extension .404
Understanding RotatedLabelItemFactoryImpl .405
Understanding RotatedLabelUI .405
Understanding RotatedLabelPresentationImpl .405
Understanding GraphicsUtil .406
Understanding RotatedLabelCategoryProviderFactory .409
Understanding RotatedLabelGeneralPage .410
Deploying and testing the rotated label report item plug-in .412
Deploying a report item extension .412
Launching the rotated label report item plug-in .412
Developing an advanced report item .416
Defining the report item model .417
Defining the report item UI design .420
Defining the report item presentation .424
Improving the report item UI design .429
Using the Image UI provider .429
Using the Figure UI Provider .435
Creating a report item builder .439
Creating a context menu .443
Creating a property editor .445
Binding the report item to data .456
Changes in the model definition .456
Changes in the report item presentation .457
Changing the report item UI to support expressions .458

Adding data binding to the builder 458
Adding data binding to the property page 460

Chapter 19 Developing a Report Rendering Extension 465

Understanding a report rendering extension 465
Developing a CSV report rendering extension 466
Creating a CSV report rendering plug-in project 466
Defining the dependencies for the CSV report rendering extension 469
Declaring the emitters extension point 470
Understanding the sample CSV report rendering extension 473
Understanding the CSV report rendering extension package 473
Understanding CSVReportEmitter 474
Understanding CSVTags 479
Understanding CSVWriter 479
Understanding CSVRenderOption 479
Testing the CSV report rendering plug-in 480
About ExecuteCSVReport class 486
About the report design XML code 488
Developing an XML report rendering extension 494
Creating an XML report rendering plug-in project 494
Defining the dependencies for the XML report rendering extension 496
Declaring the emitters extension point 497
Understanding the sample XML report rendering extension 497
Understanding the XML report rendering extension package 498
Understanding XMLReportEmitter 498
Understanding XMLTags 503
Understanding XMLFileWriter 504
Understanding XMLRenderOption 504
Understanding LoadExportSchema 504
Testing the XML report rendering plug-in 506

Chapter 20 Developing an ODA Extension 509

Understanding an ODA extension 510
Developing the CSV ODA driver extensions 511
About the CSV ODA plug-ins 512
Downloading BIRT source code from the CVS repository 512
Implementing the CSV ODA driver plug-in 513
Understanding the ODA data source extension points 517
Understanding dataSource extension point properties 517
Understanding ConnectionProfile properties 520
Understanding the dependencies for the CSV ODA driver extension 521
Understanding the sample CSV ODA driver extension 522
Understanding Driver 523
Understanding Connection 523
Understanding Query 524
Understanding ResultSet 527
Understanding ResultSetMetaData 528
Understanding DataSetMetaData 528
Understanding Messages 529

Understanding CommonConstants .529
Developing the CSV ODA user interface extension .530
Creating the CSV ODA user interface plug-in project .531
Understanding the ODA data source user interface extension points .534
Understanding the ConnectionProfile extension point .536
Understanding the propertyPages extension point .536
Understanding the dataSource extension point .537
Understanding the sample CSV ODA user interface extension .537
Implementing the ODA data source and data set wizards .538
Understanding the org.eclipse.birt.report.data.oda.csv.ui.impl package .539
Understanding the org.eclipse.birt.report.data.oda.csv.ui.wizards package .539
Understanding Constants .540
Understanding CSVFilePropertyPage .540
Understanding CSVFileSelectionPageHelper .541
Understanding CSVFileSelectionWizardPage .543
Understanding FileSelectionWizardPage .543
Testing the CSV ODA user interface plug-in .548
Developing a Hibernate ODA extension .554
Creating the Hibernate ODA driver plug-in project .556
Understanding the sample Hibernate ODA driver extension .564
Understanding HibernateDriver .565
Understanding Connection .565
Understanding DataSetMetaData .568
Understanding Statement .568
Understanding ResultSet .572
Understanding HibernateUtil .573
Building the Hibernate ODA driver plug-in .577
Developing the Hibernate ODA user interface extension .577
Understanding the sample Hibernate ODA user interface extension .584
Understanding HibernatePageHelper .585
Understanding HibernateDataSourceWizard .588
Understanding HibernatePropertyPage .588
Understanding HibernateHqlSelectionPage .588
Building the Hibernate ODA user interface plug-in .594
Testing the Hibernate ODA user interface plug-in .594

Chapter 21 Developing a Data Extraction Extension .601

Understanding a data extraction extension .601
Developing an XML data extraction extension .602
Creating an XML data extraction plug-in project .603
Defining the dependencies for the XML data extraction extension .605
Declaring the data extraction extension point .607
Understanding the XML data extraction extension .609
Implementing the data extraction interfaces .609
Understanding the XML document format .610
Understanding XMLDataExtractionImpl methods .614
Testing the XML data extraction plug-in .620
About the report design XML code .626

Chapter 22 Developing a Fragment 635
Understanding a fragment 635
Developing the sample fragment 636
Creating a fragment project 637
Understanding the sample fragment 640
Building, deploying, and testing a fragment 641

Chapter 23 Developing a Charting Extension 647
About BIRT charting extension points 648
Setting up the build environment 650
Extending the chart model 652
Creating an EMF model 654
Completing the new SeriesImpl 658
Implementing the extension points 661
Design-time plug-in extensions 661
Chart UI-types extension point 661
Series composite extension point 673
Run-time Plug-in extensions 684
Chart model types extension point 684
Data processor extension point 685
Model Renderer Extension Point 685
Debugging this example 705

Glossary 707

Index 779

Foreword

It is a common misconception that Eclipse projects are focused on simply providing great tools for developers. Actually, the expectations are far greater. Each Eclipse project is expected to provide both frameworks and extensible, exemplary tools. As anyone who has ever tried to write software with reuse and extensibility in mind knows, that is far more difficult than simply writing a tool.

"Exemplary" is one of those handy English words with two meanings. Both are intended in its use above. Eclipse projects are expected to provide tools that are exemplary in the sense that they provide an example of the use of the underlying frameworks. Eclipse tools are also intended to be exemplary in the sense that they are good and provide immediate utility to the developers who use them.

Since its inception, the BIRT project has worked hard to create both reusable frameworks and extensible tools. This book focuses primarily on how to extend BIRT and how to use BIRT in your own applications and products. As such, it illustrates BIRT's increasing maturity and value as an embedded reporting solution.

As Executive Director of the Eclipse Foundation, I'm pleased with the tremendous progress the BIRT team has made since the project's inception in September of 2004, and I'm equally pleased with the vibrant community that has already grown up around it. As you work with BIRT and the capabilities that are described in this book, I'd encourage you to communicate your successes back to the community, and perhaps consider contributing any interesting extensions you develop. The BIRT web site can be found here:

```
http://www.eclipse.org/birt
```

It includes pointers to the BIRT newsgroup, where you can communicate and share your results with other BIRT developers, and pointers to the Eclipse installation of Bugzilla, where you can contribute your extensions. If you like BIRT—and I am sure this book will help you learn to love it—please participate and contribute. After all, it is the strength of its community that is the true measure of any open source project's success.

Mike Milinkovich
Executive Director, Eclipse Foundation

This page intentionally left blank

About this book

The second of a two-book series on business intelligence and reporting technology, *Integrating and Extending BIRT*, introduces programmers to BIRT architecture and the reporting framework. Its companion book, *BIRT: A Field Guide*, shows report developers how to create reports using the graphical tools of BIRT Report Designer. Built on the open-source Eclipse platform, BIRT is a powerful reporting system that provides an end-to-end solution, from creating and deploying reports to integrating report capabilities in enterprise applications.

BIRT technology makes it possible for a programmer to build a customized report using scripting and BIRT APIs. This book informs report developers about how to write scripts that:

- Customize the report-generation process
- Incorporate complex business logic in reports

This book also informs application developers about how to:

- Debug and deploy reports
- Integrate reporting capabilities into other applications
- Extend BIRT functionality

A programmer can extend the BIRT framework by creating a new plug-in using the Eclipse Plug-in Development Environment (PDE). This book provides extensive examples that show how to build plug-ins to extend the features of the BIRT framework. The source code for these examples is available for download at http://www.actuate.com/birt/contributions.

The topics discussed in this book include:

- Installing and deploying BIRT
- Deploying a BIRT report to an application server
- Understanding BIRT architecture

- Scripting in a BIRT report design
- Integrating BIRT functionality into applications
- Working with the BIRT extension framework

This revised BIRT 2.6 edition adds the following new content:

- Updated architectural diagrams
- Expanded scripting examples
- Debugging Event Handlers
- Developing an advanced report item with data binding
- Developing a Data Extraction Extension
- Developing a Charting Extension

Who should read this book

This book is intended for people who have a programming background. These readers can be categorized as:

- Embedders and integrators

  These individuals work with the software to integrate it into their current application infrastructure.

- Extenders

  These individuals leverage APIs and other extension points to add capability or to establish new interoperability between currently disparate components or services.

To write scripts in report design, you need knowledge of JavaScript or Java. More advanced tasks, such as extending BIRT's functionality, require Java development experience and familiarity with the Eclipse platform.

Contents of this book

This book is divided into several parts. The following sections describe the contents of each of the parts and chapters.

Part 1, Installing and Deploying BIRT

Part 1 introduces the currently available BIRT reporting packages, the prerequisites for installation, and the steps to install and update the packages. Part 1 includes the following chapters:

- *Chapter 1. Introducing BIRT Report Designers.* BIRT provides a number of separate packages as downloadable archive (.zip) files on the Eclipse web site. Some of the packages are stand-alone modules, others require an existing Eclipse environment, and still others provide additional functionality to report developers and application developers. This chapter describes the prerequisites for each of the available packages.

- *Chapter 2. Installing a BIRT Report Designer.* BIRT provides two report designers as separate packages, which are downloadable archive (.zip) files on the Eclipse web site. This chapter describes the steps required to install each of the available report designers.
- *Chapter 3. Installing Other BIRT Packages.* This chapter describes the steps required to install and update each of the available packages.
- *Chapter 4. Deploying a BIRT Report to an Application Server.* This chapter introduces the distribution of reports through an application server such as Apache Tomcat, IBM WebSphere, or BEA WebLogic. The instructions in the chapter provide detailed guidance about deploying a BIRT report to Apache Tomcat version 6.0. From those instructions, a developer can infer how to deploy to other versions.

Part 2, Understanding the BIRT Framework

Part 2 introduces the BIRT architecture and the Report Object Model (ROM) and provides background information that helps programmers design or modify reports programmatically, instead of using the graphical tools in BIRT Report Designer. Part 2 includes the following chapters:

- *Chapter 5. Understanding the BIRT Architecture.* This chapter provides an architectural overview of BIRT and its components, including the relationships among the BIRT components and BIRT's relationship to Eclipse and Eclipse frameworks. Architectural diagrams illustrate and clarify the relationships and workflow of the components. The chapter also provides brief overviews of all the major BIRT components.
- *Chapter 6. Understanding the Report Object Model.* This chapter provides an overview of the BIRT ROM. ROM is a specification for a set of XML elements that define both the visual and non-visual elements that comprise a report design. The ROM specification includes the properties and methods of those elements, and the relationships among the elements.

Part 3, Scripting in a Report Design

Part 3 describes how a report developer can customize and enhance a BIRT report by writing event handler scripts in either Java or JavaScript. Part 3 includes the following chapters:

- *Chapter 7. Using Scripting in a Report Design.* This chapter introduces the writing of a BIRT event handler script in either Java or JavaScript, including the advantages and disadvantages of using one language over the other. BIRT event handlers are associated with data sets, data sources, and report items. BIRT fires specific events at specific times in the processing of a report. This chapter identifies the events that BIRT fires and describes the event firing sequence.

- *Chapter 8. Using JavaScript to Write an Event Handler.* This chapter discusses the coding environment and coding considerations for writing a BIRT event handler in JavaScript. This chapter describes several BIRT JavaScript objects that a developer can use to get and set properties that affect the final report. The BIRT JavaScript coding environment offers a pop-up list of properties and functions available in an event handler. A JavaScript event handler can also use Java classes. This chapter includes a tutorial that describes the process of creating a JavaScript event handler.
- *Chapter 9. Using Java to Write an Event Handler.* This chapter discusses how to write a BIRT event handler in Java. BIRT provides Java adapter classes that assist the developer in the creation of Java event handlers. The report developer uses the property editor of the BIRT Report Designer to associate a Java event handler class with the appropriate report element. This chapter contains a tutorial that steps through the Java event handler development and deployment process. This chapter also describes the event handler methods and their parameters.
- *Chapter 10. Working with Chart Event Handlers.* This chapter describes the BIRT event handler model for the Chart Engine. The model is similar to the model for standard BIRT report elements and supports both the Java and JavaScript environments. This chapter provides details on both environments. The Chart Engine also supports this event model when used outside of BIRT.
- *Chapter 11. Using Scripting to Access Data.* This chapter describes how to access a data source using JavaScript code. A data source that you access using JavaScript is called a scripted data source. With a scripted data source, you can access objects other than an SQL, XML, or text file data source. A scripted data source can be an EJB, an XML stream, a Hibernate object, or any other Java object that retrieves data.
- *Chapter 12. Debugging Event Handlers.* This chapter describes how to use the BIRT report debugger to debug BIRT reports and event handlers written in JavaScript and Java. The BIRT report debugger enables users to run reports step by step, set breakpoints, inspect instance properties, and watch and evaluate variable values in certain contexts. The debugger supports advanced report developers in spotting the errors in report designs quickly, and makes it easy to understand and identify exceptions during run time. The chapter provides a tutorial with step-by-step instructions about debugging Java and JavaScript report event handlers.

Part 4, Integrating BIRT into Applications

Part 4 describes the public APIs that are available to Java developers, except the extension APIs. Part 4 includes the following chapters:

- *Chapter 13. Understanding the BIRT APIs.* This chapter introduces BIRT's public API, which are the classes and interfaces in three package hierarchies:
  - The report engine API, in the org.eclipse.birt.report.engine.api hierarchy, supports developers of custom report generators.
  - The design engine API, in the org.eclipse.birt.report.engine.api hierarchy, supports the development of custom report designs.
  - The chart engine API, in the org.eclipse.birt.chart hierarchy, is used to develop a custom chart generator.
- *Chapter 14. Programming Using the BIRT Reporting APIs.* This chapter describes the fundamental requirements of a reporting application and lists the BIRT API classes and interfaces that are used to create a reporting application. This chapter describes the tasks that are required of a reporting application and provides an overview of how to build a reporting application. The org.eclipse.birt.report.engine.api package supports the process of generating a report from a report design. The org.eclipse.birt.report.model.api package supports creating new report designs and modifying existing report designs.
- *Chapter 15. Programming Using the BIRT Charting API.* This chapter describes the requirements of a charting application, either in a stand-alone environment or as part of a reporting application. The org.eclipse.birt.chart hierarchy of packages provides the charting functionality in BIRT. By describing the fundamental tasks required of charting applications, this chapter introduces the API classes and interfaces that are used to create a chart. This chapter also describes the chart programming examples in the chart examples plug-in.

Part 5, Working with the Extension Framework

Part 5 shows Java programmers how to add new functionality to the BIRT framework. By building on the Eclipse platform, BIRT provides an extension mechanism that is familiar to developers of Eclipse plug-ins. This part also provides information about how to build the BIRT project for developers who need access to the complete BIRT open source code base. Part 5 includes the following chapters:

- *Chapter 16. Building the BIRT Project.* This chapter explains how to download BIRT 2.6 source code and build the BIRT project for development. This chapter describes how to configure an Eclipse workspace, download BIRT source code, and build the BIRT report and web viewers.
- *Chapter 17. Extending BIRT.* This chapter provides an overview of the BIRT extension framework and describes how to use the Eclipse Plug-in Development Environment (PDE) and the BIRT extension points to create, build, and deploy a BIRT extension.

- *Chapter 18. Developing a Report Item Extension.* This chapter describes how to develop a report item extension. The rotated text extension example is a plug-in that renders the text of a report item as an image. The extension rotates the image in the report design to display the text at a specified angle. This chapter describes how to build the rotated text report item plug-in and add the report item to the BIRT Report Designer using the defined extension points.
- *Chapter 19. Developing a Report Rendering Extension.* This chapter describes how to develop a report rendering extension using the Eclipse PDE with sample CSV and XML report rendering extensions as the examples. The chapter describes how to extend the emitter interfaces using the defined extension points to build and deploy a customized report rendering plug-in that runs in the BIRT Report Engine environment.
- *Chapter 20. Developing an ODA Extension.* This chapter describes how to develop several types of DTP ODA extensions. The CSV ODA driver example is a plug-in that reads data from a CSV file. The Hibernate ODA driver example uses Hibernate Query Language (HQL) to provide a SQL-transparent extension that makes the ODA extension portable to all relational databases. This chapter shows how to develop an ODA extension to the BIRT Report Designer 2.6.0 user interface that allows a report designer to select an extended ODA driver.
- *Chapter 21. Developing a Data Extraction Extension.* This chapter describes how to develop a report data extraction extension using the Eclipse PDE. The data extraction example exports report data to XML format. The Eclipse PDE provides the basis for the extension.
- *Chapter 22. Developing a Fragment.* This chapter describes how to build a fragment. The BIRT Report Engine environment supports plug-in fragments. A plug-in fragment is a separately loaded package that adds functionality to an existing plug-in, such as a specific language feature in a National Language Support (NLS) localization application. The example in this chapter creates a Java resource bundle that adds translations to the messages defined in the messages.properties files for the org.eclipse.birt.report.viewer plug-in.
- *Chapter 23. Developing a Charting Extension.* This chapter discusses the process of adding a new chart type to the BIRT chart engine. BIRT provides a radar chart type with the chart engine as an example of how to build new chart types. This chapter describes how to create this radar chart example.

The *Glossary* contains a glossary of terms that are useful to understanding all parts of the book.

Typographical conventions

Table P-1 describes the typographical conventions that are used in this book.

Table P-1 Typographical conventions

Item	Convention	Example
Code examples	Monospace font	`StringName = "M. Barajas";`
File names	Initial capital letter, except where file names are case-sensitive	SimpleReport.rptdesign
Key combination	A + sign between keys means to press both keys at the same time	Ctrl+Shift
Menu items	Capitalized, no bold	File
Submenu items	Separated from the main menu item with a small arrow	File➛New
User input	Monospace font	`2008`
User input in Java code	Monospace font italics	`chkjava.exe` *`cab_name`*`.cab`

Syntax conventions

Table 3-1 describes the symbols that are used to present syntax.

Table 3-1 Syntax conventions

Symbol	Description	Example
[ ]	Optional item	int count [= <value>];
	Array subscript	matrix[ ]
< >	Argument that you must supply	<expression to format>
	Delimiter in XML	<xsd:sequence>
{ }	Groups two or more mutually exclusive options or arguments when used with a pipe	{TEXT_ALIGN_LEFT | TEXT_ALIGN_RIGHT}
	Defines array contents	{0, 1, 2, 3}

(continues)

Table 3-1 Syntax conventions (continued)

Symbol	Description	Example

{ }	Delimiter of code block	if (itemHandle == null)
		{
		  // create a new handle
		}

|	Separates mutually exclusive options or arguments in a group	[public | protected | private] <data type> <variable name>;
	Java bitwise OR operator	int newflags = flags |4

Acknowledgments

John Arthorne and Chris Laffra observed, "It takes a village to write a book on Eclipse." In the case of the BIRT books, it continues to take a virtual village in four countries to create these two books. Our contributors, reviewers, Addison-Wesley editorial, marketing, and production staff, printers, and proofreaders are collaborating by every electronic means currently available to produce the major revisions to these two books. In addition, we want to acknowledge the worldwide community of over one million Java programmers who have completed over ten million downloads of the multiple versions of the software. Their enthusiastic reception to the software creates an opportunity for us to write about it.

We want to thank Greg Doench, our acquisitions editor, who asked us to write a book about BIRT and has been supportive and enthusiastic about our success. Of course, we want to acknowledge the staff at Addison-Wesley who worked on the first and second editions and this third revision. In particular, we would like to acknowledge John Fuller, Michelle Housley, Anne Jones, Mary Kate Murray, Julie Nahil, Stephane Nakib, Elizabeth Ryan, Sandra Schroeder, Beth Wickenhiser, and Lara Wysong. We also want to thank Mike Milinkovich at the Eclipse Foundation and Mark Coggins at Actuate Corporation for providing the forewords for the books.

We particularly want to acknowledge the many, many managers, designers, and programmers too numerous to name who have worked diligently to produce, milestone by milestone, the significant upgrades to BIRT, giving us a reason for these two books. You know who you are and know how much we value your efforts. The following engineers have been of particular assistance to the authors: Linda Chan, Xiaoying Gu, Wenbin He, Petter Ivmark, Rima Kanguri, Nina Li, Wenfeng Li, Yu Li, Jianqiang Luo, Kai Shen, Aniruddha Shevade, Pierre Tessier, Yulin Wang, Mingxia Wu, Gary Xue, Jun Zhai, and Lin Zhu. We want to recognize the important contribution of David Michonneau in the area of charting. We wish to acknowledge Zhiqiang Qian who created a data bound report item extension used to develop the report item extension chapter. Yasuo Doshiro worked closely with the authors on "Developing a Fragment," which provides suggestions for the practical application of BIRT technology to the challenges of translation and localization. Doshiro manages BIRT translation and develops and executes test plans. In addition, we want to acknowledge the support

and significant contribution that was provided by Paul Rogers. Dan Melcher's and Daniel O'Connell's insights into the techniques for building reusable components that can be applied to building internationalized reports. Working examples are to be found at

`http://reusablereporting.blogspot.com/`

Creating this book would not have been possible without the constant support of the members of the Developer Communications team at Actuate Corporation. Many of them and their families sacrificed long personal hours to take on additional tasks so that members of the team of authors could create this material. In particular, we wish to express our appreciation to Terry Ryan for his work on the glossary that accompanies each of the books. In addition, Mary Adler, Minali Balaram, Frances Buran, Bruce Gardner, Shawn Giese, Mike Hovermale, Melia Kenny, James Monaghan, Lois Olson, James Turner, Jeff Wachhorst, C. J. Walter-Hague, and Forest White all contributed to the success of the books.

Actuate's active student intern program under the Executive Sponsorship of Dan Gaudreau, Chief Financial Officer, made it possible for Maziar Jamalian and Rodd Naderzad to support the project in Developer Communications while actively engaged in pursuing undergraduate and graduate degrees in accounting, business, and information science at several different universities.

Chapter

7

Using Scripting in a Report Design

BIRT provides a powerful scripting capability that enables a report developer to create custom code to control various aspects of report creation. This chapter provides an overview of scripting in BIRT. Subsequent chapters focus on implementing script event handlers in JavaScript and Java, writing event handlers for charts, and accessing data using scripted data sources.

Overview of BIRT scripting

BIRT Report Designer supports writing custom event handlers in both Java and JavaScript. BIRT RCP Report Designer supports writing custom event handlers only in JavaScript.

Choosing between JavaScript and Java

Both JavaScript and Java have advantages and disadvantages when writing an event handler. For a developer who is familiar with only one of the two languages, the advantage of using the familiar language is obvious, but for others the decision depends on the report requirements.

The advantages of using JavaScript to write an event handler include:

- Ease of adding a simple script for a particular event handler

  Adding a JavaScript event handler to a report is less complicated than adding an event handler written in Java. To write a JavaScript event handler, there is no need to create a Java environment in Eclipse or to learn the Eclipse Java development process. There is no requirement to specify a package, implement an interface, or know the parameters of the event handler you write.

To add a JavaScript event handler, first select the name of the event handler from a drop-down list on the Script tab. Then, type the code.

- Simpler language constructs, looser typing, and less strict language rules

  JavaScript is less demanding to code than Java due to its more relaxed requirements.

The advantages of using Java to write an event handler include:

- Availability of the Eclipse Java development environment

  The Eclipse Java development environment is very powerful and includes such features as autocompletion, context sensitive help, keyboard shortcuts, and parameter hints.

- Ease of finding and viewing event handlers

  All the Java event handlers for a report exist in readily viewable Java files. By contrast, the JavaScript event handlers are embedded in the design and you can view only one handler at a time.

Using both JavaScript and Java

There is no requirement to write all event handlers in one language. You can write some in JavaScript and others in Java. If both a JavaScript and a Java event handler for the same event are available, BIRT uses the JavaScript handler.

Events overview

When writing event handlers, understanding the event order is imperative. The order in which events fire depends on several factors. These factors include which BIRT processing phase is executing, the engine task executing the process, and what event type is processing.

Engine task processes

The scripting chapters make continuous reference to engine task processes. This section provides an overview of what these processes are and how they affect scripting.

The Report Engine that executes reports is task-oriented and provides three tasks related to the execution and rendering of reports. These tasks are RunAndRenderTask, RunTask, and RenderTask.

The RunAndRenderTask uses one process to open the report design and produce a specific output, such as PDF. The RunTask opens a report design and executes the report producing a report document file with a

.rptdocument extension. This report document is an intermediate binary file that can then be used by a RenderTask to produce a report output type, such as HTML, PDF, PPT, PS, Word, or XLS. The RenderTask opens a report document (.rptdocument) and renders the appropriate output format. This task can be executed any time after a RunTask. This task can even occur on a separate system.

BIRT Report Designer uses the report engine to show a preview of a report. Selecting Preview in the Editor launches a RunAndRenderTask to produce the report. Selecting any other previewing option in BIRT Report Designer toolbar launches a RunAndRenderTask to produce the output, with the exception of the Run report in BIRT Web Viewer, which produces the report using a RunTask followed by a RenderTask.

The tasks RunTask and RenderTask can be used when the tasks need to occur at separate times or on separate systems. Using separate run and render tasks therefore requires two processes to run and render a report to a particular output format.

When RunAndRenderTask processes a report, the event firing order is different from the RunTask and RenderTask are used as two separate processes. The differences in processing are discussed throughout these chapters.

The following sections describe how the report engine processes reports in the BIRT Report Designer environment.

BIRT Web Viewer

The example BIRT Web Viewer application is a J2EE application that encapsulates the report engine to produce reports. This viewer contains three Servlet mappings used to generate reports. These are the frameset, run, and preview mappings.

In the example web viewer, two processes, RunTask and RenderTask, generate and render to the output format to create a report document. Selecting the Export report icon in the Web Viewer toolbar initiates a RenderTask to execute on the current report document.

BIRT processing phases

The BIRT services for generating and presenting report data create a report during the following processing phases:

- Preparation

  RunTask or RunAndRender Task prepare the report items for execution.

- Generation

  RunTask or RunAndRender Task create an instance of each report item, connect to the data source, execute the data sets, and process the data to produce the report.

- Presentation

  RenderTask or RunAndRender Task select the correct emitters to produce the output specified for the report.

The RunTask handles the preparation and generation phases and the Render Task handles the presentation phase. RunAndRenderTask handles all phases of report processing. The types of events and the order in which these events fire depends on the processing phase currently executing in an engine task.

BIRT event types

BIRT supports the following types of events:

- Parameter
- Report design
- Data source and data set
- Report item

Each event type has a series of events that fire during report processing.

Parameter events

BIRT currently supports two parameter level events that can be used to customize the parameter selection choices and default values before the parameter entry box is displayed. In addition, BIRT supports one parameter level event that is called after the parameters have been selected. These events are summarized in Table 7-1. These events are currently available only in JavaScript. Parameter events fire only if the report contains parameters.

Table 7-1 Parameter events

Event	Description
getDefaultValueList	Sets the parameter's default value. If the parameter is set to allow multiple values, this script should return an array of values that will then be pre-selected in the parameter entry box. This event fires first.
getSelectionValue List	Returns a single value or an array of values. This event fires before the parameter entry box displays for parameters presenting a list of values. Simple text parameters do not call this event.
validate	This event fires after the user selects the report parameters and before the initialize event. This event fires for each parameter and returns a value of true or false. Returning false throws an exception that states which parameter failed validation. Returning true processes the report normally.

Listing 7-1 provides an example of getDefaultValueList and getSelectionValueList event handlers for a parameter formatted as a list box. The array dSLArray contains four values that a report user can select. The array dVLArray contains two values that are the default values of the parameter. These values are selected in the list of available values when the user runs the report.

Listing 7-1 Example event handlers for a parameter formatted as a list box

```
//getDefaultValueList
var dVLArray = [];
dVLArray[0]= "10104";
dVLArray[1] = "10108";
dVLArray;
```

```
//getSelectionValueList
var dSLArray = [];
dSLArray[0]= "10101";
dSLArray[1]= "10104";
dSLArray[2]= "10105";
dSLArray[3] = "10108";
dSLArray;
```

The getDefaultValueList can be useful to provide date parameters as well.

If changes to the parameter are necessary after a user enters a parameter or if extra parameter validation is required, write an event handler for the validate event. The following example shows a validate script for a string parameter:

```
params["MyParameter"].value = "Something Else";
true;
```

Report design events

Report design events fire for all reports. Table 7-2 describes these events. The events support event handlers to provide additional processing of the report.

Table 7-2 Report design events

Event	Description
initialize	Fires every time a task accesses the report design (.rptdesign) or the report document (.rptdocument). This event occurs once when using RunAndRenderTask. When the processing phases run separately, RunTask triggers once for the generation phase and RenderTask triggers once for every render operation. Displaying the first page of a report in the presentation phase and using the page navigation control to access a new page also triggers this event handler.

(continues)

Table 7-2 Report design events (continued)

Event	Description
beforeFactory	Fires just prior to the generation phase after the elements in the report have been prepared in the preparation phase. This event handler fires only once. Use beforeFactory( ) when modifications to the report design are required before execution.
beforeRender	Fires just prior to the presentation phase and is called for every render operation. In the RunAndRender task, this event occurs once just after the beforeFactory event.
afterFactory	Fires at the conclusion of the generation phase and fires only once. If using a RunTask or RunAndRenderTask, this event is the last one fired. If using a RenderTask, this event does not execute.
afterRender	Fires at the conclusion of the presentation phase for a specific render operation. If using a single RunAndRenderTask process, this event is called only once. If using separate RunTask and RenderTask processes, this event fires for every render operation.
onPageStart	Fires prior to placing content on a specific page and supports the use of page-level variables to modify the page. This event fires only when processing phases run separately, for example, a RunTask followed by a RenderTask. This event fires prior to the onPageBreak events for the individual elements to be placed on the page. This event is also available on the master page.
onPageEnd	Fires after all the onPageBreak events have fired for the content to be placed on a page and prior to evaluating the autotext items for the page. This event is also available on the master page.

Types of data source and data set events

There are several kinds of data sources and data sets. A data source can be a flat file, a JDBC data source, a scripted data source, a web services data source, or an XML data source. All data sources have a common set of event handlers. A scripted data source has two additional event handlers.

Data set events can be called multiple times to support multipass aggregation and data set sharing. It is not advisable to write event handlers that rely on the data set event firing order. An additional issue is that data set event handlers may only be called once because of data set caching.

Table 7-3 describes the data source events.

Table 7-3 Data source events

Event	Description
beforeOpen	This event is most often used to modify public properties of the data source, including database URL, username, password, driver class and JNDI URL. This event fires only once, before the connection to the data source opens event, when the connection uses multiple data sets. Fires prior to opening a connection to a data source.
afterOpen	Fires after the connection to the data source opens.
beforeClose	Fires at the conclusion of the generation phase, just prior to closing the data source connection.
afterClose	Fires after the data source connection closes.
open	Fires only for a scripted data source, providing a location for the developer to set up a connection to an external source.
close	Fires only for a scripted data source, providing a location for the developer to close a connection to an external source.

Data source and data set events fire in the generation phase prior to the onCreate event of the report item that uses them. Data set events can fire several times. The event order is covered in more detail later in this chapter. Table 7-4 describes the data set events.

Table 7-4 Data set events

Event	Description
beforeOpen	Fires prior to opening a data set and used most often to modify public properties of the data set. For example, when using a JDBC data set, the query text can be altered using this event. This event fires for every report item bound to the data set. If two tables use the same data set, the data set is called twice, resulting in this event firing twice. BIRT 2.6 allows report items to be bound to other report items, in which case the data set is called only once, resulting in the event triggering only once. Data set caching options continue to be added which affect how often data source and data set events fire.
afterOpen	Fires after the data set opens.

(continues)

Table 7-4 Data set events (continued)

Event	Description
onFetch	Fires as the data set retrieves each row of data. This event fires for all rows of data before the onCreate event for the particular report item that uses the data set.
beforeClose	Fires before closing the data set after creating the report item that uses the data set.
afterClose	Fires after the data set closes.
open	Fires for a scripted data set, providing a location for the developer to set up a data set.
fetch	Fires for a scripted data set, providing a location for the developer to populate rows from the scripted data set.
close	Fires for a scripted data set, providing a location for the developer to close the set.

Data source and data set events fire prior to being used on a data bound item. If the data set is never used in the report, the data source and data set are never called. A scripted data set is a data set that accesses a scripted data source. A non-scripted data set is one that accesses a standard data source. Use the scripted data source and data set to write custom code to retrieve data values.

Scripted data source events

A scripted data source contains the afterOpen, afterClose, beforeOpen, and beforeClose events common to all data sources, and open and close events. Use the event handlers for the open and close events to perform the actions of opening and closing the data source.

Scripted data set events

A scripted data set contains the afterOpen, afterClose, OnFetch, beforeOpen, beforeClose, open, close, and fetch events. Use the event handlers for the open and close events to perform the actions of opening and closing the data set. Use the fetch method to fetch a row from the data source. Using a scripted data set requires an event handler to be written for the fetch event.

ReportItem events

ReportItem events trigger for report items that are placed in the report. Most items support writing event handlers for the events listed in Table 7-5.

Table 7-5 Report item events

Event	Description
onPrepare	Fires at the beginning of the preparation phase before data binding or expression evaluation occurs. This event is useful for changing the design of the item prior to generating the item instance.
onCreate	Fires at the time the generation phase creates the element. This event is useful when a particular instance of a report item needs alteration.
onRender	Fires in the presentation phase. This event is useful for operations that depend on the type or format of the output document.
onPageBreak	Fires for all report items currently on the page when the page break occurs. Not all report items support the onPageBreak event.

Event order sequence

Table 7-6 summarizes which engine task is responsible for a particular phase. The following sections describe the order of event firing for each phase in more detail.

Table 7-6 Engine task by phase

Report engine task	Preparation phase	Generation phase	Presentation phase
RunTask	Yes	Yes	No
RenderTask	No	No	Yes
RunAndRenderTask	Yes	Yes	Yes

Preparation phase operation

The preparation phase includes parameter validation as well as initialization and report element preparation for every element in the report. The preparation phase is identical for all reports. Table 7-7 lists the event types for RunTask and RunAndRenderTask in the order in which the events execute.

The RenderTask does not have a preparation phase. In a RunAndRenderTask, the preparation phase triggers a beforeRender event.

Table 7-7 Preparation phase events

Element type and event	RunTask	RunAnd RenderTask
Parameter getDefaultValueList	Yes	Yes
Parameter getSelectionValueList	Yes	Yes
Parameter validate	Yes	Yes
ReportDesign Initialize	Yes	Yes
ReportItem onPrepare (iterative)	Yes	Yes
ReportDesign beforeFactory	Yes	Yes
ReportDesign beforeRender	No	Yes

The parameter events fire first for each parameter. Before the parameter entry box is displayed, the getDefaultValueList is called for each parameter. This script returns either a single value or a list of values, depending on the parameter type.

Next, each parameter that provides a list of choices to the user calls the getSelectionValueList. If running reports using the API for RunTask or RunAndRenderTask, the getDefaultValueList script event fires for only parameters that have no value passed. The getSelectionValueList is not called. If using createGetParameterDefinitionTask to build your own parameter entry screen, these events fire much like the example Viewer.

After a user enters values for the parameters, the next event is the validate event, which fires for each parameter. The ReportDesign event, initialize, follows. The initialize event fires once when using only one engine task. Using a separate engine task triggers the initialize event at least two times, one time for the preparation phase, and a second time for the render phase. Additional render tasks or phases also trigger the initialize method.

After the initialize event, the preparation phase triggers the onPrepare event for every element in the report. This process starts with the master page content and proceeds from left to right and top to bottom in the report body. All nested elements process before proceeding to the next element.

The beforeFactory event fires next. This event signals that report creation is about to occur and provides a location for altering a running report.

The beforeRender event fires when using a RunAndRenderTask. When using two tasks, this event does not fire until a render operation occurs.

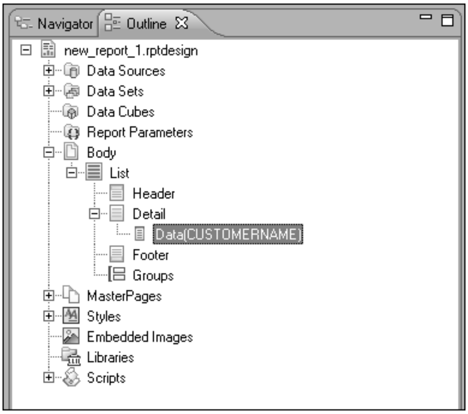
Generation phase operation

The generation phase includes connecting to data sources, executing data sets and data cubes, evaluating data bindings, and creating all the report items in the report. The data source and data set events fire before the creation of data-bound items, but this processing may not occur before the creation of other report items. For example, if a table is bound to a data set and the report uses a master page with only a label in the footer, the master page content onCreate event fires before the data source and data set events for the data set bound to the table.

This phase triggers the onCreate event for every report item. BIRT processes the report body after processing content on the master page. The report body contains all the report items to be created and rendered. BIRT processes all items at all levels in an iterative fashion, following the same process at each level as it does for the top-level items. A report item that is not contained in another report item is called a top-level report item. BIRT processes the top-level items, going from left to right and proceeding a row at a time toward the bottom right. Every report has at least one top-level report item, usually a grid, list, or table. All nested elements complete OnCreate processing before the next element's onCreate fires.

For each top-level item, BIRT processes all the second-level items before proceeding to the next top-level item. A second-level report item is a report item that is contained within a top-level item. For example, a table contained in a grid is a second-level report item.

There can be any number of levels of report items. To see the level of a particular report item, examine the structure of the report design in Outline, in the BIRT Report Designer, as shown in Figure 7-1.



- new_report_1.rptdesign
  - Data Sources
  - Data Sets
  - Data Cubes
  - Report Parameters
  - Body
    - List
      - Header
      - Detail
        - Data[CUSTOMERNAME]
      - Footer
      - Groups
  - MasterPages
  - Styles
  - Embedded Images
  - Libraries
  - Scripts

Figure 7-1 Outline showing the level of a report item

If a RunAndRenderTask process executes the generation phase, each element is created and immediately rendered, which fires the onRender event before proceeding to the next element. If a RunTask process executes the generation phase, the onRender events do not fire.

Table 7-8 lists the events triggered in the generation phase for each major report component in the order in which these events execute. The data source, data set, and onPageBreak events are optional.

Table 7-8 Generation phase events

Report component	RunTask	RunAndRenderTask
Master Page content	Data source and data set events (optional) onCreate onPageBreak (optional)	Data source and data set events (optional) onCreate onRender onPageBreak (optional)
Body (iterative)	Data source and data set events (optional) onCreate onPageBreak (optional)	Data source and data set events (optional) onCreate onRender onPageBreak (optional)

Data source and data set events do not fire if a report item is not data bound or if the report item is bound to another report item that executed previously. The onPageBreak event fires only when an actual page break occurs.

The next three sections describe the data source, data set, data binding, and page break events in more detail.

About data source and data set events

Events for data source and data set elements fire just prior to creating the report item bound to the data set. This sequence occurs for every report item bound to a data set with the exception of the data source beforeOpen and afterOpen events. These events do not fire if a data source has already been used. When a report item is bound to another report item, none of these events fire for the bound report item, supporting two data-bound items sharing one data set without re-executing the data set.

The data source beforeClose and afterClose events fire at the end of the generation phase just before the afterFactory event fires. In the generation phase, there is no difference in data source or data set processing in RunTask and RunAndRenderTask.

BIRT supports data set caching, by which data sets execute only once and are used multiple times. In this case, the data set events fire on only the first use. If a data set is parameterized and the actual parameter value is changed within a report item, the query executes again using the new parameter value. This query execution causes the data set events to fire again. Table 7-9 lists the data source and data set types and events in the order in which these events execute.

Table 7-9 Data source and data set events

Event type and event	RunTask	RunAnd RenderTask
Data source beforeOpen	Yes	Yes
Data source afterOpen	Yes	Yes
Data set beforeOpen	Yes	Yes
DataSet afterOpen	Yes	Yes
DataSet onFetch for all rows of data	Yes	Yes
Process the data-bound Report Item	Yes	Yes
DataSet beforeClose	Yes	Yes
DataSet afterClose	Yes	Yes

About data binding

Data binding in BIRT makes a logical separation between BIRT data sets and data-bound elements, such as tables and lists. Selecting a table or list and then the Binding tab in the Property Editor displays the current bindings for the item. A report developer can create bound columns that use external objects to calculate a specific column value. The bound data set continues to determine the number of rows that are processed for a given table or list.

In a table or list, evaluation of the data bindings for the detail rows occurs before the onCreate event on the current row. This approach supports the onCreate event handler retrieving the appropriate values for the row.

The bindings are evaluated for each detail row in a bound data set. Table footers containing data elements that use a binding are evaluated before the last onCreate event for the final detail row.

Trying to alter a bound column using the onCreate script is discouraged. For example, it is possible to concatenate a column value for a row with all previous rows for the given column using the onCreate script with a JavaScript variable. If this value is in the expression of a bound column, the last value is excluded from the bound column. To display the last value, use a computed column, an aggregation report item, or the dynamic text <VALUE-OF> tag in a text element to display the JavaScript value.

A report developer must not assume any particular evaluation order of binding expressions relative to the various table item events, except that a binding evaluates before the onCreate event of an item that uses it. Best practice is to perform any required manipulation of the data in a data set script or the binding expression.

Data binding evaluation for a group evaluates the bound column many times before creating the table or list item. BIRT performs multiple passes over the data to support grouping. Any script in a group-bound column expression is called every time the data binding evaluates, affecting report performance.

All bindings of a table are always calculated, whether or not these bindings are used by the table. For performance reasons, check the binding list and remove any unused items from the report design.

Report items can share data bindings. For example, two tables can share the same set of bindings. The second table does not re-evaluate the data bindings. To use this feature in the BIRT Report Designer, select the second table, choose Binding in Property Editor, select Report Item, and select the appropriate report item from the list. The list supports only named data-bound report items.

The above scenarios affect any BIRT expression that uses the Available Column Bindings category.

About page break events

Many report items support page breaks. Setting a page break interval on a table instructs the table to process a certain number of rows and then apply a page break. Setting a page break on a group section applies a page break both before and after or only after the group processes. A page break can be applied before many other report items. For example, applying a page break before a row in a table instructs the engine to apply a page break before every row of data that a table processes.

Page events are based on the paginated HTML emitter. If a report generates 5 HTML pages but has 20 PDF pages, the page events fire only 5 times.

The general order of the events is as follows.

- The engine creates the page.
- The onPageStart event fires for the report.
- The onPageStart event fires for the selected master page.
- For items that are to appear on the page, the onPageBreak event fires.
- The onPageEnd event fires for the selected master page.
- The onPageEnd event for the report fires.
- The engine evaluates the Master Page Auto Text fields.

The onCreate events for all items that are to appear on a page fire before the report's onStart event. Some items have their onCreate triggered but do not appear on a page because of the keep together flag. Such items appear on the following page.

The page-level scripts are used when the Run and Render tasks are separated.

About chart event order

Chart events are handled by the chart engine and execute within the presentation phase. These events are covered in a later chapter.

About table and list event order

The BIRT engine fires the onCreate event for report items iteratively. When processing report items that iterate over data rows, such as a table or list, the event order changes to add additional events for each row of data. Row containers process data rows.

Row execution sequence

Tables, lists, and groups have rows. BIRT processes all rows identically. There are three kinds of rows:

- Header
- Detail
- Footer

Figure 7-2 illustrates the execution sequence for a row. A list or table can contain multiple elements, such as detail rows or table header rows.

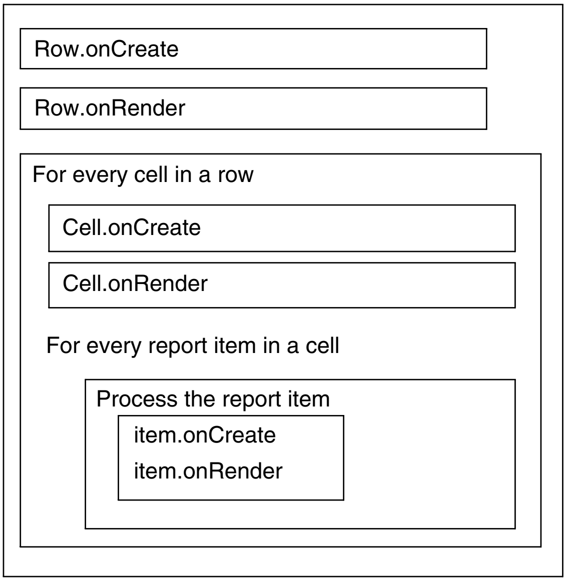


Figure 7-2 Row execution sequence

Table and list method execution sequence

A list is equivalent to a table that has only a single cell in every row. BIRT processes tables and lists identically, except that for a list, BIRT does not iterate through multiple cells. BIRT processes tables in three steps, the setup, detail, and wrap-up processing steps.

The following sections describe the table and list execution sequence steps.

Table and list setup step

The pre-table processing step is the same for all tables, both grouped and ungrouped.

Figure 7-3 illustrates the execution sequence for the pre-table processing step. This illustration shows the event order when using a RunAndRenderTask in the generation phase.

A RunTask does not fire onRender events. A RenderTask does not fire onCreate, data source, and data set events. The data source and data set events are optional.

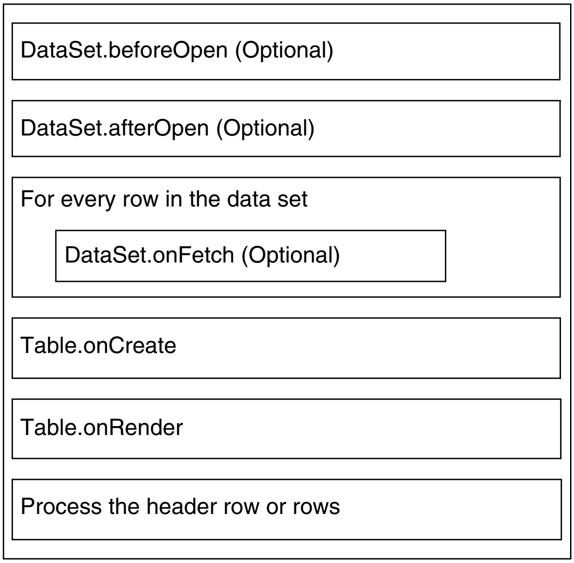

DataSet.beforeOpen (Optional)
DataSet.afterOpen (Optional)
For every row in the data set
DataSet.onFetch (Optional)
Table.onCreate
Table.onRender
Process the header row or rows

Figure 7-3 Table and list setup execution sequence

Table and list processing step

The sequence for the table and list processing step depends on whether the table or list is grouped. A table or list having no grouping has a different sequence from one that has grouping.

Figure 7-4 illustrates the execution sequence for a table or list without grouping.

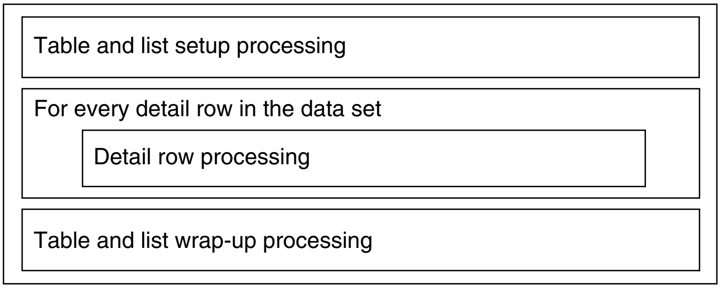


Table and list setup processing
For every detail row in the data set
Detail row processing
Table and list wrap-up processing

Figure 7-4 Ungrouped table or list detail execution sequence

For a table having grouping, BIRT creates one ListingGroup item per group.

A ListingGroup is similar to a table in that it has one or more header rows and one or more footer rows. BIRT processes grouped rows in the same way that it processes a table row.

In addition to the standard onPrepare, onCreate, and onRender events for these rows, the ListingGroup fires the onPageBreak and onPrepare events for the group. To access the script location for these event handlers, locate the group in the outline view and select the script tab. Use code in the onPrepare event handler to modify grouping behavior, such as changing the sort order.

Figure 7-5 illustrates the method execution sequence for a table that has groups.

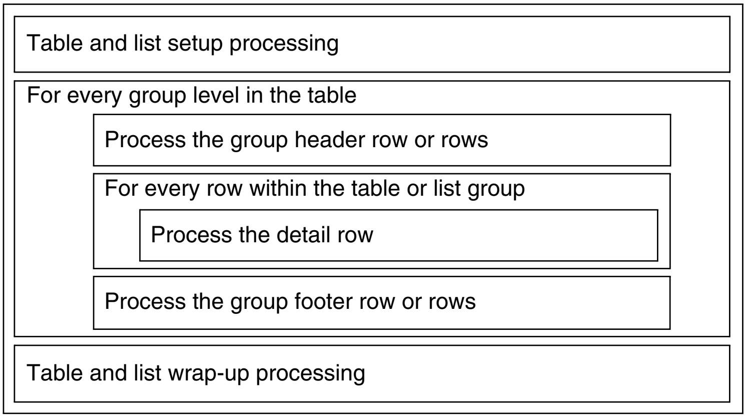


Table and list setup processing
For every group level in the table
Process the group header row or rows
For every row within the table or list group
Process the detail row
Process the group footer row or rows
Table and list wrap-up processing

Figure 7-5 Grouped table execution sequence

To verify the execution sequence of event handlers for a specific report, add logging code to the event handlers.

Table and list wrap-up step

The post-table processing step is the same for all tables, both grouped and ungrouped.

Completion of the generation phase

On completion of the generation phase, all data sources, beforeClose and afterClose events fire, followed by the afterFactory event. If using the RunAndRenderTask, the afterRender event fires before closing the data sources. Table 7-10 describes the events available at completion of the generation phase.

Table 7-10 Completing the generation phase

Element type and event	RunTask	RunAndRenderTask
afterRender	No	Yes
Data Source(s) beforeClose	Yes	Yes
Data Source(s) afterClose	Yes	Yes
afterFactory	Yes	No

Presentation phase operation

The presentation phase launches the appropriate emitter and produces report output based on the generated report. This phase fires the onRender events for all items as they are created. If using the RenderTask to render an existing report document, the initialize event triggers first and then each rendered report item's onRender event fires.

If the RenderTask renders pages individually, the onRender event fires only for items that are rendered. For example, BIRT Web Viewer uses a RunTask to create a report document. The web viewer uses a RenderTask to render the first page. This action fires the initialize event first and the onRender event for each item that appears on the page. Selecting a new page and using the pagination controls results in a new RenderTask that calls the initialize event again and triggers the onRender event for each item on the new page.

Event order summary

Table 7-11 summarizes the report design and report item events triggered in each processing phase. The events are listed in the order they are fired for a particular task. Page break, getDefaultValueList, getSelectionValueList, data source, and data set events are not shown for brevity.

Table 7-11 Event order summary

Element type and event	RunTask	RunAnd Render Task	Render Task	Notes
Parameter validate	Yes	Yes	n/a	Only available in JavaScript.

Table 7-11 Event order summary

Element type and event	RunTask	RunAnd Render Task	Render Task	Notes
Report initialize	Yes	Yes	Yes	Called multiple times using the RenderTask.
MasterPage Content onPrepare	Yes	Yes	n/a	
Body Iterate onPrepare	Yes	Yes	n/a	
Report beforeFactory	Yes	Yes	n/a	
Data source and data set beforeOpen and afterOpen	Yes	Yes	n/a	
Report beforeRender	n/a	Yes	Yes	
MasterPage Content onCreate	Yes	Yes	n/a	
Body Iterate onCreate	Yes	Yes	n/a	
Page Events	Yes	Yes	n/a	
Body Iterate onRender	n/a	Yes	Yes	RunAndRenderTask triggers this event immediately after onCreate for a report item.
Report afterRender	n/a	Yes	Yes	
Data source and data set beforeClose and afterClose	Yes	Yes	n/a	
Report afterFactory	Yes	Yes	n/a	

This page intentionally left blank

Index

Symbols

, (comma) character 476, 478
… button 731
" (double quotation mark) character
 command line arguments and 20
 expressions and 456
 JavaScript code and 320
web-based reports
 See also web applications
\ (backslash) character 320

A

absolute paths 29
abstract base class 707
abstraction 707
AbstractPageGenerator class 449
accessing
 BIRT APIs 261
 BIRT Javadoc files 261
 BIRT packages 15
 builders 439
 chart object factory 688
 chart properties 146
 charts 148, 310, 321
 component libraries 48
 connection pool services 32
 custom data sources 510
 data 222, 456
 data sets 303
 data structures 297
 Debug perspective 196
 demo database 18
 design model objects 226
 Eclipse PDE 366
 expression builder 458, 460
 external classes 279
 external data sources 60, 510, 517
 external objects 280
 Java classes 54, 112, 113, 262
 Java objects 181
 JDBC data sources 261
 metadata 222
 methods 114
 ODA data sources 99, 509
 plug-in extension examples 384
 plug-ins 265, 359
 report components 228, 229
 report design elements 271
 report design structures 257, 289
 report designs 228, 289, 291
 report documents 288
 report elements 93, 257, 271, 289
 report items 257, 293, 294, 295
 report parameters 272, 273
 report properties 228
 report viewer 26, 28, 30
 reporting libraries 257
 reports 28, 288
 resource files 264, 291
 ROM schemas 53
 script editor 83
 scripted data sources 173
 source code 359
 styles 229
 Tomcat Manager 30
 web service data sources 184
Acrobat Reader 288
 See also PDF documents
action filters 536

action handlers 284
actions 284
  See also events
Activator class 564
activators 368
adapter class names 123
adapter classes 117, 118, 124, 241
  See also Java event handlers
adapter elements 398, 448
Add Java Exception Breakpoint command 194
Add Java Exception Breakpoint icon 194
adding
  autotext items 85
  breakpoints 193, 194
  chart types 647, 648, 652
  charts to designs 323, 327, 345, 346
  connection profiles 535
  context menus 443
  custom drivers 46
  data sources 530
  emitters 465
  lists 59
  master pages 88, 283
  ODA drivers 510
  ODA user interfaces 385, 510, 530, 537
  page breaks 76
  plug-ins (required) 370
  property editors 445
  property pages 536
  report elements 68, 227
  report item builder 439
  report items 292, 302, 387
  ROM elements 58, 59, 303
  scripted data sets 173, 176
  tables 59
  update sites 382–383
addLabel method 689
addLine method 689
addPlane method 689
addSampleData method 671
Adobe Acrobat Reader. *See* Acrobat Reader
afterClose events 69, 70, 80
afterClose method 126
afterDataSetFilled events 152
afterDataSetFilled method 143
afterDrawAxisLabel events 161
afterDrawAxisLabel method 143
afterDrawAxisTitle events 161
afterDrawAxisTitle method 143
afterDrawBlock events 158
afterDrawBlock method 143
afterDrawDataPoint events 159, 160
afterDrawDataPoint method 144
afterDrawDataPointLabel events 159, 160
afterDrawDataPointLabel method 144
afterDrawFittingCurve method 144
afterDrawLegendEntry method 144
afterDrawLegendItem events 163
afterDrawLegendItem method 163
afterDrawMarkerLine events 161
afterDrawMarkerLine method 144
afterDrawMarkerRange events 161
afterDrawMarkerRange method 144
afterDrawSeries events 158
afterDrawSeries method 144
afterDrawSeriesTitle method 144
afterFactory events 68, 80
afterFactory method 101, 103, 128
afterGeneration events 154
afterGeneration method 144
afterOpen events 69
afterOpen method 126
afterRender events 68, 80
afterRender method 128
afterRendering method 144
aggregate data. *See* aggregate values
aggregate functions 240, 649, 707
aggregate package 240
aggregate rows 707
aggregate values 708
  See also aggregating data
aggregatefunctions plug-in 649
aggregating data 46, 68, 319
aliases 96, 97, 133, 708
All-in-One Install package. *See* Full Eclipse Installation
allowExpression property 397
allowsEmptyValueAsNull attribute 519
alternate names. *See* aliases; display names
alternate text strings 90
ancestor classes 708
  See also superclasses
anchor points (charts) 158
anchor property 314

annotation element 360
annotations 360, 376
Apache Ant scripts 380
Apache Derby databases 17, 18
Apache Derby software 5
Apache Tomcat manager 30
Apache Tomcat server connections 32, 33
Apache Tomcat servers 25, 26, 30, 44
api package 217, 257, 262
API Reference Javadoc 218, 261
APIs. *See* application programming interfaces
appContext parameter 184
  See also application context
appinfo attribute 360
applets 708
application context 91, 92, 184, 280
application context class loader 114
application context map 91, 184
application programming interfaces (APIs)
  accessing 261
  BIRT engines and 44
  BIRT extensions and 359
  charts and 169, 307, 334
  defined 709
  developing with 217
  event handlers and 113
  external data sources and 510
  extracting data and 609, 614, 617, 620
  ODA drivers and 509
  rendering CSV output and 480, 486
  rendering rotated text images and 424
  rendering XML output and 494
  report designs and 49, 226, 262
  report engine and 219, 220, 257
  report generators and 49
  report item extensions and 389
  reporting applications and 50, 91
  running reports and 118
application server connections 32, 33
application server plug-in 44
application servers 4, 25, 26
  See also specific type
applications
  accessing BIRT APIs for 261
  accessing report designs and 45, 257, 289, 291
  accessing report items and 290
  accessing report parameters and 272, 273
  accessing report viewer for 28
  accessing structured content and 228, 257
  changing designs and 262
  changing source code and 349
  configuring BIRT home for 263, 264
  configuring logging environments for 270
  connecting to external sources and 510, 517
  creating stand-alone 49
  customizing 49, 359
  debugging 187, 197, 269
  defined 708
  deploying 26, 262, 266, 289
  developing 217, 258, 259
  generating charts and. *See* charting applications
  generating formatted output and 286
  generating report designs and 44, 289
  generating reports and 49, 219, 257, 262
  getting context for 87
  installing plug-ins for 259, 362
  integrating with Eclipse 39
  launching Eclipse 413
  managing service 365
  rendering CSV output and 482, 485, 488
  rendering HTML documents and 266, 267
  rendering XML output and 494, 506
  running reports and 219, 220
  setting up JNDI connections for 32, 33
  setting up platform context for 265
  sharing data and 602
  starting platforms and 267
archive files
  See also jar files; war files
  BIRT Demo Database 18
  BIRT designers and 7, 8
  BIRT packages in 15
  BIRT source code 353
  chart engine 16
  extracting 8, 9
  plug-ins and 360, 378, 381
area charts 307, 310

area charts *(continued)*
  See also charts
arguments 124, 709
  See also functions; parameters
array lists 140, 141, 273, 568
array properties 97
arrays
  chart objects and 171, 172, 315, 700
  class definitions and 54
  connection properties and 99
  defined 709
  JDBC data sources and 571
  ODA result sets and 525, 527, 572
  parameter entry boxes and 66, 277
  row objects and 97
  scripted data sets and 183
ASCII files. *See* text files
assignment statements 709
attribute 709
attribute editor 376
attribute package 240, 246, 247
attribute.xsd 652
attributes
  action filters 536
  chart model classes and 240, 246
  chart utility classes and 256
  complex properties and 296
  declaring class 376
  displaying 374, 375
  plug-in extension points and 360, 363
  report item extensions and 360
  report parameters and 272, 274
  ROM elements and 53, 54, 55
  setting allowable values for 54
  testing chart 247
autobinding charting example 341
AutoDataBindingViewer class 341
autotext report items 68, 85
axes event handlers 161
axes items. *See xDefault Para Font>series items; yDefault Para Font>series items*
axes labels 161, 162
axes properties 162, 315
axes title colors 161
axes titles 149, 161
axes types (charts) 161
axes values 156, 161, 316, 319, 685
  See also charts
AxesRenderer class 155, 648, 685
axis objects 161
axis.jar 26
axis-ant.jar 26
AxisImpl class 161

B

B2B applications 494
Babel Language Packs 11
Babel Project 635
background colors 94, 110, 165
backing up program files 12
backslash (\) character 320
bar charts 152, 160, 307, 310, 648
  See also charts
bar colors (charts) 160
bar size (charts) 148, 159
BarSeriesImpl class 317
base unit (charts) 709
base URLs (hyperlinks) 283
base URLs (images) 267, 282
BaseRenderer class 155, 648, 685, 686
BEA WebLogic servers 25
bean components 744
beforeClose events 69, 70, 80
beforeClose method 126
beforeDataSet events 151
beforeDataSetFilled events 151
beforeDataSetFilled method 144
beforeDrawAxisLabel events 161, 162
beforeDrawAxisLabel method 144
beforeDrawAxisTitle events 161
beforeDrawAxisTitle method 144
beforeDrawBlock events 158
beforeDrawBlock method 144
beforeDrawDataPoint events 159, 160
beforeDrawDataPoint method 144
beforeDrawDataPointLabel events 159, 160
beforeDrawDataPointLabel method 144
beforeDrawFittingCurve method 144
beforeDrawLegendEntry method 145
beforeDrawLegendItem events 163
beforeDrawMarkerLine events 161
beforeDrawMarkerLine method 145
beforeDrawMarkerRange events 161
beforeDrawMarkerRange method 145
beforeDrawSeries events 158
beforeDrawSeries method 145
beforeDrawSeriesTitle method 145

beforeFactory events 68, 72, 87, 88, 89, 100
beforeFactory method 68, 128, 135
beforeGeneration events 148, 153
beforeGeneration method 145
beforeOpen events 69, 99, 100, 101
beforeOpen method 98, 125, 126
beforeRender events 68, 72
beforeRender method 128
beforeRendering method 145
bidirectional text 282, 710
BigDecimal class 710
binary builds 378, 379
binary files 48
  See also report document files
binding data sets to charts 150, 326
binding data sets to report items 305, 456, 458
binding expressions 75, 76
binding report items to data sets 75, 76, 90
Binding tab 75
BIRT 39, 711
  See also BIRT technology
BIRT applications 40, 43
  See also applications
BIRT components 39
BIRT Data Tools Platform 4
BIRT Data Tools Platform Integration package 17
BIRT Demo Database 18, 711
  See also Classic Models sample database
BIRT Demo Database package 5, 17
BIRT designer components 3
BIRT documentation xxii, 16, 359
  See also Javadocs
BIRT engines 40, 44
  See also specific engine
BIRT extensions 50, 349, 359, 711
  See also extensions
BIRT home directory 259, 263
BIRT home property 263
BIRT open source projects. *See* projects
BIRT packages 4, 15, 217, 218
BIRT plug-ins. *See* plug-ins
BIRT RCP Report Designer
  See also BIRT Report Designer; rich client platforms
  accessing sample database for 17, 18
  adding charting functionality to 16
  building report designs and 3
  compared to BIRT Report Designer 44
  defined 712
  installing 8–9
  specifying JVM for 10
  starting 9, 10
  testing installations for 9
  troubleshooting 9
  updating 11, 12
  writing event handlers and 63
BIRT Report Designer
  See also designs
  accessing sample database for 17, 18
  adding charting functionality to 16
  building report designs and 3, 43
  changing source code for 349, 356
  compared to BIRT RCP Designer 44
  defined 712
  extending functionality of 47
  generating reports and 263
  installing 7, 350
  integrating with ODA drivers 46, 510
  previewing reports and 65
  specifying JVM for 10
  starting 8
  testing installations for 8
  troubleshooting 9
  updating 11–12
  writing event handlers and 63, 83
  writing Java code and 117
BIRT Report Designer Full Eclipse Install software 4, 7, 351
BIRT report object model 51
  See also ROM
BIRT Reporting APIs. *See* application programming interfaces (APIs)
BIRT reports. *See* reports
BIRT Samples package 5, 21, 713
BIRT Source Code package 5, 21
  See also source code
BIRT technology xxi, 713
BIRT web site xix
BIRT Web Tools Integration package 5, 22, 23
BIRT_FONT_PATH variable 31
BIRT_HOME variable 19, 221, 263, 309

BIRT_VIEWER_WORKING_FOLDER parameter 28, 29
BIRT.exe 9
birt.war 27
block class 158
Block containment reference property 146, 147
Block interface 310, 314
blocks (charts) 153, 158, 310
  See also chart areas
BMP device renderers 649
BMP formats 155, 171, 282, 687
bookmarks 221, 271, 286, 710
Boolean class 112
Boolean expressions 710
BoundsImpl class 163
break reports. *See* grouped reports
breakpoints 192, 193, 194, 710
Breakpoints view 192
bridge driver 711
browsers. *See* web browsers
bubble charts 310
  See also charts
BubbleDataSet interface 310
bug reports 6
Bugzilla xix
Build All command 379
Build Automatically command 380
Build Configuration page (PDE Editor) 378
build errors 356
build files 21
Build page (PDE Editor) 369, 377, 378
build paths 118, 261
build settings
  See also builds
build.properties file 360, 369, 380
Build.properties page (PDE Editor) 369
build scripts 380
build settings 369, 377
build.xml 380
buildDirectory variable 358
builder element 395
builders 439, 443, 458
  See also user interfaces
building
  BIRT open source projects 349, 354–356
  chart viewer 358
  charting API example plug-ins 340
  charting extensions 650
  CSV ODA driver plug-in 513, 521
  data extraction extensions 605
  Hibernate driver plug-in 577
  Hibernate ODA UI plug-in 594
  plug-in extensions 377–380
  plug-in fragments 636, 641, 643
  plug-ins 390
  report designer 227
  report designs 228
  report rendering extensions 469, 482, 496
  report viewer 356
  rotated label extension 389
  update sites 382
building phase (charts) 153
buildItemContent method 449
builds 5, 349, 378
  See also build settings
buildUI method 410
bulls-eye subchart type 694, 698
Bundle class 115
BundleActivator interface 364
Business Intelligence and Reporting Tools 39, 711
  See also BIRT
business-to-business applications 494
button controls 458, 460

C

cache 43, 74, 362, 687
cache conflicts 9
caching options 69
calculated columns. *See* computed fields
calculated values. *See* computed values
calculations 75, 90
canBeStacked method 658
cancelling running tasks 288
canCombine method 663
canInherit attribute 519
canInherit property 397
canParticipateInCombination method 658
capitalization. *See* case sensitivity
carriage return characters 478
cascading parameter groups 273, 276
cascading parameters 273, 276, 277, 713
  See also report parameters

cascading style sheets 52, 260, 261, 713
See also styles
CASCADING_PARAMETER_GROUP value 273
case sensitivity 714
category 714
category axis values 714
See also axes values
category element 536
category series 316, 317, 319
See also data series
Cell objects 129, 130
CellHandle class 230
cells
adding grid elements and 59, 302
building programmatically 129, 130
defined 714
determining contents of 294
placing label elements in 302
writing to CSV files and 476, 478
changeLogLevel method 269
changing
chart labels 160, 161, 163
chart properties 162, 169, 312
charts 154, 170, 239, 321
connection properties 99, 125
data series 159, 670
data set bindings 305
data sources 304, 358
grouping behavior 79
locales 11
log file names 269
logging levels 269, 270
MIME types 609
model definitions 456
plug-in project settings 369, 370
property types 456, 457
property values 98, 445
queries 98, 99
report designs 68, 87, 89, 262, 289
report elements 227, 291
report item properties 296, 297, 298
report parameters 67, 74
report templates 290
reports 72
ROM elements 54
sort order 299
source code 349, 356
styles 88, 90
URL context roots 28
character encoding. *See* encoding
character keyboard mapping 741
character sets 715
character strings. *See* strings
characters
CSV output files and 476, 478
defined 714
JavaScript code and 320
JavaScript names and 96
property type conversions and 456
trimming 522
chart areas 158, 253, 313, 652
See also chart blocks
chart attributes 652, 659
chart blocks 153, 158, 310
See also chart areas
chart builder 47, 661
See also chart wizard
chart builder examples 345
chart captions. *See* chart labels
chart components. *See* chart items
chart data binding example 341
chart device package 241
chart device renderers 649
chart dimensions 146, 170, 172, 319
chart elements 310, 715
See also charts
chart engine
accessing external content and 149
adding aggregate functions for 649
configuring 308
customizing chart types for 647, 652, 661
defined 711
downloading plug-ins for 650
generating charts and 43, 150, 153, 154
handling events for 143, 150, 311
handling exceptions for 244
installing 15–16
logging tasks for 245
overview 45
chart engine API 217, 239
chart engine API library 260
chart engine archive 16, 307
chart engine classes 239, 240
chart engine documentation 16
chart engine drawing framework 686, 687

chart engine extension points 648
chart engine generator 150, 217
chart engine package 4
chart engine plug-in 652
chart event handler examples 346
chart event handler model 143
chart event handlers
  developing 143, 164, 167, 320
  getting chart objects for 170
  getting external context for 149
  rendering charts and 155, 156
  viewing 343
chart event package 241, 242
chart events 77, 143, 145, 150, 156
chart examples plug-in 340, 650, 661
Chart Examples view 346
chart exception package 244
chart factory package 244
chart format examples 341
chart generation processes 150
chart icons 663
chart instance objects 145, 146, 148, 239
  See also chart objects
chart interactive features 147, 155, 342, 688
Chart interface 239, 312
chart item extensions 45
chart item interface 250
chart items
  See also charts
  accessing data for 326
  adding to designs 322, 678
  creating 47, 325
  defining behavior of 250
  getting 170, 321
  setting dimensions of 327
  setting properties for 325
chart label objects 160
chart labels
  applying styles to 342
  changing 160, 161, 163
  determining type 161
  formatting 162
  rendering 159, 161, 702
  setting 239
chart live preview preference 671
chart model 153, 154, 169
chart model attributes 240, 246, 247
chart model extensions 666, 684
chart model implementation classes 239, 260
chart model packages 239, 240, 245
chart model renderers 648, 685, 689
chart model schema definitions 652
chart object factory 688
chart objects
  See also charts
  accessing 148
  getting 145, 310, 321
  instantiating 239, 312, 325
  setting 146
chart package 16, 217, 239
chart programming examples 340–346
chart properties
  accessing 146
  changing 162, 169, 312, 341
  charting extensions and 652
  getting 146, 251
  rendering axes values and 162
  rendering chart blocks and 158
  report items and 325
  setting 147, 239, 251
Chart property 652
chart regions 158
  See also chart areas
chart render package 255
chart report design examples 345, 346
chart report item example 389
chart reports 43
chart script context objects 145, 148, 150
chart script package 255
chart scripting example 343
  See also chart event handlers
chart series renderers 155, 156
chart subtypes 147, 323, 648, 663
chart titles 147, 158
chart types
  adding 647, 648, 652
  getting 147
  multiple series and 240
  registering 648, 661
  setting 147, 322, 323
  specifying supported 664
chart types extension point 684
chart types plug-in 648
chart UI extensions 648, 673
chart UI plug-ins 648, 661
chart utility package 256

chart view example 346
chart viewer servlet 358
chart wizard 345, 648
  See also chart builder
chart wizard context identifier 664
chart.script package 255
Chart3DViewer application 343
chartengineapi.jar 260, 261
chartexamplescoreapi.jar 260
charting API data example plug-ins 340, 341
charting API format example plug-in 341
charting API interactivity example plug-ins 342
charting API pdf example plug-in 342
charting API preference example plug-in 342
charting API processor example plug-in 342
Charting API Reference 218
charting API script example plug-in 343
charting API viewer example plug-ins 343
charting APIs
  adding charts and 45, 312
  changing chart properties and 169
  changing charts and 321
  customizing 148
  developing with 307
  generating reports and 172
  getting chart items and 170
  Swing applications and 334
charting application examples 307, 340
charting application sample plug-in 21, 713
charting applications
  accessing BIRT APIs for 261
  adding reporting functionality to 15, 169
  changing chart objects and 239
  creating 308, 309, 327, 335
  developing 16, 169, 307
  displaying charts and 46
  handling events in 143, 320
  testing 309
charting classes 239, 312
charting engine. *See* chart engine
charting extension points 648, 650, 661, 684
charting extension schema files 652
charting extensions 647, 705, 715
  See also radar chart extension
charting interfaces. *See* interfaces
charting library. *See* chart engine
charting plug-ins 650
chartitemapi.jar 260
ChartModels class 342
ChartRuntime directory 16
charts
  See also chart elements; chart items; chart objects
  accessing 148, 310, 321
  adding data series to. *See* data series
  adding interactive features to 147, 155, 342, 688
  adding to designs 323, 327, 345, 346
  aggregating data for 649
  binding data series to 311
  binding to data sets 150, 322, 326
  building custom data sets for 241
  changing data source for 358
  clearing sections of 148
  creating 43, 239, 309, 312
  customizing 217, 307, 647
  defined 715
  defining event handlers for. *See* chart event handlers
  deploying 308
  displaying 659, 671
  editing 154, 170, 239, 321
  examining values in 159
  formatting 162, 341
  generating 45, 155, 335
  getting primary base axis for 315
  getting run-time series for 151
  getting styles for 147
  getting type 147
  handling null values for 152
  hard-coding data values for 340
  initializing data series for 150
  initializing data structures for 311
  laying out 253
  modifying data series for 159
  outlining areas in 314
  plotting sample data for 147, 322, 325, 671
  precalculating data for 691
  registering custom types for 648, 661

charts *(continued)*
rendering as images 282, 663
rendering specific series for 155, 158, 648
rendering to specific output type 241, 266, 342, 473
resizing 172
retrieving run-time values for 161
returning unique values for 648
saving changes in 4
setting display names for 659
setting properties for. *See* chart properties
setting style preferences for 342
setting unit spacing for 148
specifying type. *See* chart types
validating data for 691
Charts_FAQ.doc 16
ChartScriptContext object 145
ChartSDK directory 16
chartType element 664
charttypes extension point 684
charttypes plug-in 648
chart-viewer.jar 356, 358
ChartWithAxes interface 239, 312
ChartWithAxes property 652
ChartWithAxes type 315
ChartWithAxesImpl class 153, 239, 310, 312
ChartWithoutAxes interface 239, 312
ChartWithoutAxes property 652
ChartWithoutAxesImpl class 153, 239, 310
ChartWizardContext ID 664
ChartWizardLauncher class 345
cheat sheets 49
check boxes 681
choice definitions 54
Choice element 54
class attributes 376
Class class 115
class definitions 54, 664
class element 54, 397
class files 113, 121, 376
class hierarchy 716
class loaders 113, 150, 259, 519
class names 112, 124, 360, 716
class property 396, 447, 470, 607
class variables 716
See also instance variables; variables
classDefinition attribute 664
classes
accessing BIRT 262
accessing external 279
accessing Java 54, 112, 113
accessing report parameters and 272
associating with report elements 121
building ODA drivers and 509, 511, 513, 522, 564
building ODA user interfaces and 530, 538
building report designs and 226, 227, 230
changing chart objects and 239, 240
changing report designs and 262
connecting to web services and 184, 185
debugging event handlers and 194
defined 716
deploying 113, 183
developing with 44, 217
extending adapter 118, 124
failing to compile 124
generating reports and 257, 263, 270
handling events and 117, 118, 121
hierarchical diagrams for 218
implementing data extraction extensions and 609, 610, 614
implementing rotated label plug-in and 390, 404
instantiating 115
loading 115, 519, 574
naming conventions for 123
referencing 112, 181
registering 320
rendering CSV output and 473, 485
rendering XML output and 498
running plug-in instances and 360, 364, 368
running reports and 219, 220, 223
scripting and 112, 113
setting location of 113
setting properties for 376
setting user-supplied values and 272
classes directory 113
Classic Models sample database 17, 18, 103
See also demo database

ClassicModels directory 18
ClassLoader objects 574
  See also class loaders
CLASSPATH property 279
CLASSPATH variable 263
classpaths
  accessing JAR files and 118
  CSV ODA driver extensions and 521
  debugging and 198
  Hibernate ODA drivers and 555, 574
  Java event handler classes and 121
  JNDI resource files and 33
  plug-in extensions and 369, 371, 390
  plug-in fragments and 635
  rendering extensions and 469, 496
  report engine and 279
  rotated label extension and 393
  running reports and 118
  scripting and 112, 113, 182
  web services connections and 184
  XML data extraction extension and 605
–clean command line option 10, 13
clean-up code 101, 182
clearSections method 148
CLI library 260
close events 69, 70
close method
  Hibernate drivers and 567, 572
  output files and 103
  report designs and 306
  report documents and 271
  scripted data sets and 128, 174, 182
  scripted data sources and 127, 174, 178
  task objects and 279
closing
  connections 69, 92, 522, 572
  CSV writer 476
  data sets 70, 126
  data sources 126
  output files 476, 500
  report design files 306
  report document files 271
  scripted data sets 70, 128, 174, 182
  scripted data sources 69, 127, 174
  XML writer 500
code
  accessing data sources and 173
  accessing Java source 359
  accessing report parameters and 272
  accessing reporting functionality and 228
  accessing sample 385
  adding breakpoints to 193, 194
  adding event handlers and 63, 64, 79, 83, 117
  adding to designs 262
  building data extraction extensions and 602, 626
  building rotated label plug-in and 390
  changing 349, 356
  changing connection properties and 99
  charting API examples and 340
  charting extensions and 650, 652
  checking for errors in 104
  compiling 350, 380
  controlling report creation and 63
  creating custom designers and 227
  debugging plug-ins and 390
  defining executable 60
  deploying applications and 262
  developing applications and 217
  developing ODA driver extensions and 511, 513, 555
  developing plug-in fragments and 636
  developing rendering extensions and 466
  downloading source 350, 353
  extending BIRT functionality and 52
  getting parameter values and 275
  grouping data and 79
  importing 5, 350
  installing 21–22
  loading 362, 746
  logging information and 269, 270
  opening design files and 270
  retrieving data and 70
  running plug-ins and 362, 381, 390
  setting up workspaces for 354
  tracking script execution and 101, 102
  validating 189, 192
  viewing charts and 255, 358
code archives 21, 353
code editors 29
code modules 228
Code page. *See* script editor
code points 716

codec library 260
codec.jar 260
collectCustomProperties method
  CSVFilePropertyPage 541
  CSVFileSelectionPageHelper 542
  CSVFileSelectionWizardPage 543
  HibernateDataSourceWizard 588
  HibernatePageHelper 585, 587
  HibernatePropertyPage 588
collectDataSetDesign method 593, 594
collections 273, 274, 275
color palette 228
color settings 110, 160, 165, 316
ColorDefinitionImpl class 160
column aliases 96, 97, 133
  See also column names
column binding 75, 90, 150, 305, 456, 717
column binding definitions 49
column binding expressions 458, 460
column headings 133
  See also column names
column keys 717
  See also cross-tab reports
column names 96, 97, 133, 528, 545
  See also column headings
Column type attribute 612
column types 97, 133, 619
columnar layouts 59
columnBindingsIterator method 305
columnNumbers variable 476
columns
  See also fields; computed fields
  accessing 96, 304
  adding to tables 179
  concatenating values in 75
  defined 717
  exporting data and 612, 617
  getting display values for 135
  getting information about 97, 133
  getting number of 97, 133
  getting values in 527
  iterating through 97
  querying CSV data sources and 528
  rendering CSV output and 473, 476, 477
  retrieving values in 96
Columns element 612
combination charts 658, 717
comma (,) character 476, 478
comma-separated values files. *See* CSV files
comma-separated values ODA drivers. *See* CSV ODA driver extension
comma-separated values rendering extension. *See* CSV report rendering extension
comma tags 479
command line applications 19, 219
command line arguments 20
command line library 260
command stack 228
comments 120
comments property 448
commit method 567
commit operations 567
common.jar 260
CommonConstant class 523, 529, 564
CommonDataExtractionImpl class 610
commons-cli.jar 260
communications protocol 758
compiler preferences 351
compiler settings 350, 351
compiling 124, 350, 379, 380
complex properties 294, 297
component diagrams (BIRT) 39, 40
component libraries 44, 48, 229
component package 240, 250
component palettes. *See* Palette view
component types (class definitions) 54
component.xsd 652
components
  See also report elements; report items
  accessing 228, 229
  adding report items and 46, 51, 59
  saving 44
compute method 155, 686, 691
computed columns. *See* computed fields
computed data. *See* computed values
computed fields 97, 133, 178, 718
computed values 178, 718
  See also calculations
ComputedColumnHandle type 305
concatenation 75, 320
Concurrent Versions System repository. *See* CVS repository
conditional expressions. *See* Boolean expressions
conditional formats 718

See also formats
cone charts 310, 323
See also charts
configuration files 29, 586
configuration information 87, 191
configuration manager (updates) 381, 382
configuration names (debugger) 196
configuration objects 220
configuration variables 228
configuring
BIRT home property 263
connection pooling services 32, 33
CSV ODA drivers 518
design engine 227, 265
Eclipse workspace 350, 351
Hibernate drivers 565, 574, 577, 585
caution for 555
HTML emitters 265, 266
image handlers 266
ODA data source connections 510
report emitters 264
report engine 220, 221, 258, 263, 264
report viewer 28, 29
connect method 184, 185
Connection class
Hibernate ODA drivers and 564, 565
ODA CSV drivers and 522, 523
web services data sources and 184, 185
connection constants 540
connection exceptions 185
Connection objects 565, 719
See also connections
connection pooling 32
connection profile extension. *See* connectionProfile extension point
connection profile properties 539
connection profiles 510, 520, 535
connection properties 99, 125, 184
connection properties page 511
connectionClass property 184
connectionProfile extension point 510, 520, 535, 536, 578
connectionProfile.exsd 511
connectionProfileImage extension point 535
connectionProperties parameter 184
connections
application servers and 32, 33
closing 69, 92
CSV data sources and 522
external data sources and 46, 60
getting 529, 565
handling events for 69
Hibernate data sources and 564, 565, 566, 572
JDBC data sources and 32, 33, 99
JNDI service providers and 32, 33
opening 69
report engine and 45, 258
setting properties for. *See* connection properties
web services data sources and 184, 185
Console view 194
constants 112, 719
Constants class 539, 540
constructor code 719
constructSessionFactory method 566, 574, 575
container items 59
containers 719
containment 719
containment hierarchy 719
content. *See* structured content
content emitters. *See* emitters
content objects
applications 184
content plug-in (report engine) 466
ContentEmitterAdapter class 473
See also adapter classes
ContentEmitterVisitor objects 474, 498
context class loaders 519
context IDs 664
context mapping 28, 29, 91
context menus 443, 445
context objects
application 91, 92, 184
chart engine and external 149
chart event handlers and 145, 148, 150
connections 184
deployment 264, 266
external content 146, 149, 280
report parameters 100
reports 86, 88, 89, 131
context parameters 29
context root 28
context-param element 29
contributors 217, 349

converters 720
  See also specific type
copyInstance method 660, 661
core API library 260
core plug-in 650
coreapi.jar 260, 261
counters 104, 105
Create Ant Build File command 380
create method 239, 240, 659
create_classicmodels.sql 18
createAndInitCustomControl method 540, 585, 587, 588
createCustomControl method 541, 585
createDesignEngine method 291
createFactoryObject method 258, 267, 291
createGetParameterDefinitionTask method 272, 273
createLegend method 688
createPageControl method 543, 588
createPageCustomControl method 543, 585, 587, 588
createRenderTask method 279
createReportEngine method 267
createRotatedText method 406
createRunAndRenderTask method 278
createRunTask method 278
createSampleRuntimeSeries method 148
createSeries method 688
createSeriesDataPoint method 688
createTabItems method 449
creating
  BIRT projects 349, 354–356
  charting applications 308, 309, 327, 335
  charting extensions 647, 652
  charts 43, 239, 309, 312
  Classic Models database 18
  connection profiles 536
  context menus 443
  data extraction extensions 602
  debug configurations 197
  design engine 227
  Eclipse projects xix
  event handler classes 117, 118
  event handlers 63, 64, 83, 117
    tutorial for 103–107
  Java applications 49, 743
  lists 59
  master pages 88, 283
  ODA driver extensions 509, 510, 511, 554
  ODA drivers 46, 50
  ODA user interfaces 385, 510, 530, 537
  plug-in extensions 370–377
  plug-in fragment projects 637–639
  plug-ins 360, 366
  property editors 445
  property structure objects 299
  queries. *See* queries
  report designer 228
  report designs 48, 226, 290, 306
  report elements 227, 302
  report engine 220, 221, 258, 263, 267
  report item builder 439
  report item extensions 387, 399, 416
  report items 47, 302, 387
  report rendering extensions 466, 494
  reporting applications 258, 259, 262
  reports 63, 217, 258, 279
  resource bundles 635
  scripted data sets 173, 176
  scripted data sources 173, 174, 175
  stand-alone applications 49
  tables 59
  update sites 382–383
criteria. *See* parameters
cross tabs 110, 720
cross tabulation. *See* cross tabs; cross-tab reports
cross-tab elements 720
cross-tab reports 720
cross-tab scripting example 109
crosstabcoreapi.jar 260
CSS files 52, 260, 261, 713
CSV data sources 488, 509, 517
CSV data structures 512
CSV emitter 474, 480, 486
  See also CSV report rendering extension
CSV emitter logging levels 475
csv extension point 512
CSV files
  accessing data in 509, 511
  building queries for 99
  closing 476
  connecting to 522
  defined 717
  defining content of 474

defining structure of 473
downloading sample plug-in for 385
extracting data for 601
initializing output streams for 474
opening 475
preparing queries for 524
preparing result sets for 527
reading from 384, 522
setting logging levels for 475
viewing content of 486
writing to 466, 473
CSV formats 465, 466, 601
CSV ODA driver extension
adding data access functionality to 517–520
building projects for 513–516
compiling and debugging 513
developing 511, 512, 522
downloading source code for 511
setting dependencies for 521
CSV ODA driver extension examples 384, 511
CSV ODA driver plug-ins 512, 513
CSV ODA drivers 384, 512, 518, 522
CSV ODA UI extensions 512, 530, 534, 537
CSV ODA UI plug-in 512, 535, 538, 548
CSV ODA UI plug-in projects 511, 531–534
csv package 473
csv plug-in 465, 482
CSV rendering option 479
CSV report rendering extension
building 482
creating projects for 466, 483
developing 466, 473
downloading source code for 385, 466
generating output and 486
overview 465, 473
running 480, 485
setting dependencies for 469
setting encoding scheme for 475
specifying extension points for 470, 471
testing 480, 482, 488
viewing output for 481, 486
CSV report rendering plug-in 385
csv ui extension 512
csv ui wizards extension 539
CSV writer 473, 474, 475, 476
CSVBufferReader class 522, 526
CSVFilePropertyPage class 539, 540
CSVFileSelectionPageHelper class 539, 541
CSVFileSelectionWizardPage class 539, 543
CSVPlugin class 473
CSVRenderOption class 473, 479
CSVReportEmitter class 473, 474, 476
CSVReportEmitter method 474
CSVTags class 473, 479
csvTest.reportdesign 488
CSVWriter class 473, 479
cubes 721
See also multidimensional data
currentSession method 576
cursors 527
CurveFittingViewer application 343
custom chart generator 217
custom data sources 510
See also ODA data sources
custom desktop reporting application 219
custom emitters 45
custom Java classes 182
custom report design tools 227
custom report designer 49, 53
custom report generators 49, 217
custom status handlers 265
CustomDataSetWizardPage class 538
customizing
BIRT framework 50, 349, 359
chart builder 648
chart types 661
charting API 148
charts 217, 307, 647
connections 184
data sets 241
event handlers 118
ODA drivers 46, 50
output formats 278, 388, 465, 602
property pages 446, 448, 530
report designs 217
report emitters 50
report items 47, 50, 387, 389, 416
reporting applications 49, 359
reports 85, 262
user interfaces 274

CVS repository 359, 390

D

data
  accessing 222, 456
  aggregating 46, 68, 319
  defined 721
  displaying 276
  exporting 466, 601, 612
  extracting. *See* data extraction
  filtering 46, 49, 89, 276
  generating sample 147, 322, 325, 671
  grouping 46, 49, 79, 151
  hard-coding 340
  precalculating 691
  retrieving 43, 46, 173, 181, 276, 509
  sharing with external applications 494, 602
  sorting 46, 89, 151, 311
  validating 691
data access services 46
data adapter API library 260
data adapters. *See* adapter classes
data binding 75, 90, 150, 305, 456, 717
data binding definitions 49
data binding expressions 458, 460
data binding phase (charts) 150
data chart examples 340
data charts. *See* charts
data charts viewer 340
data components. *See* data elements
data cubes 721
  See also multidimensional data
data drivers. *See* drivers
Data element 612
data elements 57, 60, 721
  See also data
data engine 45, 46, 510
data engine extensions 46
Data Explorer 721
data extension names 134
data extraction 466, 622
data extraction extension
  creating projects for 603–605
  developing 601, 602, 609
  downloading source code for 602
  implementing 610
  overview 601, 609
  running 620
  setting dependencies for 605
  testing 620, 626
data extraction extension points 607
data extraction interfaces 222, 609
data extraction library 260
data extraction plug-ins 601, 602, 609, 610
data extraction schema definitions 602
data filters 735
  See also filtering data
data package 240, 251, 340
data point 722
data point definitions 649
data point events 159
data point hints 159, 160
data point labels 159, 160, 702
data points
  See also charts
  comparing 647
  displaying series values and 310, 311
  rendering 155, 159, 160
  validating 691
data rows 722
  See also rows
data series
  See also charts
  adding 316
  associating data with 318
  binding to charts 311
  binding to data sets 150
  building run-time instances of 148, 151, 152, 156, 311
  calculating values for 693
  changing 159
  comparing multiple values in 647
  creating 315, 316, 658
  defined 310, 766
  determining type 152
  getting run-time 151
  grouping 341, 688
  initializing 150
  labeling data points in 702
  nesting 154
  preserving changes to 670
  providing hints for 159, 160
  rendering 155, 158, 648
  retrieving run-time 161
  returning unique values for 648
  scaling 161
  setting bar size for 148

setting type 240, 253, 653
setting unit spacing for 148
stacking 658
validating data points for 691
z-ordering 688
data series composite extension points 673
data series definition objects 151, 311, 316, 319
data series identifier 152, 159
data series objects 152
data series palette 684
data series renderers 155, 156, 158
data series sample plug-ins 341
data series types 316, 323, 653, 654
data services 46, 345
data set adapters 126
data set classes 304
data set definitions 58
data set editor page 530, 537
data set elements 58, 60, 126
See also data sets
data set events 68, 69, 70, 73, 74
data set extension properties 134
data set extensions. *See* ODA data set extensions
data set handles 303, 304
data set instance interface 133
data set objects 133
See also data sets
data set page (ODA UI) 512, 537, 555, 585
data set parameters 101, 185, 722
See also parameters; report parameters
data set processors (charts) 151, 648, 649, 685
data set types 134
data set UI extension 537
data set wizard page 539, 543, 588
data set wizards 537, 538
data sets
See also data set elements; data set objects
accessing 133, 303
binding charts to 150, 322, 326
binding report items to 75, 76, 90
binding to report items 305, 456, 458
building queries for 98, 134, 555
caching 74
changing data sources for 304
changing properties for 98, 304
changing queries for 98
closing 70, 126
creating 173
customizing 241
defined 722
developing data extraction extension for 601, 602, 609
developing ODA extensions for 46, 523, 530, 537
editing 510, 537
fetching 134
getting column information for 96, 97, 133, 134
getting data sources for 134
getting names 134
getting number of columns in 133
getting properties for 304
getting type 134
handling events for 68, 73, 74, 126, 133
iterating through columns in 97
opening 69, 126
replacing null values in 152
retrieving from JDBC data sources 32
retrieving from web services 184
retrieving values in 276
reusing 49
scripting for. *See* scripted data sets
setting properties for 58, 304
sharing 68
viewing data bindings for 75
data source adapters 125
data source classes 303
data source drivers. *See* drivers
data source editor page 530, 538
data source elements 58, 60, 125, 723
See also data sources
data source events 68, 69, 70, 73, 74
data source extension elements 518, 537
data source extensions 509, 517, 530
data source handles 230, 303
data source objects 99, 134
See also data sources
data source page (ODA UI) 512, 555, 585
data source UI extension 537
data source wizard page (ODA UI) 539, 543
data source wizards (ODA) 538

data sources
  See also specific data source type
  accessing 303, 509, 510, 517
  adding 530
  changing 304, 358
  changing data set 304
  changing properties for 98
  changing queries for 98, 99
  closing 126
  connecting to. *See* connections
  creating 173
  defined 722
  defining external 60
  developing ODA extensions for 46, 510, 511, 537, 554
  developing reporting applications and 258
  editing charts and 4
  extending functionality of 509
  getting 134
  handling events for 68, 73, 74, 80, 125
  loading JDBC drivers for 261
  opening 125
  retrieving data from 46, 173, 181, 509
  reusing 49
  scripting for. *See* scripted data sources
  selecting ODA 510, 538
  unsupported 46
data structures 57, 297
Data Tools Platform (BIRT) 4
  See also DTP Integration package
Data Tools Platform (Eclipse) 46, 509
data transform services 46
data type mappings 520
data types
  charting extensions and 652, 675
  CSV files and 523
  data series run-time values and 150
  defined 723
  expressions and 456, 457
  extracting 619
  getting native 97
  Hibernate data sources and 563
  inspecting 195
  plug-in extension points and 360
  report parameters and 274
  ROM metadata structures and 57
data.xsd 652
dataadapterapi.jar 260
dataaggregationapi.jar 260
database classes 512
database drivers. *See* JDBC drivers
database management systems 724
database schemas 764
  See also schemas
databases 509, 510, 723
  See also Classic Models sample database; data sources
DataCharts charting example 340
DataChartsViewer application 340
DataException exception 195
dataExtraction package 602
  See also data extraction extension
dataExtraction plug-in 601, 603, 609, 610
dataExtraction.exsd 602
dataextraction.jar 260
DataExtractionExtensionBase interface 609
datafeed package 240, 241
datapointdefinitions plug-in 649
DataPointHints objects 159
DataPointHints property 160
DataSet element 58, 126
dataSet extension element 519
DataSet interface 150, 310
DataSetAdapter class 126
  See also adapter classes
DataSetHandle class 303, 304
DataSetImpl class 318
DataSetMetaData class
  csv package 523, 528
  hibernate package 564, 568
dataSetPage element 537
DataSetProcessorImpl class 151
datasetprocessors extension point 685
datasetprocessors plug-in 648, 649
dataSetUI extension 537
dataSetUI page wizard 585
dataSetWizard element 537
DataSetWizard extension point 537
DataSetWizardPage class 539, 588
DataSetWizardPage extension point 537
DataSource element 58, 125
dataSource extension element 518, 519
dataSource extension point 509, 510, 517
DataSource object 99, 134
  See also data sources
dataSource.exsd 510

DataSourceAdapter class 125
  See also adapter classes
DataSourceEditorPage class 530, 538
DataSourceHandle class 230, 303
dataSourceUI extension 537
DataSourceWizardPage class 539
dataTypeMapping elements 563
dataTypeMapping type 520
DataTypes class 564
Date data type 724
date parameters 274
date values 612, 675, 724, 773
DATE_TIME_LITERAL value 161
date-and-time data type 724
DateTimeDataSet interface 310
DBMS (defined) 724
Debug Configuration dialog 197
Debug Configurations command 196, 197
debug extension point 188
debug launch configuration interface 196
Debug perspective 190, 191, 192
debug targets 191, 192
debug types 197
Debug view 191, 192
debugger 187, 188, 190, 196
debugger interface. *See* Debug perspective
debugging
  applications 187, 197, 269
  charting extensions 705
  defined 724
  event handlers 188, 190, 194, 198
  Java event handlers 142
  ODA driver extensions 513
  plug-ins 390
  report engine 188
  report items 194
  reports 187, 197, 198–214
  scripts 193
debugging configurations 196, 197
debugging processes 192, 195
debugging tasks 188
debugging tutorial 198–214
declarations 724
declarations section 724
default locale 20
default output formats 257
default values 66, 228, 272, 274
DEFAULT_MAX_ROWS constant 540
DefaultChartTypeImpl class 662
DefaultDataServiceProviderImpl class 345
defaultDisplayName attribute 519
defaultDisplayName property 519
DefaultSeriesUIProvider class 674
defaultStyle property 396
defaultValue attribute 519
defaultValue property 397
DeferredCache class 689
DeferredCacheManager class 688
deleteGlobalVariable method 86, 131
deletePersistentGlobalVariable method 86, 131
deleting
  BIRT program files 12
  cached information 10
  global variables 86
  temporary files 220
demo database 18, 711
  See also Classic Models sample database
Demo Database archive 18
Demo Database package 5, 17
Dependencies page (PDE Editor) 369, 370
deploy 725
deploying
  applications 26, 262, 266, 289
  charts 308
  fragment package 637
  Hibernate ODA UI plug-in 594
  Java classes 113, 183
  JDBC drivers 92
  plug-in extensions 381–384
  plug-in fragments 643
  plug-ins 360, 382
  report engine 259
  report item extensions 390, 412
  report viewer 4, 25, 29
  reports 25, 118, 183, 282
DeploymentRuntime directory 16
Derby databases 17, 18
derived classes. *See* descendant classes
descendant classes 725
description element 396
design configuration objects 227, 291
design definitions 58, 59

design element API classes 260
design element handles 230, 232
design elements 128, 271
design engine 43, 44, 227, 290
design engine API 226, 227, 257, 262
    See also report model API
design engine API documentation 257
design engine API package 217, 230
design engine classes 226, 227, 230, 291
design engine configurations 227, 265
design engine extensions 47
design engine interface 238
design engine objects 291
    See also design engine
design engine plug-ins 227, 228
design engine tasks 227
design environments. *See* BIRT; Eclipse
design events 67
design files
    creating reports and 219, 257, 258, 270, 286
    defined 759
    designing reports and 48
    generating 43, 44, 49, 227
    loading 221
    opening 258, 270, 291
    renaming 327
    running 222, 279
    saving 227
    setting location of 29
    validating 44, 53
design handles
    accessing properties and 229, 289
    accessing report items and 292
    creating report elements and 227
    getting 87, 88, 89
    opening report designs and 228, 291, 292
design interfaces 129
design model 47, 51
design model objects 226
design objects 228
    See also designs
design perspective 7, 8
design properties 59
design time 725
design tools 3, 43, 226, 227, 228
design.xsd 52, 53
DesignChoiceConstants interface 294, 297
DesignConfig class 227
DesignConfig objects 227, 291
DesignElement element 58
    See also design elements
DesignElementHandle class 230, 232, 296
DesignEngine class 227, 291
DesignEngine objects 227, 291
designer applications 3
designer package 4
designer UI extensions. *See* user interface extensions
designers 43, 49, 219
    See also BIRT Report Designer; BIRT RCP Report Designer
designing reports 43
    See also designs
designs
    See also page layouts
    accessing 228, 289, 291
    accessing parameters in 273
    accessing properties for 229
    accessing ROM schemas for 53
    accessing structured content in 257
    adding charts to 323, 327, 345, 346
    adding master pages to 88
    adding report items to 292, 302
    changing 68, 87, 89, 262, 289
    creating 226, 290, 306
    customizing 217
    defined 725
    deploying to web servers 28
    generating XML files and 626
    getting properties for 270
    initializing 102
    installing report viewer and 29
    retrieving external data for 510, 522
    reusing 43, 48
    saving 290, 306, 327
    selecting ODA data sets and 538
    selecting report items for 46
    standardizing 48
    testing for parameters in 273
    validating 53, 57
    viewing CSV output in 488
    viewing report items in 73
    viewing XML output in 507
    writing event handlers for 67, 128, 135

desktop applications 219
  See also Java applets
destroy method 220
detail reports. *See* subreports
detail rows 59, 75, 77
detailType property 397
developers xxi
developing
  chart event handlers 143, 164, 167, 320
  charting applications. *See* charting applications
  charting extensions 647, 652
  data extraction extensions 601, 602, 609
  design tools 226
  ODA extensions 385, 509, 510, 511, 554
  ODA user interfaces 510, 530, 537, 577, 584
  plug-ins 360, 366
  rendering extensions 465, 466, 494
  report engine 219
  report item extensions 387, 389, 416
  reporting applications 217, 258, 259
  reports 49
  rotated label extension 390, 404–412
development environments 64, 259
development languages 25, 742
  See also scripting languages
device package 240, 241
device renderers 155, 649, 687
DeviceAdapter class 242, 649
  See also adapter classes
devicerenderers extension point 686
devicerenderers plug-in 649
DHTML (defined) 727
dial charts 307, 652
  See also charts
DialChart interface 239, 312
DialChart property 652
DialChartImp class 239
DialChartImpl class 310, 312
DialChartViewer application 344
difference charts 310
  See also charts
DifferenceDataSet interface 310
Dimension attribute (charts) 146, 147, 148
dimension property 312, 313
dimensions 725
  See also data cubes
directories
  accessing JAR files and 118, 228
  accessing Java classes and 113
  accessing report designs and 28
  building charting API examples and 340
  building charting extensions and 652, 663
  developing reporting applications and 259
  displaying reports and 26
  extracting BIRT API Javadocs to 261
  extracting chart engine package and 16
  extracting source code to 354
  getting location of plugins 265
  installing language packs and 11
  installing plug-ins and 359, 360, 362
  installing report viewer and 27, 28
  installing updates and 382
  referencing BIRT home 263
  referencing Eclipse home 358
  saving event handlers and 113
  saving image files and 266
  saving temporary files and 265
  unpacking program archives and 9
  updating designers and 12
  viewing PDF documents and 31
directory paths. *See* paths
disconnect method 184, 185
disk writes 219
display adapter classes 241, 649
  See also adapter classes
display names 519, 520, 659, 725
  See also DisplayName property
display server 155
DisplayAdapter class 649
displaying
  breakpoints 192, 194
  cascading report parameters 277
  charts 659, 671
  CSV output 481, 486
  data 276
  data bindings 75
  error messages 104, 107
  event handlers 64, 83
  extension point schemas 374
  HTML output 32, 44
  images 429

displaying *(continued)*
  manifest header settings 369
  PDF documents 31, 32, 288
  plug-in project settings 369
  property annotations 376
  report items 46, 73
  reports 28, 31, 32, 43, 219
  scripts 193
  update site content 383
  XML output 507, 609
DisplayName property 397
displayNameID property 396, 397
displayservers plug-in 649
displayText element 396
distinct values 648
distributing reports. *See* deploying
DOC emitter 45
document files
  accessing contents 222, 271
  accessing structures 271
  building 44
  closing 271
  creating 80, 257, 278
  defined 760
  exporting data from 601
  generating reports from 65
  opening 221, 258, 270, 271
  overview 48
  rendering specific pages in 271
  writing to disk 222
document model (BIRT). *See* ROM
document object model. *See* DOM
document objects 258, 270, 271
document type definitions 726
documentation xxii, 16, 359
  See also Javadocs
documentation attribute 360
documents 280, 288
  See also document files; reports
DOM (defined) 726
dom package 614
domain names 726
Double class 726, 735
double data type 727
double quotation mark (") character
  command line arguments and 20
  expressions and 456
  JavaScript code and 320
double-precision floating point numbers 726, 727, 735
download messages 609
downloading
  BIRT Full Eclipse Install 7
  BIRT RCP Report Designer 8
  BIRT Report Designer packages 15
  builds 5
  chart engine archive 16
  chart engine plug-in 652
  charting plug-ins 650
  demo database archive 18
  extension examples 384
  JAR files 118
  JDK software 7
  language packs 11, 635
  ODA driver sample plug-ins 511, 513
  report engine archive 19
  Samples archive 21
  source code 21, 350, 353, 390
  Web Tools Integration archive 22
DPI resolution (charts) 155
drawCircle method 687
drawing framework (charts) 686, 687
drawing primitives (charts) 155, 649, 686
drawLine method 687, 700
drawPolygon method 700
drawRectangle method 687
drawSeriesLabel method 702
Driver class 522, 523
driver classes 261
driver constants (ODA) 529
driver manager 9
driverClass property 518
drivers
  See also specific type
  accessing external sources and 510
  adding connection profiles for 510, 520, 535
  changing connection properties for 511
  connecting to JDBC data sources and 33
  connecting to web services and 184, 185
  creating ODA 46, 50, 554
  defined 727
  defining error messages for 523
  deploying 92

designing reports and 32
developing applications and 261
developing ODA 509, 510, 511
downloading sample plug-ins for 384
initializing connections for 522
installing report viewer and 26, 27, 28
locating JNDI resource files for 33
registering 574
running 522
setting location of 92
testing 511
using nonstandard 26
drivers directory 27, 261, 574
DTD (defined) 726
dteapi.jar 260, 261
DTP Integration package 17
DTP ODA API 509, 510, 522
DTP ODA API Reference 511
DTP ODA extensions. *See* ODA extensions
DTP ODA framework 46, 510, 538
DTP ODA interfaces 509, 519, 522
DTP ODA wizards 509
dynamic data 151, 311
Dynamic HyperText Markup Language (DHTML) 727
dynamic parameters 100
dynamic text elements 727
dynamic variables 727

E

-e command line option 19
e.reports. *See* reports
ebXML formats 494
Eclipse 10, 11
Eclipse applications 264
Eclipse Babel Project 635
Eclipse BIRT packages 4, 15, 217
Eclipse compiler 350
Eclipse Data Tools Platform 46, 509
See also BIRT Data Tools Platform
Eclipse debug extension point 188
Eclipse environments 64, 727
Eclipse Graphical Editing Framework 4
Eclipse home directory 358
Eclipse IDE 650
Eclipse Java Development Tools interface 194
Eclipse launcher 728
Eclipse Modeling Framework 4, 654, 728
See also EMF libraries
Eclipse perspective 728
Eclipse platform 40, 728
See also Eclipse Rich Client Platform
Eclipse Plug-in Development Environment 360, 366, 729
See also PDE Workbench
Eclipse plug-ins 359
Eclipse projects xix, 39, 368, 729
See also projects
Eclipse Rich Client Platform 729
See also rich client platforms; BIRT RCP Report Designer
Eclipse SDK 3, 351, 352
Eclipse Software Development Kit. *See* Eclipse SDK
Eclipse Update Manager 12
Eclipse views 369, 730
Eclipse Web Tools Platform 4
Eclipse Workbench 380, 413, 730
Eclipse Workbench projects 368
Eclipse workspace 350, 354, 730
ECMAScript implementation 83
ecore.jar 260
EDI standards 494
editor controls 539
editor element 396
editor pages 421, 537, 538
EJBs 173, 731
Electronic Business extensible markup language 494
Electronic Data Interchange standards 494
element adapters extension point 388, 395, 398
element adapters schema definition 388
element adapters. *See* property pages
element attribute 360
element definitions (ROM) 55
See also ROM elements
Element element 55
element interface 118
elementAdapters extension point 388, 395, 398, 446
elementAdapters.exsd 388
ElementDetailHandle class 236
ElementFactory class 302
ElementFactory objects 227, 323

elements
  See also report elements; ROM elements
  class attributes and 376
  creating report extensions and 399
  defined 730
  running plug-in extensions and 360, 363, 373
elements package 233, 235
ellipsis (...) button 731
embeddable HTML output 281
embedded report engine 219
EMF (defined) 728
EMF Generator Model project 654
EMF libraries 260
EMF model 652, 654
EMF software 4
emitter csv package 473
emitter csv plug-in 465, 482
emitter extension IDs 470
emitter interfaces 473, 497
emitter logging levels 475, 476, 499, 500
emitter objects 474, 498
emitter package 466
emitter plug-in 466, 497
emitter xml plug-in 465, 498
emitterconfig.jar 260
emitters
  adding 465
  configuring 264, 265
  creating 474, 498
  customizing 50
  extending functionality of 385, 466
  generating CSV output and 470, 473, 486
  generating reports and 45, 80, 266
  generating XML output and 497, 498
  listed 42
emitters extension points 388, 470, 497
emitters plug-in 465, 473
emitters.exsd 388, 466
enableLineSettings method 684
enablement elements 448
encapsulation 731
encoding 773
Encoding element 611
Encoding property 614
encoding variable 614
end method 476, 500
endCell method 478
endRow method 478
engine API library 260
engine api package 217, 223, 225
engine APIs. *See* chart engine API; design engine API; report engine API
engine configuration objects. *See* EngineConfig objects
engine constants 279
engine execution phases 86
engine plug-in 652
engine processing phases 65, 71, 80, 86
engine task processes 64–65, 84
engine task types 197
engine variable 284
engineapi.jar 260, 261
EngineConfig class 92, 220, 264
EngineConfig objects 220, 263, 264, 265, 266
EngineConstants class 279
EngineException exceptions 286
engines 40, 44
  See also specific reporting engine
EngineTask class 92
enterprise 731
enterprise environments 731
Enterprise JavaBeans. *See* EJBs
enterprise reporting 731
enumeration classes 239, 240
environment settings 32
environments 7, 219, 265, 755
error messages 104, 107, 189, 523
errors 9, 104, 187, 188, 356
  See also exceptions
evaluate method 86
event handler examples 107, 135, 343
event handler interfaces 117, 123, 124, 128
event handlers
  See also events
  accessing application context and 91
  accessing JAR files for 117
  accessing methods for 114
  adding logging code to 79
  adding page specific content and 85
  adding to designs 64
  associating context objects with 86, 131

building charts and. *See* chart event handlers
building data sets and 68, 126, 127, 174
building data sources and 125, 126, 174
changing column bindings and 75
changing connection properties and 99
changing designs and 89, 128
changing styles and 88, 90
closing connections and 92
creating 63, 64, 83, 117
tutorial for 103–107
customizing 118
debugging 142, 188, 190, 194, 198
defined 52, 732
displaying 64, 83
dropping report elements and 90
fetching data and 70
generating report elements and 93, 117, 123
generating report items and 125
getting column information for 96, 97
getting report parameters and 67, 100
grouping data and 79
nested report elements and 73
page-level scripts and 77
previewing 96
rendering report items and 52
retrieving external content and 149
running queries and 98, 100
running reports and 93
saving 113
selecting 83
setting data set parameters and 101
sharing data and 140
tracking execution sequence for 101
type casting and 152
validating report parameters and 67
verifying execution sequence of 79
event listeners 732
event object cache 687
event order sequence 71–81
event package 240, 241, 242
event types 66
eventhandlers.jar 198
EventObjectCache class 687
events
See also event handlers
accessing data sets and 68, 69, 70, 73, 74
accessing data sources and 69, 70, 73, 74
adding report items and 71, 73, 77
caching data and 69, 74
creating charts and 77, 143, 145, 150, 156
customizing reports and 262
defined 732
designing reports and 67, 71, 128, 135
displaying data and 80
firing 52, 64, 68, 71, 74
generating reports and 73, 74, 80
grouping data and 78, 79
iterating over data rows and 77
iterating over multiple cells and 78
running reports and 76, 86
scripting for 63, 64, 83, 103, 117, 123
selecting parameters and 66, 72
subscribing to 365
example charting API plug-ins 340–346
example charting applications 16, 307, 340
example database 18
See also Classic Models sample database
example event handlers 107, 135, 343
example ODA extensions 509, 511
example plug-in extensions 384
example plug-in fragment 385, 636, 640
examples core plug-in 650
examples plug-in 340, 650, 661
Excel converter 732, 756
Excel spreadsheets 284
EXCELRenderOption class 281, 284
exception breakpoints 194
exception classes 194
exception package 240, 244
exceptions 185, 194, 296, 523, 732
executable files 732
ExecuteCSVReport application 485
ExecuteCSVReport class 485, 486
ExecuteCSVReport project 484, 488
executeQuery method 184, 185, 525, 571
ExecuteXMLReport application 507
executing reports 64, 86, 93, 118, 465
execution errors 194
execution mode parameter 19

execution phases 86
execution processes 86
execution requests 93
experts. *See* wizards
exploding pie slices 153
Export data types property 615
Export Data user interface 601, 611, 622
export destination options 381
Export dialog 381
export options 382
Export Wizard (Eclipse PDE) 381
exporting
  data 466, 601, 612
  plug-in extensions 390
  plug-in fragments 637
  plug-ins 381
exportIntoString method 620
expression builder 188, 458, 460, 733
expressions
  binding data and 75, 76, 458, 460
  checking syntax of 188
  concatenating values and 75
  converting strings to 456, 457
  defined 733
  evaluating JavaScript 86
  manipulating numeric values with 751
  manipulating string data with 768
  matching text patterns and 759
  returning Boolean values from 710
  validating 188, 193
Expressions view 193
.exsd files 360, 373
  See also extension point schema definitions
extended-item name element 53
ExtendedItemHandle class 310, 398, 446
ExtendedItemHandle objects 322, 325, 326
extendsFrom property 396
Extensible Markup Language. *See* XML
extension APIs 218
extension element 363
extension examples 384
extension names 134, 360, 417
extension package (report engine) 389
extension point schema definitions
  connection profiles 511
  data extraction plug-in 602
  defined 359
  displaying as HTML 374
  displaying property annotations in 376
  ODA driver extensions 510
  ODA driver sample plug-ins 564
  ODA UI extensions 510
  opening 373, 374
  plug-in extensions 360, 373
  property pages 511
  report engine emitters 466
  report item extensions 360, 387, 389, 399
  rotated label sample plug-in 395
extension points
  accessing external data sources and 510
  accessing Hibernate data sources and 577, 580
  adding chart types and 648, 650, 661, 684
  adding ODA drivers and 46
  building ODA user interface and 534
  building rotated label sample plug-in and 395, 398, 399
  compiling source code for 351
  customizing BIRT framework and 50, 359
  customizing property pages and 446
  customizing report items and 387, 417
  debugging and 188
  declaring 369, 375
  defined 734
  extracting data and 602, 607
  finding 373
  installing Eclipse SDK for 352
  rendering charts and 155
  rendering output and 466, 470, 497
  running plug-ins and 362
  specifying 363
  viewing attributes of 374
Extension Points page (PDE Editor) 369, 372
extensionName property 396, 417
extensionProperties array 99
extensions
  adding chart items and 47
  adding emitters and 465
  adding report items and 387, 389, 416

adding reporting functionality and 359
building plug-in 377–380
changing MIME types for 609
charting data and 647, 652, 658
creating 53, 370, 373, 375
customizing BIRT framework and 50, 349, 359
customizing report items and 47
debugging and 194
declaring 369
defined 734
deploying 381
developing ODA 46, 509, 510, 511, 554
downloading sample plug-ins for 384
extracting data and 601
implementing 359
rendering reports and 465
setting class attributes for 376
specifying 372
testing 380
unloading 220
viewing property annotations for 376
Extensions page (PDE Editor) 369, 372, 373, 375
external classes 280
external context objects 146, 149
external data sources 60, 510, 517
external Java classes 279
external JavaScript files 111
external libraries 112
external objects 280
external values 272

F

-F command line option 19
-f command line option 19
factory API classes 260
factory method 222
factory package 240, 244
factory processes 86
factory property 398, 447
factory service 267, 291
features 5, 6, 383
fetch events 70
fetch method
retrieving data rows and 174, 178, 182
writing event handlers and 127
field names. *See* column names
field variables 734
See also member variables; variables
fields 304, 734
See also columns; computed fields
Figure UI provider 421, 435
File class 524
file names 20, 258, 269, 285
file objects 524
file output streams 257, 474, 499, 609
file systems 27, 264, 265, 266
file types 47, 52, 735
FileDialog method 586
files
See also specific type
accessing resource 264
changing source code and 356
creating 101
creating reports and 43
downloading designer 7, 8
extracting program archive 8, 9
generating output and 257
generating temporary 221
mapping to report viewer 29
overview 47–49
referencing image 266
removing BIRT program 12
rendering output and 465, 473, 497
ROM schemas and XML 53
updating designers and 11
writing logging information to 269
writing to disk 219
FileSelectionWizardPage class 539, 543
fill colors (charts) 160
Fill interface 159
fill patterns (charts) 159
fillOval method 687
fillPolygon method 700
filter element 536
filtering data 46, 49, 89, 276
filters 536, 735
finalization code 101, 182
Find declaring extension point setting 373
findDataSet method 303
findDataSource method 303
findElement method 293
finding source archives 378
findTOCByValue method 286
flat file data sources 98, 509

flat files 735
See also text files
Float class 735
float data type 735
floating footers 283
floating point numbers 726, 727, 735
flute.jar 260
folders 12, 197, 379
font files 31
font style constants 297
fontFamily style specifications 54
fonts 31, 735
footer rows 77, 79
footers 75, 85, 283, 327, 736
form e-capable browser. *See* web browsers
format property 470, 607
format styles. *See* styles
FormatCharts charting example 341
FormatCharts class 341
FormatChartsViewer class 341
formatQueryText method 524
formats
chart labels and 162
charting extensions and 649, 687
configuring emitters for 264
customizing report generators for 49
debugging and 197
defined 736
exporting and 466, 601
extracting data and 609, 617
generating ebXML data and 494
generating reports and 257, 278
getting output 87, 88
loading emitters for 388
rendering charts and 155
rendering images and 282, 471
rendering output and. *See* output formats
rendering report items and 397
viewing reports and 43
formatted output 257, 281, 286, 288
Formula Editor. *See* expression builder
formulas. *See* expressions
fragment. *See* plug-in fragments
fragment tag 635
fragment.xml 635
frame objects 113
frameset servlet 32, 65
framework 736
See also classes
Full Eclipse Install package 4, 7
Full Eclipse Installation 7–8, 351
function stubs 118, 124
functions 93, 95, 320, 736
See also methods
fundamental data types. *See* data types

G

Gantt charts 310
See also charts
GanttDataSet interface 310
GEF plug-in 4
GeneratedChartState objects 153, 154
generating
binary documents 281
charts 45, 155, 335
ebXML data formats 494
formatted output 257, 281, 286
HTML documents 266, 278, 281
PDF documents 278
report design files 44, 49, 227
report document files 44
report elements 73, 130
report items 73, 388, 389
reports 49, 73, 227, 262, 270, 286
sample data 325
update site listings 383
generation phase (engine) 65, 73, 80, 86, 91
generation phase events 74
generation services 42, 44, 45, 219
genReport script 20
get method 293
getAllExtensionProperties method 134
getAppContext method 87, 92, 131, 265
getBaseUI method 662, 672
getBlock method 146
getBody method 293
getBookmarkInstance method 271
getBookmarks method 271
getBooleanOption method 285
getBundle method 114
getCategory method 170
getCategoryAxis method 171
getCategoryProvider method 409
getChartInstance method 145
getChartSubtypes method 662, 665

getChildren method 286
getColumnAlias method 97, 133
getColumnCount method 97, 133, 135, 476
getColumnLabel method 97, 133
getColumnMetaData method 97, 134
getColumnName method 97, 133, 135, 528
getColumnNativeTypeName method 97, 133
getColumnType method 97, 133
getColumnTypeName method 97, 133
getColumnTypes method 619
getColumnValue method 134, 135
getCompatibleAxisType method 674
getConnection method 523, 529, 565
getContents method 273
getConvertedChart method 667
getConvertedSeries method 670
getDataSet method 134
getDataSets method 303, 304, 326
getDataSource method 134
getDataSources method 303, 304
getDataType method 274
getDefaultDimension method 662, 672
getDefaultOrientation method 662
getDefaultValue method 274
getDefaultValueList events 66, 72
getDefaultValues method 274
getDescription method 146, 170
getDesignHandle method 87, 292
getDimension method 146, 170
getDisplayName method 659, 662
getDsMinMax method 693
getElementFactory method 323
getEventObject method 688
getExportLabelTag method 501
getExpressionValue method 135
getExtendedProperties method 146
getExtensionID method 134
getExtensionProperty method 134
getExternalContext method 146, 149
getFactory method 170
getGlobalVariable method 87, 131
getGridColumnCount method 146
getHibernateProp method 577
getHibernatePropTypes method 577
getHttpServletRequest method 87, 131
getImage method 662, 665
getInteractivity method 146
getLabel method 274, 405
getLabelPropValue method 501, 502
getLegend method 146, 171
getLocale method 87, 131, 146
getLogger method 146
getMaxConnections method 565
getMaxQueries method 567
getMeasure method 297
getMessage method 87, 131
getMetaData method 566, 570
getModel method 662, 666
getName method 134, 662
getNext method 293
getOption method 285
getOptions method 285
getOutline method 314
getOutputFileName method 285
getOutputFormat method 87, 88, 132
getOutputType method 171
getPageNumber method 286
getPageVariable method 87
getParameterDefn method 273
getParameterDefns method 273
getParameterDisplayText method 87
getParameterType method 273
getParameterValue method 87, 100, 132
getParameterValues method 274
getPersistentGlobalVariable method 87, 90, 132
getPlatform method 265
getPlot method 146
getPrimaryBaseAxes method 315
getPrimaryOrthogonalAxis method 315
getProperty method 270, 310
getPropertyHandle method 298
getQuery method 544
getQueryColumnNames method 545
getQueryText method 134
getRealPath method 266
getRenderOption method 132
getReportElement method 170
getReportName method 616
getReportRunnable method 87, 132, 271
getResource method 87
getResult method 572
getResultSetColumn method 305
getResultSetExpression method 305

getResultSetMetaData method 546, 592, 593
getRoot method 286
getRow method 527
getRowData method 135
getRunTimeSeries method 151, 311
getSampleData method 146
getScript method 147
getSelectionList method 274
getSelectionListForCascadingGroup method 277
getSelectionValueList events 66, 72
getSeries method 663, 673, 689
getSeriesAttributeSheet method 673, 676
getSeriesClass method 673, 674
getSeriesDataComponent method 673, 674, 675
getSeriesDefinitions method 319
getSeriesForLegend method 147
getSeriesThickness method 147
getSQLStateType method 568
getStatus method 288
getString method 527, 529, 572
getStyle method 294
getStyles method 147
getSubType method 147
getSupportedDimensions method 662, 672
getSupportedImageFormats method 282
getTaskType method 132
getText method 294
getTimeZone method 88
getTitle method 147, 171
getTOCTree method 286
getType method 147, 161
getUnits method 147
getURI method 294
getValue method 274
getValueAxes method 171
getValueSeries method 171, 172
getVersion method 147
getWidth method 294, 297
GIF formats 282
global options (design engine) 227
global options (report engine) 220
global variables 84, 85, 87, 90, 736
  See also variables
Glossary 707
glyph 736
  See also character sets; fonts
GObjectFactory class 688
grandchild classes. *See* descendant classes
grandparent classes. *See* ancestor classes
Graphical Editing Framework 4
graphical report design tool 43
graphical user interfaces. *See* user interfaces
graphics. *See* images
graphics API 424
GraphicsUtil class 406
graphs. *See* charts
grid cells. *See* cells
Grid Column Count attribute 146, 147, 148
grid elements 737
grid items 59, 73, 302
GridHandle objects 289
grids 59, 737
  See also grid elements; grid items
group fields. *See* group keys
group footers 736
group headers 738
group keys 738
group sections 76
group slots 294
grouped reports 737
grouping breakpoints 192
grouping data 46, 49, 79, 151
  See also groups
grouping plug-ins 383
GroupOnXSeries charting example 341
GroupOnYAxis charting example 341
groups
  binding data and 76
  building data rows for 77
  charting data and 311, 688
  defined 737
  generating tables or lists and 78
  setting page breaks for 76
GUI components 46
  See also user interfaces

H

handle classes 228, 229, 230
  See also report elements
handle objects 297
handleEvent method 682
handler classes 117

See also event handler classes
HashMap objects 222, 223, 274
hasOption method 285
hasOwnModel property 397
hasStyle property 396
header rows 77, 79
headers 85, 283, 738
headings. *See* column headings
Helios Eclipse SDK software 351, 352
help 752
See also BIRT documentation
hexadecimal numbers 738
Hibernate Core for Java 510
Hibernate data sets 555, 585
Hibernate data sources 173, 554
Hibernate data types 563
Hibernate libraries 577
Hibernate objects 173
Hibernate ODA driver example 555, 564
Hibernate ODA driver plug-in 556, 577
Hibernate ODA drivers 384, 554, 555, 565, 574
Hibernate ODA extensions 554, 555
Hibernate ODA UI example 555
Hibernate ODA UI extension points 580
Hibernate ODA UI extensions 577
Hibernate ODA UI plug-in
building 594
creating projects for 578
developing 554, 584
launching 595
specifying dependencies for 580
specifying run-time settings for 580
testing 594–599
hibernate package 564
hibernate plug-in 554, 556, 577
Hibernate Query Language. *See* HQL statements
hibernate ui plug-in 554
hibernate.cfg 555
HibernateClassSelectionPage class 588
HibernateDataSourceWizard class 584, 588
HibernateDriver class 564, 565
HibernateHqlSelectionPage class 585
HibernatePageHelper class 584, 585, 588
HibernatePropertyPage class 584, 588
HibernateUtil class 565, 573, 577
hibfiles directory 555, 574
hiding report content 90
hierarchy 738
highlight rules 299
home directory 358
home directory parameter 524
HOME property 524
HOME value 519
homeDir parameter 524
host instance (PDE Workbench) 368
HQL (defined) 510
HQL statements
See also Hibernate data sources
adding user interface for 585
creating 554, 566
downloading sample plug-in for 384
executing 571
retrieving data with 572
verifying 590
HTML (defined) 739
HTML elements 730
HTML emitter configurations 265
HTML emitter options 266
HTML emitters 45, 266
HTML extension 727
HTML format parameter 20
HTML rendering options 266, 282
HTML reports
See also web pages
configuring properties for 284
displaying 32, 44
generating 266, 278, 281
opening 288
paginating 283
referencing images in 266
rendering output for 258
saving image files for 266
setting up rendering context for 282
writing event handlers for 76
writing to disk 219
HTML tables 283
HTML tags 282
HTML value 20
HTMLActionHandler class 284
HTMLCompleteImageHander class 266, 267
HTMLRenderOption class 258, 266, 281, 282, 284
HTMLRenderOption objects 266

HTMLServerImageHandler class 266, 267
HTTP (defined) 740
HTTP servlet requests 87, 93, 131
hypercharts 739
hyperlinks 283, 473, 739
HyperText Markup Language. *See* HTML
HyperText Markup Language pages. *See* web pages
HyperText Transfer Protocol. *See* HTTP

I

IActionFilter interface 537
IAggregateFunction interface 649
IBM WebSphere servers 25
ICascadingParameterGroup interface 273
ICell interface 129
ICellInstance interface 130
IChart interface 170
IChartType interface 648, 661, 664
IChartWithAxes interface 170, 171
IChartWithoutAxes interface 170, 172
IColumnMetaData interface 97, 133
ICommonDataExtractionOption interface 610
icon property 470
IConnection interface 522, 564
icons 46, 396, 421, 535
icons directory 663
IContentEmitter interface 470
ICU library 260
icu.jar 260
id attribute 636
id element 537
ID property
  CSV ODA UI extension 535
  CSV rendering extension 470
  data extraction extension 607
  ODA data source extension 518
  rotated label plug-in 395
id property
  data extraction extension 607
  emitter extension 470
  ODA data sets 519
  ODA data sources 518
  property pages 398
  report element adapters 447
IDataExtractionExtension interface 609
IDataExtractionOption interface 610
IDataExtractionTask interface 222
IDataSetEventHandler interface 126, 127
IDataSetInstance interface 133
IDataSetInstance objects 126
IDataSetMetaData interface 523, 564
IDataSetProcessor interface 151
IDataSetRow interface 134
IDataSetRow objects 126
IDataSourceEventHandler interface 125, 126, 127
IDataSourceInstance interface 126, 127
IDataSourceInstance objects 125, 127
identifiers 740
IDesignElement interface 129
IDeviceRenderer interface 155
IDisplayServer interface 155
IDriver interface 518, 522, 564, 565
IEmitterServices interface 474, 499
IEngineTask interface 222
IExternalContext objects 146
IExtractionResults interface 609
IFigure interface 435
IGetParameterDefinitionTask interface 222
IGetParameterDefinitionTask objects 220, 273
IGetParameterDefnTask interface 272
IHTMLImageHandler interface 266, 267
IHTMLImageHandler objects 266
image constants 294
image directory 266
image elements 740
  See also images
image file names 266
image files 29, 266
image formats 282
image handlers 266, 267, 294
Image UI provider 421, 429, 435
ImageHandle objects 294
images
  defined 740
  displaying 429
  exporting 473
  generating HTML reports and 266
  handling events for 137
  referencing 266
  rendering charts as 282, 663
  rendering context and 282

rendering labels as 385
rendering text items as 404, 424, 425
resizing chart legend 163
rotating 408
saving 266
specifying formats for 471
IME files 741
IMenuBuilder interface 443
impl packages 239, 240, 310
Import Projects dialog 354
import statements 112, 654
importing
BIRT projects 350, 354
Java packages 112, 181
source code 5, 350
source code projects 22
importPackage method 112, 181
in_count parameter 183
incrementing record counters 105
index.jsp 28
indexing column position 96
inheritance 740
initCustomControl method 542, 585, 587
initialization code 181
initialize events 67, 72, 80
initialize method
building report designs and 102, 128
charting extensions and 659
importing Java packages and 112
rendering CSV output and 474
rendering XML output and 499
script execution and 101
writing event handlers and 85
XMLDataExtractionImpl 615
initializeControl method 590
initializing report elements 71
initSessionFactory method 574
initXMLOptions method 615
inner joins 740
See also joins
inner reports. *See* subreports
input 194, 272, 276, 539
Input Method Editor files 741
input parameters 183
input sources. *See* data sources
input streams 306
inputParams array 183
Install New Software command 384
install new software manager 381, 382, 384
installation
BIRT Full Eclipse Install 7–8
BIRT RCP Report Designer 8–9
BIRT Report Designer 7, 350
chart engine 15–16
demo database 18
DTP Integration package 17
Eclipse SDK software 352
language packs 10–11
plug-ins 259, 362
report viewer 26–28
Samples package 21
source code 21–22
testing 8, 9, 350
troubleshooting 9–10
TrueType fonts 31
upgrades and 11
WTP plug-ins 22
installation errors 9
instance interfaces 130
instance property 322, 325
instance variables 741
See also class variables; variables
instances. *See* objects
instantiation 362, 439, 741
int data type 742
Integer class 742
integer data type. See int data type
integrating with report generators 49
interactive features (charts) 146, 147, 155, 688
interactive features (report items) 421
Interactivity property 435
InteractivityCharts example 342
interfaces
See also application programming interfaces; user interfaces
adapter class extensions and 118
adapter classes compared to 124
chart engine API and 239, 240, 310
chart events and 151, 155, 161, 163
charting applications and 169, 170
charting extensions and 648, 658, 661
data extraction extensions and 609
data row objects and 134, 135
data set objects and 133
defined 742

interfaces *(continued)*
design engine API and 230, 238
design model objects and 226
developing with 217
event handlers and 117, 124, 128
generating HTML documents and 266, 267
generating reports and 263
hierarchical diagrams for 218
loading BIRT plug-ins and 265, 266
naming conventions for 123
ODA drivers and 509, 517, 522, 565
opening report files and 258
plug-in extensions and 360
rendering output and 258
rendering reports and 473
report designs and 128
report document files and 271
report elements and 129–135
report engine API and 219, 225, 257
report item extensions and 398
report parameters and 272
scripted data sets and 127
scripted data sources and 126
web services connections and 185
interfaces package 233, 235
International Components for Unicode. *See* ICU library
internationalization 742
See also locales
invalid data types 195
IP addresses 742
IPageGenerator interface 398, 448
IParameterDefnBase interface 272
IParameterGroupDefn interface 272, 273
IParameterSelectionChoice interface 273, 274
IParameterSelectionChoice objects 277
IPlatformContext interface 264, 265, 266
IPlatformContext objects 265
IPrimitiveRenderer interface 686, 687
IQuery interface 522, 564
IRenderOption interface 281
IRenderOption objects 258
IRenderTask interface 222, 279, 281
IRenderTask objects 221, 258
IReportContent interface 475, 499
IReportContext interface 131
IReportContext objects 125, 126, 128
IReportDesign objects 128
IReportDocument interface 221, 271
IReportDocument objects 258, 270, 271
IReportElement interface 129
IReportElementInstance interface 135
IReportEngineFactory interface 220, 267
IReportEventHandler interface 128
IReportItemBuilderUI interface 440
IReportItemFactory interface 396
IReportItemFigureProvider interface 395, 421
IReportItemImageProvider interface 395, 421
IReportItemLabelProvider interface 421
IReportItemPresentation interface 397, 424
IReportRunnable interface 221
IReportRunnable objects 258, 270, 271
IResultSet interface 523, 564
IResultSetMetaData interface 523, 564
IRowData interface 135
IRowInstance interface 130
IRunAndRenderTask interface 223, 278, 281
IRunAndRenderTask objects 223, 258, 284
IRunTask interface 222, 278
IRunTask objects 222
IScalarParameterDefn interface 272, 274
IScalarParameterDefn objects 273
isColorByCategory method 171
isComputedColumn method 97, 133
IScriptedDataSourceEventHandler interface 127
isDimensionSupported method 662
isEmpty method 273
isEncryptable attribute 519
isEncryptable property 397
ISeriesRenderer interface 155, 648, 686
ISeriesUIProvider interface 673
isExportDataType variable 615
isHidden property 471, 607
isHorizontal method 171
isList property 397
isLocaleNeutral element 612
isLocaleNeutral variable 615
isNameRequired property 396
isRowInFooterBand method 477
IStatusHandler interface 265

IStyleProcessor interface 342
isv plug-ins 261
isValidAggregationType method 673
iterator method 293, 572
iterator objects 182
ITextItem interface 129
ITOCTree interface 286

J

J2EE application servers 4
J2EE applications 31
J2EE environments 15, 19, 743
J2SE environments 743, 744
.jar files
  accessing demo database and 18
  accessing Java classes and 113
  accessing JDBC data sources and 261
  adding to classpaths 118
  building plug-in fragments and 642
  building plug-ins and 390
  building update sites and 382
  building web viewer 350
  changing source code and 356
  charting API examples and 340
  charting applications and 308
  configuring BIRT engines and 220, 228
  creating event handlers and 117, 164
  creating JNDI connections and 33
  debugging and 198
  defined 743
  deploying Java classes and 183
  deploying to web servers and 26
  developing applications and 259, 261
  developing ODA extensions and 513
  installing designers and 9
  installing report viewer and 27, 28
  running reports and 118
  setting driver location and 92
  testing CSV rendering extension and 484
Java. *See* Java programming language
Java 2 Enterprise Edition. *See* J2EE environments
Java 2 Runtime Standard Edition. *See* J2SE environments
Java adapter classes. *See* adapter classes; Java event handlers
Java APIs 44
Java applets 708
Java applications
  See also applications
  adding charting capabilities to 15, 307
  adding reporting capabilities to 19
  creating 49, 743
  generating charts and 46
  rendering reports and 219
Java archives. *See* .jar files
Java Attribute Editor 376
Java Build Path page 118
Java Build Path settings 261
Java classes
  accessing 54, 112, 113, 279
  associating with report elements 121
  building report designs and 226, 227, 230
  building scripted data sources and 182
  changing chart objects and 239, 240
  changing report designs and 262
  debugging event handlers and 194
  deploying 113, 183
  developing with 44, 217
  generating reports and 263, 270
  handling events and 117, 118, 121
  implementing data extraction extensions and 609, 610, 614
  implementing rotated label plug-in and 390, 404
  instantiating 115
  loading 115, 519, 574
  naming 119, 123
  opening data sets and 127
  opening data sources and 125, 127
  referencing 112, 181
  registering 320
  rendering CSV output and 473, 485
  rendering XML output and 498
  running plug-in instances and 360, 364, 368
  running reports and 219, 220, 223
  scripting for 112, 113
  setting location of 113
  setting properties for 119, 376
Java code 25, 63, 112, 114, 117
Java compiler 350
Java Database Connectivity. *See* JDBC
Java development environment 64
Java Development Kit. *See* JDK software
Java Development Tools interface 194

Java editor 120
Java event handler classes. *See* Java classes
Java event handler examples 135–142, 343
Java event handlers
  See also event handlers
  accessing classes for 121
  accessing JAR files for 118
  accessing methods for 114
  associating context objects with 131
  building charts and 143, 164–167, 170, 320
  building data sets and 126, 127, 180
  building data sources and 125, 126
  changing designs and 128
  compared to JavaScript 63
  creating 64, 117
  debugging 142, 188, 194, 198
  extending adapter classes for 118
  generating report elements and 117, 123
  generating report items and 125
  integrating with JavaScript 112, 113
  passing JavaScript variables to 84
  retrieving external content and 149
  saving 113
  tracking execution sequence for 101
  type casting and 152, 153
  viewing 64
Java EventHandler debugger 188
Java files 64
Java interfaces 117, 128, 217, 742
Java methods 113
Java Naming and Directory Interface 743
Java objects 113, 173, 181
Java packages 112, 181
Java programming language 742
  See also Java code
Java projects 117, 119, 121
Java report generator 49
Java Runtime Environment 25
Java source files 119
Java Virtual Machines. *See* JVMs
JAVA_HOME variable 350
java.lang package 112
JavaBeans 744
Javadoc files 261
Javadocs 218, 257, 261
JavaScript
  See also source code
  accessing data and 173
  accessing Java classes for 112
  concatenating values and 75
  creating event handlers and. *See* JavaScript event handlers
  customizing reports and 262
  defined 744
  entering 94
  line breaks in 320
  validating syntax of 189
  wrapping Java code in 112, 113
JavaScript array properties 97
JavaScript debugger 188
JavaScript event handler examples 107–111, 343
JavaScript event handlers
  See also event handlers
  accessing application context and 91
  accessing methods for 114
  adding to designs 83
  adding variables to 84, 85, 90
  associating context objects with 86
  building charts and 143, 148, 167–169, 170, 320
  building data sets and 181
  building data sources and 181
  changing connection properties and 99
  changing designs and 89
  changing property values and 98
  changing styles and 88, 90
  closing connections and 92
  compared to Java 63
  creating 63, 83
    tutorial for 103–107
  debugging 188, 190
  dropping report elements and 90
  generating report elements and 93
  getting column information for 96, 97
  getting parameter values and 100
  previewing 96
  retrieving external content and 149
  running queries and 98, 100
  running reports and 93
  saving 113
  setting parameter values and 101
  setting properties with 94

tracking execution sequence for 101
type casting and 152
viewing 64
JavaScript expressions 86, 188, 456, 458
JavaScript files 111
JavaScript functions 93, 95
JavaScript implementation 83
JavaScript language 744
JavaScript library 260
JavaScript names 96
JavaScript objects 113
JavaScript palette 106
JavaScript variables 84, 90
JavaScriptViewer application 343
JavaServer Pages. *See* JSPs
JavaViewer application 343
JBoss servers 25, 26
JDBC (defined) 743
JDBC connection profiles 510
JDBC connections 33, 99
JDBC data sets 32, 69, 98
JDBC data sources 32, 98, 261, 510
JDBC driver manager 9
JDBC drivers
adding nonstandard 26
configuring Hibernate data sources for 555, 574
configuring JNDI service providers for 32, 33
loading 261
setting location of 92
jdbc plug-in 92
JDK software 4, 7, 350, 743
JDT interface 194
JNDI (defined) 743
JNDI connection objects 33
JNDI connection properties 33
JNDI name paths 32
JNDI resource files 33, 34
JNDI service providers 32, 33
JNDI URL property 32, 33
jndi.properties file 33
join conditions 745
joins 60, 745
joint data sets 60, 741, 745
JointDataSet element 60
JPEG formats 155
JPG device renderers 649
JPG formats 171, 282, 687
JRE software 25
js.jar 260
JSPs 28, 744
JVM path 10
JVMs 10, 744

K

keyboard input 194
keywords 745

L

-l locale command line option 20
label elements 57, 389, 745
label items 94, 302
Label object (charts) 160
label tag 501
Label UI provider 421
LabelImpl class 160
labels (charts). *See* chart labels
labels (reports) 56, 385, 405, 745
See also label elements; label items
LabelStyleProcessor class 342
language packs 10–11, 635, 645
language settings 643
language-specific environments. *See* locales
Launch an Eclipse application setting 381
launch configuration entry point 191
launch configuration types 196
launching. *See* opening
layout editor 388, 746
layout package 240, 253
Layout page. *See* layout editor
LAYOUT_PREFERENCE_AUTO value 283
LAYOUT_PREFERENCE_FIXED value 283
layout.xsd 652
layouts 49, 229, 746
lazy load 746
See also source code
left outer joins. *See* outer joins
legend anchor points 158
legend area (charts) 146, 163, 314
legend entry bounds 163
Legend interface 314
legend line properties 314
legend markers 155

Legend objects 314
LegendEntryRenderingHints class 163
Level class 269
level-break listings. *See* grouped reports
lib directory 118, 259
lib plug-in 182
libraries
  See also component libraries
  accessing 48, 226, 257, 290
  building plug-in extensions and 369, 377, 378
  building plug-in fragments and 636
  configuring report engine for 263
  creating reporting applications and 290
  defined 746
  deploying applications and 262
  deploying report engine and 259
  developing applications and 259
  developing report designers and 229
  finding source archives for 378
  installing BIRT and 5
  naming 378
  reusing designs and 44
  running charting applications and 16
  running Hibernate drivers and 577
  scripting and 112
Library element (designs) 59
library extension element 636
library files 48, 290, 747
  See also libraries
library names 195
library-specific properties 229
LibraryHandle class 229, 290
Life Cycle property 435
line attributes (charts) 678, 681, 682, 689
line break characters 320
line breaks (JavaScript) 320
line charts 307, 310
  See also charts
line-render events 688
LINEAR_LITERAL value 161
LineAttribute objects 314
LineRenderEvent method 700
LineSeries objects 317
links. *See* hyperlinks
Linux servers 10, 15, 19
list elements 302, 747
  See also list items; lists
list execution sequence (events) 78–79
list items 58, 59, 73
  See also list elements; lists
Listener interface 677
listeners. *See* event listeners
Listing element 58
listing reports 747
ListingGroup items 79
lists
  See also list elements; list items
  accessing report designs and 272
  adding to reports 59
  binding to data sets 75
  building data rows for 77
  displaying parameters and 274, 277
  getting column information for 96, 97
  grouping data for 78
  handling events for 75, 77, 78
literal strings 456
load_classicmodels.sql 18
loadClass method 115
LoadExportSchema class 498, 504
loading
  class files 376
  document files 221
  Java classes 115, 519
  JDBC drivers 261
  plug-in fragments 635
  plug-ins 265, 362, 370, 465
  report designs 221, 258, 270, 291
  report documents 258, 270, 271
  resource files 265
  source code 362
  XML schema files 504
local classes and interfaces 218
local variables 84, 747
  See also global variables; variables
locale codes 11
  See also locales
Locale element 611
Locale objects 146
Locale parameter 20
Locale property 614
locale settings 11
locale variable 614
locale-independent formats 273
locales
  building charts for 146
  changing 11

converting strings and 273, 635
creating session handles for 228
defined 748
developing report item extension for 417
exporting data and 612
getting 87, 131, 146
installing language packs for 10–11
localizing display names for 520
localizing property groups for 519
localizing viewer messages for 636, 640
selecting 11
setting 146, 222, 258
testing plug-in fragments for 637, 643
localization 10, 748
See also locales
localized messages 87
log file names 269
log files 29, 264, 269
log package 240, 245
LOGARITHMIC_LITERAL value 161
Logger class 269
Logger objects 146
logging chart engine tasks 245
logging classes 269
logging configurations 264
logging environments 269
logging levels (applications) 269, 270
logging levels (emitters) 475, 499, 500
logging messages 146, 269
logging output 221, 475, 499
logging script execution 101–103
logging threshold 269

M

-m command line option 19
machine-readable metadata 360
manifest files 360, 362, 635, 748
manifest headers 368, 369
Manifest wizard 521
MANIFEST.MF 368
See also manifest files
MANIFEST.MF page (PDE Editor) 369
mapping to report viewer 28, 29
margins 283
marker lines (charts) 161
marker range (charts) 161
markers (charts) 147, 689, 690
markup languages 727, 733, 739
See also elements; tags
mashup 748
master page components 229
master page footers 283
master pages
adding autotext items to 85
adding report items to 302
creating 88, 283
defined 748
handling events for 72, 73, 74
reusing 49
setting margins for 283
setting properties for 58
MasterPage element 58
Math class 112
matrix reports. *See* cross-tab reports
measurement units 228
measures 749
See also data cubes
Member element 57
member variables 749
See also field variables; variables
members 749
See also methods; variables
memory 362
menu builders 443
menuBuilders plug-in 443
menus 443, 445
messages 87
See also error messages; logging messages
Messages class
CSV ODA driver extension 523, 529
CSV ODA UI extension 538
Hibernate ODA extension 564, 585
Messages_es.properties 640
messages.properties file 538, 585, 635, 636
metadata
accessing CSV data sources and 523, 524, 528
accessing Hibernate data sources and 567, 570
adding ROM elements and 53–57
building JDBC data sets and 32
building ODA data sets and 545
creating plug-in extensions and 360
creating property sheets and 54

metadata *(continued)*
defined 750
examining data row 222
extracting data and 609, 619
getting column information from 97, 133
getting data set 134
updating ODA data sets and 539, 547
metadata interface 133
meter charts 239, 310, 312, 314
See also charts
MeterChartExample application 345
Method property 60
method stubs 485, 511
methods
See also functions
calling 114
chart blocks and 158
chart engine generator and 150
chart event handlers and 143, 145
chart instance objects and 146, 147, 148, 239
chart properties and 313
charting extensions and 658, 661, 674, 686
charting interfaces and 155, 161, 170, 171
column metadata and 97, 133
data extraction extension and 609, 615
data rows and 134, 135
data set elements and 126, 133
data source elements and 125
defined 750
documentation for 257
event handler classes and 123
event handler interfaces and 117, 124
event handlers and 84, 85, 112, 113
external context objects and 149
importing Java packages and 112
ODA extensions and 522
ODA UI extensions and 540, 541, 543
opening report files and 258
overriding 124
providing external values for 265
rendering output and 473, 498
report components and 228, 229, 292
report context objects and 86, 131
report design events and 128
report element events and 129
report engine configurations and 264
report item events and 125
run requests and 93
running plug-in instances and 364
running rotated label plug-in and 404
scripted data sets and 127, 174
scripted data sources and 127, 174
selecting 95
starting platforms and 267
viewing arguments for 124
viewing stack for 191
web services connections and 184
Microsoft Internet Explorer. *See* web browsers
milestone build 5
MIME types 470, 609
mimeType property 470, 607
missing designer functionality 9
modal windows 750
mode 750
model API 289, 290
model api package 257, 262
model definitions 456
model element 360, 395
model packages (charts) 240, 245
model plug-in 53
model renderer elements 686
model renderers (charts) 155, 648, 685, 689
model.xsd 652
modelapi.jar 260, 261
modeless windows 750
modelodaapi.jar 260
modelRenderer element 686
modelrenderers extension point 685
modelrenderers plug-in 648
ModifyListener method 586
module handles 228
ModuleHandle class 228
Mozilla Firefox. *See* web browsers
Mozilla Rhino 83, 112, 114
multicolumn page layouts 59
multidimensional data 750
multilingual reports. *See* locales; localization
multipass aggregation 68
Multipurpose Internet Mail Extensions. *See* MIME types
multithreaded applications 751

multivalued parameters 100
myChart.chart file 309
mysql command line interface 18
MySQL databases 17, 18
MySQL installation scripts 577
MySQL software 5

N

name attribute
  dataSource element 519
  plug-in fragments 636
name collisions 112
name element 397
Name property
  CSV ODA UI extension 535
  CSV rendering extension 470
  data extraction extension 607
  ODA data source extension 518
  rotated label plug-in 395
name property 396, 607
name space 229
name variable 284
names 96, 134
  See also aliases; display names
naming
  data extraction extensions 607
  Java classes 119, 123
  ODA driver extensions 518
  ODA UI extensions 535
  output files 281, 285
  plug-in libraries 378
  report item extensions 360, 395, 396
  report items 293
  report rendering extensions 470
  scripted data sets 176
  scripted data sources 175
naming conventions 123
National Language Support applications. *See* NLS applications
nativeDataType property 520
nativeDataTypeCode property 520
Navigator 751
needOutputResultSet property 471
Neutral local property 615
New Chart wizard. *See* chart builder
New Data Set wizard 176
New Data Source wizard 175
New Extension wizard 372
New Java Class dialog 119
New Java Class wizard 485
new line characters 478
new line tags 479
New Plug-in Project wizard 366, 367
New Project wizard 366
New Source Folder dialog 379
new_report.rpdesign 626
newElement method 302
newExtendedItem method 325
newInstance method 115
newQuery method 523, 566
newReportItem method 405
newSession method 228, 292
newSessionHandle method 227
newsgroups xix
newWizard element 510
next method 527, 572
nightly builds 6
-nl command line option 11
NL1 fragment 636
NLpack1 language pack 10
NLS applications 635
NLS language pack 645
NLS messages.properties file 640
NLS plug-in 635, 640
node 751
NonGroupOnXSeries.rptdesign 341
NonGroupOnYAxis.rptdesign 341
non-relational database sources 509
nonstandard drivers 26
non-visual components 51
non-visual elements 291
null values 152, 184, 519, 751
Number class 112
number parameters 274
NumberDataSet interface 151, 311
numeric data types 751
numeric expressions 751
numeric values 612
N-up reports. *See* multicolumn page layouts

O

-o command line option 20
object attribute filters 536
object cache 687
Object class 112
object factory (charts) 688

object libraries.
  See component libraries; libraries
object references 182, 184
object-oriented programming 510, 752
objects
  accessing 173, 181, 280
  adding to application context 91
  defined 752
  determing if exists 91
  instantiating 181, 362, 439
  iterating through 92
  registering service 365
  resetting to null 182
  saving 132
  storing 91
ODA (defined) 752
ODA API 509, 510, 522
ODA API Reference 511
ODA connection constants 540
ODA connection profiles 510, 520, 535
ODA connection properties page 511
oda connectionProfile extension 510, 520
ODA context class loader 519
ODA CSV driver extension
  adding data access functionality to 517–520
  building projects for 513–??
  compiling and debugging 513
  developing 511, 512, 522
  downloading source code for 511
  setting dependencies for 521
ODA CSV driver plug-ins 512, 513
ODA data set designer page 538
ODA data set editor page 530, 537
ODA data set elements 60
  See also ODA data sets
ODA data set extensions 519, 530
  See also ODA extensions
ODA data set handles 304
ODA data set IDs 537
ODA data set UI extension 537
ODA data set wizard page 539, 543, 588
ODA data set wizards 537, 538
ODA data sets
  adding 304, 530
  creating 523, 528
  editing 510, 537
  mapping to 520
  selecting 538, 552
  specifying 537, 596
  updating 539, 547
ODA data source editor page 530, 538
ODA data source elements 60
  See also ODA data sources
ODA data source extension elements 518, 537
ODA data source extensions 509, 517, 530
  See also ODA extensions
ODA data source handles 303
ODA data source wizard page 539, 543
ODA data source wizards 538
ODA data sources
  accessing 99, 509, 510, 517
  adding 530
  closing connections to 92
  connecting to 510, 511, 522
  creating queries for 544
  preparing queries for 524
  retrieving data from 509, 522
  selecting 510, 538
  setting connection properties for 99
  specifying 537, 539, 551, 596
ODA data types 520
oda dataSource extension 509, 510, 517
ODA driver constants 529
ODA driver extension examples
  CSV data sources 384, 511
  Hibernate data sources 555, 564
ODA drivers
  accessing CSV data sources and 511
  accessing JDBC data sources and 99
  adding connection profiles for 510, 520, 535
  adding to BIRT framework 510
  adding user interface for 385, 510, 530, 537
  changing connection properties for 511
  configuring 518
  creating 554
  customizing 46, 50
  defined 752
  defining error messages for 523
  extending functionality of 509, 512
  initializing connections for 522
  running 522
  testing 511

ODA DTP framework 46, 509, 538
ODA DTP interfaces 509, 519, 522
ODA extension identifiers 518, 519
ODA extension points 510, 511
ODA extensions
building 513, 521
creating projects for 509
data type mappings for 520
defining connection properties and 99
developing 509, 510, 511, 554
downloading sample plug-ins for 511, 513
retrieving data and 46
setting design-time properties for 518
setting display names for 519, 520
setting run-time properties for 518
testing 548
ODA framework 510
ODA Hibernate drivers. *See* Hibernate drivers
ODA packages 510
oda plug-in 512, 513
ODA Runtime driver wizard 521
ODA UI extensions 512, 530, 534, 537
ODA UI plug-in 512, 535, 538, 548
ODA UI plug-in projects 511, 531–534
ODA user interface
adding connection profiles to 510, 535
adding data sources to 530
building CSV drivers and 512
creating 385, 510, 530, 537
defining editor pages for 537
defining property pages for 511, 536, 538
defining wizard pages for 538, 543
enabling page editing functionality for 540, 541
generating messages for 538
testing 548
ODA wizards 509, 511, 521, 538
ODA_DEFAULT_CHARSET constant 540
OdaDataSet element 60
OdaDataSetHandle class 304
OdaDataSource element 60
OdaDataSourceHandle class 303
odadesignapi.jar 260
odaDriverClass property 99
ODAHOME constant 540
OdaJDBCDriverPassInConnection setting 92
OdaJDBCDriverPassInConnectionClose AfterUse setting 92
odaPassword property 99
odaScalarDataType property 520
odaURL property 99
odaUser property 99
odaVersion property 518
ODBC (defined) 753
OLAP (defined) 752
olap package 233
onCreate events
bound columns and 75
lists and 78
report items and 52, 71, 73, 76, 77
run-time properties and 130
tables and 78
onCreate method 84, 125
onCreate scripting example 108
onFetch events 70
onFetch method 96, 126
Online Analytical Processing systems. *See* OLAP
online help 752
See also documentation
onPageBreak events 52, 68, 71, 74, 79
onPageBreak method 125
onPageBreak script 85
onPageEnd events 68
onPageEnd script 85
onPageStart events 68
onPageStart script 85
onPrepare events 52, 71, 72, 79
onPrepare method 125
onPrepare script 107
onRender events
lists and 78
report elements and 73, 88
report items and 52, 71, 80
run-time properties and 130
tables and 78
onRender method 109, 125
onRender script 109
onRowSets method 406, 425, 457
open data access 752
See also ODA
open database connectivity. *See* ODBC
Open Debug Dialog command 142

open events 69, 70
Open extension point description setting 374
Open extension point schema setting 373
open method
  Hibernate drivers and 565
  iterator objects and 182
  scripted data sets and 127, 178, 181
  scripted data sources and 127, 174, 178
Open Perspective command 8
Open Services Gateway Initiative. *See* OSGi
open source 753
open source projects xix, 349
  See also projects
open-source software development. *See* Eclipse
opening
  builders 439
  class files 376
  configuration files 29
  connections 69, 522
  data sets 69, 126
  data sources 125
  debug configurations 197
  Debug perspective 196
  document files 221, 258, 270, 271
  Export Wizard 381
  expression builder 458, 460
  HTML reports 288
  output files 475, 499
  PDF documents 288
  Plug-in Development perspective 366
  report designs 221, 258, 270, 291
  schema definition files 373, 374
  scripted data sets 70, 127
  scripted data sources 69, 127
  Tomcat Manager 30
openReportDesign method 220, 221, 258, 270
openReportDocument method 220, 221, 258, 270
operators 753
Organize Manifest wizard 521
orthogonal axis values 315, 714
  See also pie charts
orthogonal series 319, 675, 691
OSGi Alliance web site 365
OSGi arguments 220, 227
OSGi class loader 519
OSGi configuration property type 264
OSGi error messages 309
OSGi platform 265, 267, 365
OSGi resource bundles 364, 368
out_msg parameter 183
outer joins 753
  See also joins
outline element 396
Outline property 421
Outline view (BIRT) 194
Outline view (Eclipse) 369, 753
outlining chart legends 314
output
  accessing formatted 288
  changing source code and 356
  creating charts and 43, 241, 649, 687
  customizing reports and 262
  determining 93
  generating 257, 286
  handling events for 71
  rendering 45, 64, 80, 219, 258, 465
  setting options for 281, 471
  setting pagination for 470
  storing XML 614
  validating 53
  viewing CSV 481, 486
  viewing HTML 20, 32, 44, 281
  viewing XML 507, 609
output devices 241
output file name parameter 20
output file names 258, 473, 497
output files
  See also specific file type
  accessing 288
  closing 476, 500
  creating 101
  debugging and 197
  exporting to 609
  naming 281, 285
  opening 475, 499
  rendering CSV output and 473
  rendering XML output and 494, 497, 507
  tracking script execution in 101, 103
output format parameter 19
output formats
  charting extensions and 649, 687
  configuring emitters for 264

customizing 278, 388, 465, 602
debugging and 197
default 257
developing report generator for 49
exporting and 466, 601
extracting data and 609, 617
generating ebXML data and 494
generating reports and 45, 281
getting 87, 88, 466, 602
rendering charts and 155
rendering images and 471
rendering report items and 397
rendering reports and 221, 222, 223
setting 258, 470
specifying MIME types for 470, 609
specifying properties for 264
viewing reports and 43, 466
output method 617
output parameters 183
output streams
converting XML data for 620
creating file 474, 499, 609
formatting 617
generating reports and 281, 288
setting 258
output types 171
See also output formats
outputDisplayNone property 471
outputParams array 183
overriding methods 124
overriding report parameters 20, 93
Overview page (PDE Editor) 369, 380, 381
overwrite property 448

P

-p command line option 20
package (defined) 754
Package Explorer 369
package names 119
package visibility settings 369, 371
packages 4, 15, 217, 218
Packages object 112
Packages prefix 112
page break events 76
page breaks 71, 76, 470
page components 754
page counts 221
page events 68, 76
page extension element 536
page footers 85, 283, 736
page generator 449
page headers 85, 283, 738
page information 271
page layouts 49, 229, 746
See also designs; master pages
page properties 58
page variables 68, 85, 87
pages 754
See also windows
pagination 85, 470
pagination controls 80
pagination property 470
palette (color) 228
palette (data series) 684
palette (report items) 388, 420
palette extension element 396
Palette page. *See* Palette view
Palette view 46, 754
parameter definition tasks 273, 276
parameter definitions 222
parameter entry box 66, 72
parameter events 66, 72
parameter files 19
parameter groups 272, 273
parameter names 272, 273
PARAMETER_GROUP value 273
parameters
See also data set parameters; report parameters
accessing information about 222
chart script context and 145, 149
configuring report viewer and 28, 29
connecting to web services and 184
creating scripted data sets and 183
customizing connections and 184
debugging event handlers and 188
defined 755
defining session-specific 227
extracting data and 611, 616
grouping. *See* parameter groups
running reports and 93
setting multiple values for 66
validating 222, 223
viewing reports and 32
Parameters element 611
parameterValues parameter 185
param-value element 30

parent class. *See* superclasses
parse method 274
passwords 99, 755
PATH language. *See* XML PATH language
paths
  BIRT APIs 261
  BIRT home directory 259, 263
  debugger and 196
  Eclipse home directory 358
  font files 31
  image files and 266
  JNDI resource files 33
  report design files 28
  report viewer configurations 29
  reports and 610, 616
patterns 755
  See also object-oriented programming
PDE (defined) 729
PDE Manifest Editor 369, 517
PDE Workbench 368–384
  See also Eclipse Plug-in Development Environment
PDF converter 755
PDF device renderers 649
PDF documents
  building charts for 342
  displaying 31, 32
  generating 278
  installing fonts for 31
  missing content in 31
  opening 288
  rendering output for 258
PDF emitter 45
PDF formats 155, 687
PDF reports. *See* PDF documents
PDF viewer. *See* Acrobat Reader
PDFChartGenerator charting example 342
PDFChartGenerator class 342
PDFRenderOption class 281
performance 76
persistent files 43
persistent variables 86, 87, 90, 132
perspectives 7, 728
pictures. *See* images
pie chart events 153
pie charts 239, 307, 310, 314
  See also charts
placeComponents method 678
Platform class 114, 267
platform context 265
Platform context property type 264
PlatformFileContext class 265
platforms 7, 267, 755
PlatformServletContext class 266
PlatformServletContext objects 266
plot area (charts) 146, 154, 158, 313, 314
Plot interface 313
Plot object 314
plot properties (charts) 313, 314
plot property 312
plug-in 756
plug-in activation framework 362
plug-in archive files 360, 371
Plugin class 364
Plug-in Development Environment 360, 366, 729
Plug-in Development perspective 366
  See also PDE Workbench
plug-in drivers 510
plugin element 363
plug-in extension examples 384
plug-in extension points 362, 369, 373
  See also extension points
plug-in extension schema definitions 373, 374
plug-in extensions
  See also extensions
  accessing ODA data sources and 509, 510
  adding report items and 47, 387, 389, 416, 417
  adding reporting functionality and 359
  building 377–380
  changing MIME types for 609
  creating 370–377
  debugging and 194
  declaring 369
  deploying 381–384
  developing CSV ODA driver 511
  downloading source code for 384
  exporting 390
  extracting data and 601
  implementing 359
  rendering reports and 465
  setting class attributes for 376
  specifying 372

testing 380–381
unloading 220
viewing property annotations for 376
plug-in fragment example 636, 640
plug-in fragment manifest file 635
plug-in fragment projects 637–639
plug-in fragment source code 385
plug-in fragments 635, 641, 756
plug-in manifest file 360, 362, 363, 369
See also manifest files
plug-in manifest headers 368, 369
plug-in projects 366–368, 369
plug-in registry 362, 466, 602
plug-in run-time class 360, 362, 364
Plug-in Selection dialog 370
plugin.xml 362, 363
See also plug-in manifest file
Plug-in.xml page (PDE Editor) 369
plugin-id attribute 636
plug-ins
accessing BIRT 265, 359
accessing ODA data sources and 509, 510
activating 368, 384
building rotated label sample extension and 390, 393
caching conflicts and 9
calling methods in 114
charting extensions and 648, 649, 650
compiling code for 350, 354, 380, 390
configuring report engine for 263
controlling life cycle of 368
creating Hibernate driver 556, 577, 584
creating Hibernate ODA UI 594
debugging 390
defined 756
deploying 262, 360, 382
designing reports and 227, 228
developing applications and 259
developing content for 4, 360, 366, 373
developing with 359
downloading sample extensions for 384
downloading source code for 21
editing project settings for 369, 370
exporting 381
extending functionality of 370
finding source archives for 378
grouping as features 382, 383
installing 22, 259, 362
integrating with Eclipse 39
loading 265, 362, 370, 465
rendering output and 465
running 362, 364
selecting run-time libraries for 378
setting dependencies for 370
setting location of 220
setting properties for 367
testing 360, 368
updating 384
verifying run-time archive for 371
viewing information about 369
viewing project settings for 369
plugins directory 259, 265, 362, 513
plug-ins packages 382
PNG device renderers 649
PNG formats 155, 171, 282, 687
PolygonRenderEvent event 700
polymorphism 756
populate method 151
PostScript emitter 45
PowerPoint converter 756
PPT emitter 45
predefined layouts. *See* master pages
predefined values 719
PreferenceServlet charting example 342
preparation phase (engine) 65, 71, 86
preparation phase events 72
preparation services 42
prepare method 524, 568
prepare phase (charts) 150
prepareMetaData method 524
presentation engine 388, 390, 466, 602
presentation implementer class 424
presentation phase (engine) 66, 80, 86, 91
Presentation property 435
presentation services 42, 45
preview engine 482
preview servlet 65
previewer 757
previewing
charts 671
reports 44, 65, 219
rotated text images 429
scripts 96
source code 104
primary base axis (charts) 315
PrimitiveRenderEvent class 687

println method 102
PrintWriter objects 102
priority property 398, 447
private classes 217
private styles 297
PrivateStyleHandle class 297
procedures 757
  See also functions; methods
process 757
Process Console 194
ProductsByProductLineDBG.rptdesign 197
program updates 381, 382
program upgrades 11
programmers xxi
programming environments 64, 259
programming interfaces 128
programming languages 25, 742
  See also scripting languages
progressive viewing 80
projects
  accessing custom data sources and 509
  accessing Java classes for 113
  adding event handler classes to 118, 121
  building BIRT 349, 354–356
  building Eclipse xix
  building update sites and 382
  changing settings for 369, 370
  charting extensions and 654
  creating
    data extraction extension 603–605
    ODA driver extension 511, 513, 556
    ODA UI extension 511, 531, 578
    report rendering extension 466, 494
    rotated label extension 390–392
  developing plug-in extensions and 378, 380
  developing plug-in fragments and 637–639
  developing plug-ins and 366, 367
  importing 22, 350, 354
  running rotated label extension and 381
  selecting 119
  setting Javadoc locations for 261
  testing 356
  viewing settings for 369
properties
  accessing 93, 228, 229, 294
  adding text controls for 536
  builders and 439
  changing 98, 445
  charting applications and 310, 312–315, 341
  charting extensions 652
  charts and. *See* chart properties
  connection profiles 539
  connections 99, 125, 184
  controlling visibility of 56
  CSV emitters and 470
  data extraction extensions 607, 610, 614
  data sets and 134, 304
  defined 758
  expressions and 456, 457
  getting 87
  Hibernate data sources and 587, 588
  Java classes and 119, 376
  Java packages and 112
  JavaScript arrays and 97
  JNDI connection 33
  libraries and 229
  master pages and 58
  ODA CSV extensions and 518, 519
  ODA data type mappings and 520
  plug-ins and 367
  report designs and 59, 229, 270
  report elements and 130, 291, 299
  report engine and 263, 264, 265, 279
  report items and 59, 94, 294, 296, 298
  ROM elements and 52, 53
  rotated label plug-in 395, 396
  rotated text items 418, 420, 429, 439
  scripting and 90
  user sessions and 228
  XML emitters and 501, 503
Properties element 610
.properties files 10, 360
properties list 95, 757
Properties page. *See* Properties view
properties tag 614
Properties view 757
property annotations 376
property binding 98
property definitions 227
Property Editor 758
property editors 445, 530

property groups 519
property handles 297, 298
property name element 53
property names 519
property page adapters 388, 395, 398, 446
property page generators 448
property pages
  adding to user interfaces 388, 389, 410, 449, 536
  building ODA user interfaces and 530, 538, 584
  building text item extensions and 446
  changing ODA driver settings and 511
  customizing 446, 448, 530
  enabling editing functionality for 445, 451, 540, 541
  enabling expression builder for 460
  extending report items and 418, 420, 429
  implementing read-only 454
  specifying categories for 409, 446, 449
  specifying properties for 398
property pages extension points 511, 535, 536, 578
Property property 60
property sheets 54
  See also Properties view
property structure objects 298, 299
property types 52, 456, 457
PropertyHandle class 298
propertyPages extension point 511, 530, 535, 536, 578
propertyPages.exsd 511
protocol 758
PS emitter 45
public classes 119, 217
public interfaces 217
publish 758
pUserParameters variable 614
pyramid charts 310, 323
  See also charts

Q

queries
  See also HQL statements; SQL statements
  accessing web service data sources and 185
  building charts and 151, 316, 648
  building ODA data sets and 544
  changing 98, 99
  defined 758
  developing ODA extensions for 510, 522, 554
  executing 525
  getting parameter values and 274
  preparing 388, 524
  retrieving data with 32, 43, 276
  returning unique set of values 648
  writing event handlers for 69, 74, 100
Query class 522, 524
query editor 759
query preparation extensions 388
query strings (request objects) 93
Query Text control 539
QueryImpl class 318
queryProperties parameter 185
queryText method 98
queryText parameter 185
queryTimeOut map key 185
quick start package 7
quotation mark characters. *See* double quotation mark character

R

radar chart example 345
radar chart extension
  building 650
  creating projects for 654
  creating user interface for 648, 661, 678
  debugging 705
  developing custom chart types for 652
  developing data series for 658
  implementing 649, 650, 661
  overview 647
  rendering 687, 689
  setting dependencies for 664
radar chart extension points 648, 650, 661, 684
radar chart schema definitions 652
radar charts 310, 647, 675, 689, 694
  See also charts
Radar class 686, 689
radar model package 684
RadarChart class 664
RadarModelLoader class 684
RadarSeriesAttributeComposite class 676, 677

RadarSeriesImpl class 658
RadarSeriesUIProvider class 674
range 759
RCP (defined) 729
  See also rich client platforms
RCP applications 264, 267
RCP Report Designer package 4
readLine method 526
read-only property pages 454
readSchemaFile method 504, 505
record counters 104, 105
  See also rows
referencing
  image files 266
  Java classes 112, 181
  objects 184
  report designs 29
refresh method 449
registering service objects 365
registry (plug-ins) 362, 466, 602
regular expressions 759
  See also JavaScript expressions
relational data sources 509
relational databases 510
  See also databases
relative paths 28, 29
release build 5
removing. *See* deleting
renaming report design files 327
render method 281, 286, 694, 698
render option classes 281
render option objects 223, 258, 281
render options 132, 223, 281, 285
render package 240, 255
render processes 86
render task processes 65, 66, 71, 78, 80
render tasks 221, 279
renderAxes method 703
renderer attribute 686
rendering classes 473, 498
rendering devices 242
rendering environments 219
rendering extension IDs 470
rendering extension points 466
rendering extensions
  adding emitters and 465
  building 482
  creating projects for 466, 494
  defined 759
  developing 465, 466, 494
  downloading sample code for 466
  downloading sample plug-ins for 385
  naming 470
  overview 465
  running 485
  setting dependencies for 469, 496
  setting encoding scheme for 475, 499
  specifying extension points for 471, 497
  testing 480, 506
  viewing output for 481, 486
rendering options 223, 281, 282, 285
  See also render option objects
rendering output 45, 64, 80, 219, 258
rendering phase (charts) 155, 156
renderLegendGraphic method 155, 686, 688, 689
renderMarker method 690
RenderOption class 258, 281, 285
renderOvalBackgrounds method 698, 699
renderPolys method 700
renderSeries method 155, 686, 693, 698
RenderTask objects 279
RenderTask processes 65, 66, 71, 78, 80
report components 48, 51, 59, 228, 229
  See also report elements; report items
report context objects. *See* reportContext objects
report creation phases 52
report debug configurations 196, 197
report debugging tool. *See* debugger
report descriptions 48
report design configuration objects 227, 291
report design definitions 58, 59
report design element API classes 260
report design element handles 230, 232
report design elements 128, 271
report design engine 43, 44, 227, 290
report design engine API 226, 227, 257
  See also report model API
report design engine API package 217, 230
report design engine classes 226, 227, 230, 291
report design engine configurations 227, 265

report design engine extensions 47
report design engine interface 238
report design engine objects 291
  See also report design engine
report design engine plug-ins 227, 228
report design engine tasks 227
report design environments. *See* BIRT; Eclipse
report design events 67
report design file parameter 20
report design files
  creating reports and 219, 257, 258, 270, 286
  defined 759
  designing reports and 48
  generating 43, 44, 49, 227
  loading 221
  opening 258, 270, 291
  renaming 327
  running 222, 279
  saving 227
  setting location of 29
  validating 44, 53
report design handles
  accessing properties and 229, 289
  accessing report items and 292
  creating report elements and 227
  getting 87, 88, 89
  opening designs and 228, 291, 292
report design interfaces 129
report design model 47, 51
report design model objects 226
report design objects 228
  See also report designs
report design perspective 7, 8
report design properties 59
report design tools 3, 43, 226, 227, 228
report designer applications 3
Report Designer Full Eclipse Install package 4, 7
report designer package 4
report designer UI extensions. *See* user interface extensions
report designers 43, 49, 219
  See also BIRT Report Designer; BIRT RCP Report Designer
report designs
  See also page layouts
  accessing 228, 289, 291
  accessing parameters in 273
  accessing properties for 229
  accessing ROM schemas for 53
  accessing structured content in 257
  adding charts to 323, 327, 345, 346
  adding master pages to 88
  adding report items to 292, 302
  changing 68, 87, 89, 262, 289
  creating 226, 290, 306
  customizing 217
  defined 725
  deploying to web servers 28
  generating XML files and 626
  getting properties for 270
  initializing 102
  installing report viewer and 29
  retrieving external data for 510, 522
  reusing 43, 48
  saving 290, 306, 327
  selecting ODA data sets and 538
  selecting report items for 46
  standardizing 48
  testing for parameters in 273
  validating 53, 57
  viewing CSV output in 488
  viewing report items in 73
  viewing XML output in 507
  writing event handlers for 67, 128, 135
report document files
  accessing contents 222, 271
  accessing structures 271
  building 44
  closing 271
  creating 80, 257, 278
  defined 760
  exporting data from 601
  generating reports from 65
  opening 221, 258, 270, 271
  overview 48
  rendering specific pages in 271
  writing to disk 222
report document model (BIRT). *See* ROM
report document objects 258, 270, 271
report documents. *See* documents; reports
report editor 760
Report element 52, 610
report element definitions 58
report element handles 230, 233, 236, 299
report element interfaces 129–135, 233

report element services 230
report elements
  See also specific type
  accessing 93, 257, 271, 289
  adding 68, 227
  applying styles to 52
  associating event handlers with 121–123
  changing 88, 227, 291
  creating 227, 302
  defined 760
  defining at run time 130
  dropping 90
  getting 271
  handling events for 68, 72, 73, 117, 123
  initializing 71
  loading property definitions of 227
  mapping 494
  operations unique to 230
  rendering 73
  reusing 49
  ROM specifications and 53
  setting properties for 130, 291, 299
  validating 44
report emitters. *See* emitters
report engine
  accessing external classes and 279
  accessing external data sources and 522
  adding emitters for 465
  adding fonts for 31
  cancelling running tasks and 288
  closing connections and 92
  configuring 220, 221, 258, 263, 264
  connecting to data sources and 258
  creating 220, 221, 258, 263, 267
  customizing output formats for 388, 465, 493
  debugging 188
  defined 712
  defining platform context for 264, 265, 266
  deploying 259
  designing reports and 221
  developing 219
  enabling reporting functionality for 272
  extending functionality of 47
  extracting data and 601
  firing events and 71, 77, 80
  generating HTML documents and 267
  generating report items and 388, 389
  generating reports and 43, 65, 258, 270
  getting information about 87
  getting output formats for 466, 602
  getting parameter values for 274, 277
  installing 19
  integrating with CSV output 466, 473, 486
  integrating with XML output 494, 497, 498, 602
  logging information about 269, 270
  opening design files and 270
  opening document files and 271
  overview 45
  rendering reports and 64, 465
  running plug-in fragments and 635
  running plug-ins and 362
  running reports and 64, 465
  setting BIRT home for 263
  setting global options for 220
  setting HTML rendering options for 266
  setting properties for 263, 264, 265
  setting up as stand-alone application 267
  setting up as web application 268
  shutting down 220, 258
  starting and stopping platforms and 267
  starting debugger and 196
  testing installations for 19, 20
report engine API 45, 118, 219, 220, 257
report engine API classes 113, 221, 223
report engine API interfaces 225
report engine api package 217, 223, 257
Report Engine API Reference Javadoc 257
report engine archive 19
report engine configuration objects 220, 263, 264, 265, 266
report engine constants 279
report engine content plug-in 466
report engine dataExtraction plug-in 601, 603, 609
report engine emitter plug-in 466, 497
report engine emitters extension 388
report engine emitters plug-in 465, 473

report engine emitters schema 388
report engine execution phases 86
report engine extension package 389
report engine objects 258, 263, 267
  See also report engine
report engine package 4
report engine plug-in 118, 496
report engine plug-in locations 220
report engine processing phases 65, 71, 80, 86
report engine report generation example 19
report engine services 42, 65, 219
report engine task types 197
report engine tasks 64–65, 84, 222
report example applications (charts) 345
report executable files 760
  See also report object executable files
report execution exceptions 194
report file types 47, 735
report files 43, 47–49
  See also specific type
report generation example 19
report generation services 219
report generators 49, 217
report item builder 439, 442, 458
report item definitions 58, 422
report item elements 59–60
  See also report items
report item emitters 388
report item events 71, 137
report item extension points 387, 398, 399
report item extension sample plug-in 21, 713
report item extensions
  adding descriptions for 396
  creating 387, 399, 416
  defined 761
  deploying 390, 412
  developing 47
  displaying reports and 45
  downloading sample plug-in for 385
  naming 360, 395, 396
  overview 387–389
  setting dependencies for 393
  testing 412, 422
report item generation extension point 388
report item generation schema 388
report item handles 235
report item icons 421
report item interfaces 235
report item menu builders 443
report item model 417
report item model extension point 387, 395, 396, 417
report item model schema 388
report item palette 388, 420
report item plug-ins 385, 389, 420
report item presentation extension point 388, 395, 397, 424
report item presentation schema 388
report item property editor 445, 448, 455
report item query extension point 388
report item query schema 388
report item rotated label sample plug-in. *See* rotated label plug-in
report item schema definitions 360, 387, 389, 399
report item UI designs 420, 429, 462
report item UI extension point 388, 395, 420
report item UI extension schemas 360, 361
report item UI property page 418, 420
report item UI property page adapters 388, 398
report item UI providers 421, 429, 435
report item UI schema 388
report items
  See also report elements
  accessing 257, 293, 294, 295
  adding 292, 302, 387
  binding data sets to 305
  binding to data sets 75, 76, 90, 456, 458
  binding to other report items 74
  changing model definitions for 456
  changing parameters and 74
  changing properties for 296, 297, 298, 445
  creating 47, 302
  customizing 47, 50, 389
  debugging 194
  defined 59, 760
  displaying 46, 73, 405
  examining 294
  extending functionality of 50, 53, 387, 416

report items *(continued)*
  generating 73, 388, 389
  getting display text for 395
  getting handles to 228, 229
  getting properties for 294
  handling events for 52, 71, 73, 77, 125, 137
  hiding 90
  instantiating 405
  naming 293
  overview 46–47
  providing interactivity for 421
  setting properties for 59, 94, 296, 297, 439
  setting rendering formats for 397
  setting styles for 58, 297, 396
  setting visibility of 421
  testing 428, 439
  viewing data bindings for 75
report layout editor. *See* layout editor
report layouts. *See* report designs; page layouts
report level events 135
report library files 48, 290, 747
  See also libraries
report model API 169, 226, 227
report model api package 217, 226, 230
report model plug-in 53
Report name property 614
report names 196, 610, 614, 616
report object model. *See* ROM
Report Object Model API Reference 257
report objects. *See* reports
report parameter collections 273, 274, 275
report parameter definitions 222
report parameter files 19
report parameters
  See also cascading parameters
  accessing 272, 273
  adding to URLs 32
  assigning default values to 66
  changing 67, 74
  converting string values for 273, 275
  creating report documents and 221
  defined 762
  displaying data and 90
  generating reports and 279
  getting attributes of 274
  getting data type of 274
  getting default values for 274
  getting display values for 87
  getting valid values for 274
  getting values 66, 87, 100
  handling events for 66, 72, 74
  iterating through 273
  overriding 20, 93
  providing input and 272
  selecting 66, 100
  setting chart axis title as 149
  setting display text for 132
  setting values 88, 132, 274, 275, 284
  testing for 273
  testing if grouped 273
  updating values for 274
  validating 67, 71, 222, 223
report projects. *See* projects
report rendering extensions. *See* rendering extensions
report rendering plug-ins 385
Report Script Virtual Machine 188
report sections 76, 765
report specifications. *See* report designs
report template files 48, 762
report template properties 229
report templates
  accessing structures 257, 290
  changing 290
  defined 771
  developing 226
  developing reports and 49
  saving designs as 43
report variables 85
report viewer
  See also web viewer
  accessing 26, 28, 30
  building 356
  changing context root for 28
  checking status of 31
  components of 42
  deploying 4, 25, 29
  developing for 219
  displaying reports and 28, 31
  displaying web pages and 28
  generating charts and 45
  generating reports from 258
  installing 26–28
  mapping to 28, 29
  overview 44

placing Java classes for 113
referencing report designs for 29
running on application servers 25, 26
setting context parameters for 29
starting 26
testing installations for 28
translating messages for 635, 640
troubleshooting Apache servers for 30
report viewer plug-in 44
report viewer servlets 31, 762
reportContext objects
chart elements and 149
event handlers and 86, 131
parameter values and 100
report designs and 89
report elements and 88
ReportDesign element 59, 128
ReportDesignHandle class 227, 229, 290, 292
ReportDesignHandle objects 227, 228, 229, 291
ReportElement element 58
See also report elements
ReportElementHandle class 230, 233
ReportElementHandle objects 230
ReportEngine class 221, 258, 263, 272
ReportEngine objects 220, 221, 258, 263, 267
ReportEventAdapter class 128
See also adapter classes
reporting applications. *See* applications
reporting platform. *See* BIRT
ReportItem element 58, 59
reportitem plug-in (chart) 389
reportitem plug-in (rotatedlabel) 385, 389
See also rotated label plug-in
reportItem.exsd 388
ReportItemFactory class 405
reportItemFigureUI element 395
reportitemGeneration extension point 388
reportitemGeneration schema definition 388
reportitemGeneration.exsd 388
ReportItemHandle class 235
reportItemImageUI element 395
ReportItemLabelProvider class 405
reportItemLabelUI element 395
reportItemLabelUI property 421
reportItemModel extension point 387, 395, 396, 417
reportItemModel schema definition 388
reportitemPresentation extension point 388, 395, 397, 424
reportitemPresentation schema definition 388
reportitemPresentation.exsd 388
ReportItemPresentationBase class 405
reportitemQuery extension point 388
reportitemQuery schema definition 388
reportitemQuery.exsd 388
reportitemUI extension point 388, 395, 420
reportitemUI extension schema 360, 361
reportitemUI schema definition 388
reportitemUI.exsd 360, 361, 388
ReportletNoCSS value 20
ReportName element 610
reportName variable 614
ReportParameterConverter class 273, 275
reports
accessing 28, 288
changing 72
controlling content of 49
creating 63, 217, 258, 279
customizing 85, 262
debugging 187, 197, 198–214
defined 759
deploying 25, 118, 183, 282
designing. *See* designing reports; designs
developing 49
displaying 28, 31, 32, 43, 219
generating 49, 73, 227, 262, 270, 286
getting number of pages in 221
getting styles for 294
hiding content in 90, 471
paging through 80
previewing 44, 65, 219
rendering output for 45, 64, 80, 219, 258, 465
rendering specific pages for 221
running 64, 86, 93, 118, 465
saving 219
selecting language for 11
setting styles for 52, 57
testing report viewer for 28
viewing metadata for 53

reports *(continued)*
  viewing schema definitions for 53
  viewing structure of 194
request objects (HTTP servlet) 87, 93, 131
reserved words. *See* keywords
resizing charts 172
resizing web pages 283
resource bundles 364, 368, 635
Resource Explorer 48
resource factory 34
resource files 33, 264, 291, 763
Resource files property type 264, 265
resource keys 763
resource locations 383
resources
  defined 763
  getting URLs for 87
  HTML emitters and 266
  JNDI connections and 34
  module handles and 229
  plug-in fragments and 635
  updating report designers and 12, 384
  user sessions and 228
result sets
  See also data sets; queries
  building charts and 151, 311, 648
  defined 763
  extracting data and 609, 617, 619
  getting column names in 545
  getting metadata for 546
  getting number of columns in 97
  querying CSV data sources and 522, 523, 527
  querying Hibernate data sources and 568, 572
  rendering reports and 471
  returning 510
  testing ODA drivers and 511
ResultSet class 523, 527, 564, 572
ResultSetMetaData class 523, 528, 564
rich client designer. *See* BIRT RCP Report Designer
rich client platforms 43, 729
right outer joins. *See* outer joins
right to left generation 282
riser types (charts) 151
rollback method 567
ROM 47, 51, 761
ROM API Reference Javadoc 257
ROM definition file. *See* rom.def
ROM Definitions Reference 51
ROM element definitions 55
ROM element design interfaces 123
ROM element handles 289
ROM element instance interfaces 123
ROM element interface names 123
ROM elements
  accessing 289
  adding 58, 59, 303
  applying styles to 52
  changing 54
  defined 761
  defining event handlers for 52
  defining executable code for 60
  defining properties and 52, 53, 54
  overview 51, 58, 60
  viewing metadata definitions for 53–57
ROM report item elements 59–60, 360
  See also report items
ROM schemas 52, 53, 762
  See also XML schemas
ROM specification 47, 51, 761
ROM types 52
rom.def 53, 54, 761
rotated label manifest file 363
rotated label plug-in 365, 385, 404
  See also rotated label report item extension
rotated label report item extension
  creating projects for 390–392
  deploying 390
  developing 390, 404–412
  downloading BIRT plug-ins for 390
  downloading source code for 385, 390
  implementing 389, 412
  launching 412
  loading BIRT plug-ins for 394
  running 381
  setting dependencies for 393
  specifying content of 399
  specifying extension points for 395, 398, 399
  specifying run-time packages for 394
rotated label report item extension points 377
rotated text builder 439, 442, 458

rotated text extension. *See* rotated label report item extension
rotated text images 404, 424, 425, 429
rotated text item classes 419, 424
rotated text item extension 417
rotated text items
  adding to user interfaces 416
  binding to data sets 456
  creating 389, 406
  rendering as images 404, 424, 425
  setting rotation angles for 428, 443
  testing 428
rotated text menu builder 443
rotated text property editor 445, 448, 455
rotated text UI designs 420, 429, 462
rotated text UI providers 421, 429, 435
RotatedLabel application 413
rotatedlabel plug-in 365, 385, 404
  See also rotated label report item extension
RotatedLabelCategoryProviderFactory class 409
RotatedLabelGeneralPage class 410
RotatedLabelItemFactoryImpl class 405
RotatedLabelPlugin class 365
RotatedLabelPresentationImpl class 405
RotatedLabelUI class 405
RotatedText property 420, 424, 456
RotatedTextBuilder class 439, 440
RotatedTextCustomPage class 454
RotatedTextEditor class 440
RotatedTextFigure class 436
RotatedTextFigureUI class 435
RotatedTextGeneralPage class 451, 454
RotatedTextItem class 419
RotatedTextItemFactory class 417
RotatedTextLabelUI class 421
RotatedTextMenuBuilder class 443
RotatedTextPageGenerator class 449
RotatedTextPageGeneratorFactory class 448
RotatedTextPresentationImpl class 424, 457
rotateImage method 406, 408
rotation angle properties 446, 451
rotation angles 428, 443
rotation rendering algorithm 425
rotationAngle property 396, 418
Row element 612
row execution sequence (events) 77
row keys 764
  See also cross-tab reports
row numbers 96
row objects 96, 127, 134
  See also rows
rows
  accessing columns in 96
  building 75, 77, 135, 178
  concatenating column values and 75
  counting 84
  defined 764
  exporting data and 612
  generating at run time 130
  getting information for 134
  getting position of 527
  grouping 79
  incrementing cursors for 527
  iterating through 182
  retrieving 70, 126, 127, 174
  returning from Java objects 182
  returning result sets and 522
  setting page breaks for 76
  writing to CSV files and 476, 477, 478
.rptdesign files. *See* report design files
.rptdocument files. *See* report document files
.rptlibrary files. *See* report library files
.rpttemplate files. *See* report template files
RTF converter 776
__rtl parameter
  See also bidirectional text
run 764
run method 281, 286
run requests 93
run servlet 32, 65
run time 764
RunAndRenderTask events 72, 74, 80
RunAndRenderTask objects 278
RunAndRenderTask processes 64, 66, 71, 72
runnable variable 284
running reports 64, 86, 93, 118, 465
running tasks, cancelling 288
RunTask events 72, 74, 80
RunTask objects 278
RunTask processes 64, 66, 71, 78
run-time archives (plug-ins) 360, 371

run-time connections 99
run-time drivers 522
run-time engine 482
run-time environments 362, 384
run-time extensions 684
run-time instance (PDE Workbench) 368
run-time libraries 378
Runtime page (PDE Editor) 369, 371
run-time properties 130
runtime tag 636
run-time workbench 380, 381, 413

S

sac.jar 261
SalesReport application 345
sample charting API plug-ins 340–346
sample charting applications 16, 307, 340
sample data 322, 325, 671
Sample Data containment reference (charts) 146, 147
sample database 18
 See also Classic Models sample database
sample event handlers 107, 135, 343
sample ODA extensions 509, 511
sample plug-in extensions 384
sample plug-in fragment 385, 636, 640
Samples archive 21
Samples package 5, 21
save method 306
saveAs method 306
savePage method 547, 592
saving
 event handlers 113
 images 266
 report components 44
 report designs 227, 290, 306, 327
 reports 219
Scalable Vector Graphics. *See* svg attribute
scalar data types 520
scalar parameters 272, 273
scale (charts) 693
scatter charts 307, 310
 See also charts
schema directory 360
schema element 360
schema files 360, 498, 504, 652
schemas
 charting extensions 652
 connection profiles 511
 data extraction extension and 602
 defined 764
 design specifications and 47
 displaying as HTML 374
 displaying property annotations in 376
 mapping report elements to 494
 ODA driver extensions 510
 ODA driver sample plug-ins 564
 ODA UI extensions 510
 opening 373, 374
 plug-in extensions 359, 360, 373
 property definitions and 53
 property pages 511
 report engine emitters 466, 499
 report item extensions 360, 387, 389, 399
 ROM specifications and 52, 53
 rotated label sample plug-in 395
 validating designs and 53
scope 765
script API library 261
Script attribute (charts) 147
script debugging tasks 188
script editor 83, 189, 193, 765
script package 240, 255
Script page. *See* script editor
Script tab 83
Script Virtual Machine 188
scriptable objects (external) 146
scriptapi.jar 118, 261
ScriptCharts class 343
ScriptDataSetHandle class 304
ScriptDataSourceHandle class 303
scripted data set elements 60, 127
scripted data set events 70
scripted data set interface 127
scripted data sets
 See also data sets
 accessing Java classes for 181
 accessing ODA data sources for 304
 accessing parameters for 183
 closing 70, 128, 174, 182
 creating 173, 176
 getting parameter values for 101
 handling events for 127
 initializing 178, 181

initializing variables for 174
naming 176
opening 70, 127
retrieving data for 70, 182
setting parameter values for 101
writing event handlers for 68, 70
scripted data source elements 60, 127
scripted data source events 70
scripted data source interface 126
scripted data sources
See also data sources
accessing Java classes for 182
closing 69, 127, 174
creating 173, 174, 175
handling events for 126
initializing 174
naming 175
opening 69, 127
retrieving data and 70, 173
tutorial for 175–180
writing event handlers for 68, 70
ScriptedDataSet elements 60, 127
ScriptedDataSet interface 127
ScriptedDataSource elements 60, 127
ScriptedDataSource interface 126
ScriptedDataSourceAdapter class 127
ScriptedDataSrc.rptdesign 175
ScriptHandler class 240
scripting 52, 63, 217, 280
Scripting configuration property type 265
scripting context 222
scripting examples 107, 135
scripting languages 63, 765
scripting specifications 83
ScriptingJava.html 114
scriptlib directory 113
scripts
accessing Java classes and 112, 113, 121
accessing methods for 114
additional references for 114
autotext items and 85
binding data and 76
building charts and 145, 156, 161, 169, 172, 255, 320
building report designs and 63
changing connection properties and 99
changing queries and 99
concatenating code for 320
customizing reports and 262
debugging 193
determining execution sequence for 101–103
displaying 193
grouping data and 79
importing Java packages and 112, 181
previewing 96
providing external values for 265
referencing Java classes and 112, 181
retrieving data and 70
returning parameter values and 100
setting application context and 92
writing event handlers and 63, 64, 83, 117, 143
ScriptViewer charting example 343
scrollable properties list 95
SDK framework 3
SDK package 351, 352, 765
SDK software 3, 4
SDO Runtime component 4
search paths 31, 266
sections 76, 765
security 49
security identifiers 184
select 765
select painter. *See* query editor
SELECT statements. *See* SQL statements
Selected column names property 614
selectedColumnNames variable 614
selection formulas. *See* parameters
SelectionAdapter method 586
SelectionListener interface 677
semantic validators 57
SemanticValidator element 57
sequential files. *See* flat files
serializable objects 132
serialize method 306
series
See also charts
adding 316
associating data with 318
binding to charts 311
binding to data sets 150
building run-time instances of 148, 151, 152, 156, 311
calculating values for 693

series *(continued)*
  changing 159
  comparing multiple values in 647
  creating 315, 316, 658
  defined 310, 766
  determining type 152
  getting run-time 151
  grouping 341, 688
  initializing 150
  labeling data points in 702
  nesting 154
  preserving changes to 670
  providing hints for 159, 160
  rendering 155, 158, 648
  retrieving run-time 161
  returning unique values for 648
  scaling 161
  setting bar size for 148
  setting type 240, 253, 653
  setting unit spacing for 148
  stacking 658
  validating data points for 691
  z-ordering 688
series composite extension points 673
series definition objects 151, 311, 316
series definitions 319
series identifier 152, 159
Series interface 658
series objects 152
series palette 684
series renderers 155, 156, 158
series sample plug-ins 341
Series Thickness attribute (charts) 147, 148
series types 316, 323, 653, 654
seriescomposites plug-in 648
SeriesDefinition objects 316, 319
SeriesDefinitionImpl interface 316
SeriesDefinitions objects 311
SeriesImpl class 240, 317, 658, 659
SeriesPalette objects 316
seriesType attribute 673
seriesUIProvider attribute 673
server connections 32, 33
servers 25, 26, 155, 649
service applications 365
Service Data Objects component 4
service objects 365
service references 365
service registry 365
services 40, 44, 65, 365
servlet context 264
ServletContext class 266
servlets 766
session handles 228, 291, 292
session parameters 227
SessionFactory objects 574
SessionFactory operations 555, 565
SessionHandle class 228, 306
SessionHandle objects 227, 228, 291, 292
set method 660
setAbsolute method 297
setActionHandler method 284
setAttribute method 91
setBaseImageURL method 282
setBaseURL method 283
setBeforeOpen method 230
setBIRTHome method 263
setBlock method 147
setChartInstance method 146
setColorByCategory method 171
setDataSet method 303, 305
setDataSource method 304
setDescription method 147
setDimension method 147, 148, 171
setEmbeddable method 282
setEmitterConfiguration method 264, 266, 267
setEnableAgentStyleEngine method 283
setExtensionProperty method 134
setExternalContext method 146
setGlobalVariable method 84, 88, 132
setGridColumnCount method 147
setHeight method 172
setHideGridlines method 284
setHorizontal method 171
setHtmlPagination method 283
setHtmlRtLFlag method 282
setHtmlTitle method 282
setImageDirectory method 282
setImageHandler method 266, 282
setInitialProperties method 587, 588
setInteractivity method 147
setLayoutPreference method 283
setLogConfig method 264, 269
setLogFile method 269
setLogger method 146
setLogMaxLogBackupIndex method 269

setLogRollingSize method 269
setMasterPageContent method 283
setName method 293
setOfficeVersion method 284
setOption method 285
setOSGiArguments method 264
setOSGiConfig method 264
setOutputFileName method 285
setOutputMasterPageMargins method 283
setOutputType method 171
setPageFooterFloatFlag method 283
setPageNumber method 286
setPageRange method 286
setPageVariable method 88
setParameterDisplayText method 132
setParameterValue method 88, 132, 274, 279
setParameterValues method 279
setPersistentGlobalVariable method 85, 88, 90, 132, 133, 140
setPlatformContext method 264, 265, 266
setProperty method 265
setQueryText method 134
setRenderOption method 282
setResourceLocator method 265
setResourcePath method 265
setResultSetMetaData method 547, 593
setSampleData method 147
setScript method 147, 320
setSeriesThickness method 147
setStatusHandler method 265
setSubType method 147, 323
setSupportedImageFormats method 282
setTempDir method 265, 266
setThreadContextClassLoader property 519
settings. *See* properties
setType method 147, 323
setULocale method 146
setUnits method 147, 313
setUnitSpacing method 148
setupConfigLocation method 585
setupFileLocation method 541
setURI method 296
setVersion method 147
setWidth method 172
setWrappingText method 285
shared styles 297
SharedStyleHandle class 297
Show extension point description setting 373, 374
shutdown method 267
Simple Object Access Protocol. *See* SOAP
simple properties 294
singleton pattern 755
  See also object-oriented programming
singleton property 398, 447
site.xml 382
slot definitions (ROM) 56
Slot element 56
slot handles 293, 294, 305
Slot property 56, 60
slots 52, 56, 59, 293, 766
SOAP (defined) 766
SOAP messages 184
software 381, 384
Software Development Kit 3, 765
  See also JDK software; SDK package
Software Development Kit package 4
sort 767
sort conditions 89
sort fields. *See* sort keys
sort keys 767
sort order 299
sort-and-group-by fields. *See* group keys
sorting data 46, 89, 151, 311
source builds 378
source code
  accessing data sources and 173
  accessing Java 359
  accessing report parameters and 272
  accessing reporting functionality and 228
  accessing sample 385
  adding breakpoints to 193, 194
  adding event handlers and 63, 64, 79, 83, 117
  adding to designs 262
  building data extraction extensions and 602, 626
  building rotated label plug-in and 390
  changing 349, 356
  changing connection properties and 99
  charting API examples and 340
  charting extensions and 650, 652
  checking for errors in 104

source code *(continued)*
compiling 350, 380
controlling report creation and 63
creating custom designers and 227
debugging plug-ins and 390
defining executable 60
deploying applications and 262
developing applications and 217
developing ODA driver extensions and 511, 513, 555
developing plug-in fragments and 636
developing rendering extensions and 466
downloading 350, 353
extending BIRT functionality and 52
getting parameter values and 275
grouping data and 79
importing 5, 350
installing 21–22
loading 362, 746
logging information and 269, 270
opening design files and 270
retrieving data and 70
running plug-ins and 362, 381, 390
setting up workspaces for 354
tracking script execution and 101, 102
validating 189, 192
viewing charts and 255, 358
Source Code archive file. *See* source code archives
source code archives 21, 353
source code editors 29
source code modules 228
source data. *See* data sources
source extension point 378
source files (Java) 119
source parameter 688
spider charts. *See* radar charts
spider subchart type 703
spinner elements 678, 681
splitQueryText method 524
spreadsheet converter 732, 756
spreadsheets 284
SQL (defined) 767
SQL databases. *See* MySQL databases
SQL language 767
SQL statements
See also queries
changing 98, 99
CSV data sources and 522, 524
defined 767
getting 98, 134
Hibernate data sources and 510
JDBC data sources and 32
sqlResultSet objects 151
src attribute 267
stable build 5
stack frame 191
stand-alone applications 49
accessing engine plug-ins for 265
configuring BIRT home for 263
configuring report engine as 267
creating charts and 239
generating HTML documents and 266
stand-alone environments 264
stand-alone report engine 219
STANDALONE variable 309
stand-alone web pages 32
Standard Widget Toolkit. *See* SWT applications
start method 475, 499
starting
BIRT RCP Report Designer 9, 10
BIRT Report Designer 8
debugger 196
Eclipse 10, 11
Eclipse Workbench 380
report viewer 26
startLabel method 501
startRow method 477
startTable method 476
startText method 478, 501
startup method 267
start-up parameters 188
Statement class 564, 568
statements 768
static constants 112
static data 150, 310
static variables 768
See also dynamic variables; variables
Status handling property type 265
STATUS_CANCELLED value 288
STATUS_FAILED value 288
STATUS_NOT_STARTED value 288
STATUS_RUNNING value 288
STATUS_SUCCEEDED value 288
stock charts 307, 310, 648
See also charts

StockDataSet interface 311
StockReport application 345
streams. *See* input streams; output streams
string 768
String class 112
String data type 768
string expressions 768
string properties 297
strings
  changing property type for 456, 457
  concatenating 320
  conditionalizing 90
  connecting to data sources and 184
  converting 273
  converting to locale-independent formats 273
  converting XML documents to 620
  exporting data and 612, 617
  retrieving report parameters and 93
  setting parameter values and 274, 275
  translating 635
  writing to CSV files and 473
Structure element 57
structure property objects 298
structured content 768
Structured Query Language. *See* SQL
structured report items 294
StructureFactory class 299
StructureHandle class 236
structures 57
StructureSource class 688
stubs 118, 124
style attributes 53, 54, 296
style components 52, 57
style definitions 57
  See also style attributes; styles
Style element 57
style properties 228
style property 53, 57
style sheets 52, 260, 261, 713
StyleChartViewer application 342
StyleHandle class 297
StyleProcessor charting example 342
StyleProcessor class 342
StyleProperty property 60
styles
  accessing 229
  adding report items and 58, 297, 396
  changing 88, 90
  creating charts and 342
  creating reports and 52
  defined 769
  generating HTML output and 283
  getting 147, 294, 297
  reusing 49
  selecting 95
  setting attributes for 53, 54, 296
  setting properties for 60
styles property 90
Sub Type attribute (charts) 147
subclasses 769
  See also descendant classes
subreports 769
subroutines. *See* procedures
subType property 397
summary data. *See* aggregate values
Superclass Selection dialog 119
superclasses 119, 769
  See also ancestor classes
supportedFormats element 397
supportedImageFormats property 471
supportsMultipleResultSets method 568
supportsTransposition method 662
suspended threads 191, 192
SVG device renderers 649
SVG formats 155, 171, 282, 687
SvgInteractivityViewer application 342
Swing applications 334, 341
Swing device renderers 649
Swing display servers 155
Swing formats 687
Swing graphics API 424
Swing renderers 155
SwingChartViewersSelector application 344
SwingGraphicsUtil class 430
SwingInteractivityViewer application 342
SwingLiveChartViewer application 344
SWT applications 341, 343
SWT device renderers 155, 649
SWT display servers 155
SWT formats 687
SWTchartViewerSelector application 344
SwtInteractivityViewer application 342
syntax (programming languages) 769
syntax checker 188

syntax conventions (documentation) xxvii
syntax errors 188, 189
system variables 19

T

-t command line option 20
table cells. *See* cells
table elements 302, 770
  See also tables
table execution sequence (events) 78–79
table footers 75
table items 58, 59, 73
  See also tables
table of contents 286, 771
table of contents markers 221
tables
  See also table elements; table items
  adding columns to 179
  binding to data sets 75, 76
  building data rows for 77
  counting rows in 84
  creating 59
  defined 770
  filtering data in 89
  generating HTML content and 283
  getting column information for 96, 97
  grouping data for 78
  handling events for 73, 75, 77, 78, 139
  rendering CSV output and 473, 476, 477
  setting page breaks for 76
  sorting data in 89
tabs 769
  See also page
tab-separated values. *See* TSV formats
tabular layouts 59
tags 771
  See also elements
target encoding parameter 19
targetNamespace attribute 658
task classes 278
task objects 278
task processes 64–65, 84
task types (report engine) 197
tasks 132, 222, 288
template files 48, 762
template properties 229
templates
  accessing structures 257, 290
  changing 290
  defined 771
  developing 226
  developing reports and 49
  saving designs as 43
Temporary file location property type 265
temporary files 220, 221, 265
temporary folders 197
temporary images 266
test.rptdesign 28
testCharts.chart 345
testing
  BIRT installations 8, 9, 350
  BIRT projects 356
  chart attributes 247
  charting applications 309
  context menus 445
  CSV emitter 480, 488
  CSV ODA UI plug-in 548
  data extraction extension 620, 626
  Demo Database installations 18
  DTP Integration installation 17
  Hibernate ODA UI plug-in 594–599
  ODA drivers 511
  plug-in extensions 380–381
  plug-in fragments 637, 643
  plug-ins 360, 368
  report engine installations 19, 20
  report item extensions 412, 422
  report viewer installations 28
  rotated text items 428, 439
  source code package 21
  Web Tools Integration archive 23
  XML emitter 506
text 90, 194, 294, 675
  See also text elements; text items
text controls 536, 539
text elements 129, 389, 772
text file data sources. *See* text files
text files
  See also CSV files
  accessing data in 509
  reading from 384, 522, 526
  rendering output to 465
  tracking script execution in 101, 103
text item builders 439, 442, 458
text item design interface 129

text item property editor 445, 448, 455
text items
  See also TextItem objects
  binding to data sets 456
  changing model definitions for 456
  changing properties for 445
  changing values for 456
  handling events for 68
  rendering as images 404, 424
  rotating 389, 406, 428, 443
  testing 428, 439
  writing to CSV files 474
  writing to XML files 498, 501
text metrics 155
text objects. *See* text
text property 419
text strings. *See* strings
TEXT_LITERAL value 161
TextDataSet interface 311
TextRenderEvent event 702
themes 228, 229, 772
  See also styles
this object 93, 94, 95
thread context class loader 519
threads 191, 192
tick 772
tick interval 772
time data type 724
time values 675, 724, 773
time zones 88
Timestamp data type 773
title area (charts) 147, 158
title tag 282
tmpdir variable 266
TOCNode class 286
Tomcat manager accounts 30
Tomcat server connections 32, 33
Tomcat servers 25, 26, 30, 44
toolbars 32, 773
top-level report items 73
transactions 567
transform package 620
transient files. *See* temporary files
translations. *See* locales; localization
translators. *See* converters
troubleshooting installation 9–10
TrueType fonts 31
TSV formats 466
tube charts 310, 323
  See also charts
tutorials
  creating event handlers 103–107
  creating scripted data sources 175–180
  debugging reports 198–214
.txt files. *See* text files
type. *See* data types
type attribute 519, 636
Type attribute (charts) 147
type casting 152
Type elements 60
type package 240, 253
type parameter 688
type property 397, 398, 447
type.xsd 653
TypeFactory class 659
types plug-in 648, 661
typographic conventions (documentation) xxvii

U

UI extensions. *See* user interface extensions
ui plug-ins 194, 510, 511, 512, 648, 661
  See also user interfaces
UI providers 421, 429, 435
UiPlugin class 538
uisheets plug-in 649
ULocale objects 146
Unicode encoding 773
Unicode standard 773
Uniform Resource Locators. *See* URLs
unique values 648
unit of measure 228
Units attribute (charts) 147
universal hyperlinks. *See* hyperlinks
Universal Resource Identifiers. *See* URIs
UNIX systems 10, 15, 19
unpacking BIRT archives 8, 9
unpaginated HTML formats 278
unsetDimension method 148
unsetGridColumn method 148
unsetSeriesThickness method 148
unsetVersion method 148
unsupported data sources 46
update configuration manager 381, 382
Update Manager 11
update manifest file 382
Update Site Editor 381

update site maps 383
Update Site Project wizard 383
update sites 381, 382–383
updates 381, 382, 384
updating
  ODA data sets 539, 547
  plug-ins 384
  report designers 11–13
  source code 349
  user interfaces 458
upgrades 11
uploading update sites 382
URIs 294, 296, 774
URL parameters 342
URLClassLoader objects 519
URLs
  accessing ODA data sources and 99
  accessing reports and 28, 32, 93
  adding report parameters to 32
  defined 773
  generating HTML documents and 266, 283
  generating reports and 258
  getting resource 87
  image files and 267
  JNDI connections and 32, 34
  localized reports and 11
  Mozilla Rhino specification and 83
  report viewer and 28, 44
  running reports and 611
Usage property 435
user IDs 93, 100
user interface classes 194
user interface extensions
  custom chart types and 648, 673
  custom report items and 360, 388, 395, 420
  downloading sample plug-ins for 385
  ODA data drivers 510, 511, 512
  property page adapters and 388, 395, 398
user interface providers 421, 429, 435
user interface schema definitions
  property page adapters 388
  report items 360, 361, 388
user interfaces
  accessing ODA data sources and 46
  adding connection profiles to 510, 535
  adding data sources to 530
  adding property pages to 388, 389, 410
  adding report items to 389
  adding rotated text items 416
  adding slots to 56
  building context menus for 443–445
  changing property values and 445
  charting data and 648, 661, 678
  creating chart reports and 47
  creating custom report designer and 49, 227
  creating ODA driver. *See* ODA user interface
  customizing 274
  displaying images in 429
  displaying output formats and 471
  displaying properties in 56, 439
  exporting data and 601, 611, 622
  generating reports and 263
  HQL queries and 585
  integrating with Eclipse 4
  mapping report items to 395
  registering 395
  replicating parameter groups and 273
  selecting ODA data sources and 510
  updating 458
user names 99
User parameters property 614
user sessions 227, 228, 291
user-supplied values. *See* input
util package 240, 256

V

validate events 66, 67, 72
validateDataSetCount method 691
validateOpenConnection method 524
validateParameters method 222, 223
validateQueryText method 524
validateSeriesBindingType method 673, 675
validating report designs 53, 57
validating report output 53
validationIndex method 674
validator classes 57
validator definitions (ROM) 57
value axes. *See* axes values
value series 319
  See also data series
values
  See also data

adding page-specific 85
calculating specific 75
changing dynamically 456
concatenating 75
defined 774
displaying external 272
getting default 274
getting parameter 87, 100
getting property 270
overriding parameter 20
populating property 439
replacing null 152
retrieving 96
returning unique 648
selecting 67, 100, 276
setting default 66, 272
setting parameter 88, 132, 279
setting property 297
setting user-supplied 272
storing 91

var identifier 84

Varchar data type 774

variables
See also specific type
building report pages and 68
charting data and 693, 698
compiling and 350
concatenating values and 75
data extraction and 614
defined 774
deleting 86
engine processes and 90
external context objects and 149
getting 87, 228
initializing 174
inspecting 190, 192, 194
resetting 85
retrieving parameter values and 100
scripting and 84, 85, 86, 93
setting 88

Variables view 192

verifyQuery method 590

Version attribute (charts) 147, 148

version attribute (plug-in fragments) 636

view time 775

viewer. *See* report viewer; web viewer

viewer applications (charts) 342

Viewer charting examples 343

viewer package 343

viewing
breakpoints 192, 194
cascading report parameters 277
charts 659, 671
CSV output 481, 486
data 276
data bindings 75
error messages 104, 107
event handlers 64, 83
extension point schemas 374
HTML output 32, 44
images 429
manifest header settings 369
PDF documents 31, 32, 288
plug-in project settings 369
property annotations 376
report items 46, 73
reports 28, 31, 32, 43, 219
scripts 193
update site content 383
XML output 507, 609

views 369, 730, 774

viewservlets.jar 356

visibility property 56, 90

visitor objects 474, 498

visual components 46, 51, 59
See also report items

visual elements 58

W

.war files
application deployment and 264, 266
defined 775
platform context and 264
report viewer deployment 29, 30
report viewer installations 26, 27, 28
Tomcat servers and 26
report engine installations 19

web applications
See also applications
accessing report viewer for 28
adding charting capabilities to 16
changing source code and 349
configuring BIRT home for 264
configuring report engine as 268
generating HTML documents and 266
generating reports and 49, 219
installing report viewer for 26, 27, 28

web archive files. *See* .war files

web-based reports 65, 219
web browsers 31, 283, 288, 609
web pages
  See also HTML reports
  defined 775
  displaying 28, 32
  resizing 283
  update sites and 383
web resources 383
web servers 25, 775
web services 775
web services connections 185
web services data sets 184
web services data sources 184–186
web sites xix
web standards 776
Web Tools Integration archive 22
Web Tools Integration package 5, 22, 23
Web Tools Platform 4
  See also WTP plug-ins
Web Tools Platform extension 16
web viewer
  See also report viewer; web browsers
  building JAR files for 350
  changing source code and 349, 356
  extracting data and 601
  generating reports and 65
  report engine processes and 65
  viewing XML schema definitions in 374
Web Viewer application. *See* web viewer
web.xml 26
webapps directory 26
WebLogic servers 25
WebSphere servers 25
WebViewerExample folder 27
well-formed XML 776
widgetSelected method 681
windows 750
Windows systems 10, 11, 15, 19
Wizard class 537
wizardPageClass property 585
wizards 366, 509, 577
  See also ODA wizards
wizards package 538, 539
workbench projects 368
  See also Eclipse Workbench; projects
workspace 13, 350, 354, 776
workspace directory 12, 354
World Wide Web Consortium (W3C) 776
WrappedStructureSource class 688
wrapping Java objects 173
writer objects 474, 499
WTP plug-ins 4, 5, 22, 23
WYSIWYG 429

X

x-axis labels 161, 239
x-axis properties 162
x-axis titles 161
XLS emitter 45
xmi.jar 260
XML (defined) 733
XML converter 776
XML data extraction extension
  creating projects for 603–605
  developing 602, 609
  downloading source code for 602
  implementing 610
  overview 601, 609
  running 620
  setting dependencies for 605
  testing 620, 626
XML data extraction extension points 607
XML data extraction interfaces 609
XML data extraction schema definitions 602
XML document conversions 620
XML documents 610, 614, 617, 776
XML elements 51, 52, 730
  See also ROM elements
XML emitter 498, 506
  See also XML report rendering extension
XML emitter logging levels 499
xml emitter plug-in 465, 498
XML emitter properties file 494
XML extension point schema definitions. *See* XML schemas
XML file output streams 499, 609
XML files 48, 51, 494
XML formats 465, 466, 494, 609
XML PATH language 777
xml plug-in 610
XML rendering option 504
XML report rendering extension
  creating projects for 494

developing 494, 498
downloading source code for 385, 466
overview 465, 497
setting dependencies for 496
setting encoding scheme for 499
specifying extension points for 497
testing 506
XML report rendering plug-in 385
XML schema files 360, 498, 504
XML Schema language 53
XML schemas
connection profiles 511
data extraction extension and 602
defined 764
design specifications and 47
displaying as HTML 374
displaying property annotations in 376
mapping report elements to 494
ODA driver extensions 510
ODA driver sample plug-ins 564
ODA UI extensions 510
opening 373, 374
plug-in extensions 359, 360, 373
property definitions and 53
property pages 511
report engine emitters 466, 499
report item extensions 360, 387, 389, 399
ROM specifications and 52, 53
rotated label sample plug-in 395
validating designs and 53
XML streams 173
XML writer 497, 498, 499, 500, 504
XMLDataExtractionImpl class 610, 614
.xmlemitter properties file 494
XMLFileWriter class 498, 504
XMLPlugin class 498
XMLRenderOption class 498, 504
XMLReportEmitter class 498, 500
XMLReportEmitter method 498
XMLSpy utility 53
XMLTags class 498, 503
XPath expressions 777
xsd directory 652
XSD files 652
XSD SDK package 650
x-series items 159
See also axes values; data series

Y

y-axis labels 161
y-axis properties 162
y-axis titles 161
y-series items 159, 161, 317
See also axes values; data series

Z

.zip files. *See* archive files
z-ordering (charts) 688