

Installing and Deploying Eclipse BIRT

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

© 1995 - 2015 by Actuate Corporation. All rights reserved. Printed in the United States of America.

Contains information proprietary to:

Actuate Corporation, 951 Mariners Island Boulevard, San Mateo, CA 94404

www.opentext.com

www.actuate.com

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

Actuate Corporation trademarks and registered trademarks include:

Actuate, ActuateOne, the Actuate logo, Archived Data Analytics, BIRT, BIRT 360, BIRT Analytics, The BIRT Company, BIRT Content Services, BIRT Data Analyzer, BIRT for Statements, BIRT iHub, BIRT Metrics Management, BIRT Performance Analytics, Collaborative Reporting Architecture, e.Analysis, e.Report, e.Reporting, e.Spreadsheet, Encyclopedia, Interactive Viewing, OnPerformance, The people behind BIRT, Performancesoft, Performancesoft Track, Performancesoft Views, Report Encyclopedia, Reportlet, X2BIRT, and XML reports.

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

Mark Adler and Jean-loup Gailly (www.zlib.net): zlib. Adobe Systems Incorporated: Flash Player, Source Sans Pro font. Amazon Web Services, Incorporated: Amazon Web Services SDK. Apache Software Foundation (www.apache.org): Ant, Axis, Axis2, Batik, Batik SVG library, Commons Command Line Interface (CLI), Commons Codec, Commons Lang, Commons Math, Crimson, Derby, Hive driver for Hadoop, Kafka, log4j, Pluto, POI ooxml and ooxml-schema, Portlet, Shindig, Struts, Thrift, Tomcat, Velocity, Xalan, Xerces, Xerces2 Java Parser, Xerces-C++ XML Parser, and XML Beans. Daniel Bruce (www.entypo.com): Entypo Pictogram Suite. Castor (www.castor.org), ExoLab Project (www.exolab.org), and Intalio, Inc. (www.intalio.org): Castor. Alessandro Colantonio: CONCISE Bitmap Library. d3-cloud. Day Management AG: Content Repository for Java. Dygraphs Gallery. Eclipse Foundation, Inc. (www.eclipse.org): Babel, Data Tools Platform (DTP) ODA, Eclipse SDK, Graphics Editor Framework (GEF), Eclipse Modeling Framework (EMF), Jetty, and Eclipse Web Tools Platform (WTP). Bits Per Second, Ltd. and Graphics Server Technologies, L.P.: Graphics Server. Dave Gandy: Font Awesome. Gargoyles Software Inc.: HtmlUnit. GNU Project: GNU Regular Expression. Google Charts. Groovy project (groovy.codehaus.org): Groovy. Guava Libraries: Google Guava. HighSlide: HighCharts. headjs.com: head.js. Hector Project: Cassandra Thrift, Hector. Jason Hsueh and Kenton Varda (code.google.com): Protocol Buffer. H2 Database: H2 database. IDAutomation.com, Inc.: IDAutomation. IDR solutions Ltd.: JPedal JBIG2. InfoSoft Global (P) Ltd.: FusionCharts, FusionMaps, FusionWidgets, PowerCharts. InfoVis Toolkit. Matt Inger (sourceforge.net): Ant-Contrib. Matt Ingenthron, Eric D. Lambert, and Dustin Sallings (code.google.com): Spymemcached. International Components for Unicode (ICU): ICU library. JCraft, Inc.: JSch. jQuery: jQuery, jQuery Sparklines. Yuri Kanivets (code.google.com): Android Wheel gadget. LEAD Technologies, Inc.: LEADTOOLS. The Legion of the Bouncy Castle: Bouncy Castle Crypto APIs. Bruno Lowagie and Paulo Soares: iText. Membrane SOA Model. MetaStuff: dom4j. Microsoft Corporation (Microsoft Developer Network): CompoundDocument Library. Mozilla: Mozilla XML Parser. MySQL Americas, Inc.: MySQL Connector/J. Netscape Communications Corporation, Inc.: Rhino. NodeJS. nullsoft project: Nullsoft Scriptable Install System. OOPS Consultancy: XMLTask. OpenSSL Project: OpenSSL. Oracle Corporation: Berkeley DB, Java Advanced Imaging, JAXB, Java SE Development Kit (JDK), Jstl, Oracle JDBC driver. PostgreSQL Global Development Group: pgAdmin, PostgreSQL, PostgreSQL JDBC driver. Progress Software Corporation: DataDirect Connect XE for JDBC Salesforce, DataDirect JDBC, DataDirect ODBC. Quality Open Software: Simple Logging Facade for Java (SLF4J), SLF4J API and NOP. Raphael. RequireJS. Rogue Wave Software, Inc.: Rogue Wave Library SourcePro Core, tools.h++. Sencha Inc.: Extjs, Sencha Touch. Shibboleth Consortium: OpenSAML, Shibboleth Identity Provider. Matteo Spinelli: iscroll. StAX Project (stax.codehaus.org): Streaming API for XML (StAX). Sam Stephenson (prototype.conio.net): prototype.js. SWFObject Project (code.google.com): SWFObject. ThimbleWare, Inc.: JMemcached. Twitter: Twitter Bootstrap. VMware: Hyperic SIGAR. Woodstox Project (woodstox.codehaus.org): Woodstox Fast XML processor (wstx-asl). World Wide Web Consortium (W3C) (MIT, ERCIM, Keio): Flute, JTIty, Simple API for CSS. XFree86 Project, Inc.: (www.xfree86.org): xvfb. ZXing Project (code.google.com): ZXing.

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

Document No. 141215-2-430360 February 11, 2015

About Installing and Deploying Eclipse BIRT	v
Chapter 1 Introducing Eclipse BIRT Report Designers	1
About Eclipse BIRT Designers	2
BIRT components	2
Eclipse BIRT packages	3
Types of BIRT builds	4
Chapter 2 Installing an Eclipse BIRT Report Designer	5
Installing Eclipse BIRT Report Designer overview	6
Installing BIRT Report Designer	6
Installing BIRT RCP Report Designer	7
Troubleshooting installation problems	8
Avoiding cache conflicts after you install a BIRT report designer	8
Specifying a Java Virtual Machine when starting a BIRT report designer	9
Installing a language pack	9
Updating a BIRT Report Designer installation	10
Updating a BIRT RCP Report Designer installation	11
Chapter 3 Installing other Eclipse BIRT packages	13
Eclipse BIRT packages overview	14
Installing Chart Engine	14
Installing BIRT Data Tools Platform Integration	15
Installing BIRT Demo Database	16
Installing Report Engine	17
Installing BIRT Samples	19
Installing BIRT Source Code	19
Installing BIRT Web Tools Integration	21
Configuring report previews	22
Chapter 4 Deploying a BIRT report to an application server	23
BIRT report deployment overview	24
About application servers	24
Deploying to Tomcat	24
Deploying to other application servers	24

Setting up the BIRT report viewer	25
Installing the BIRT report viewer	25
Testing the BIRT report viewer installation	26
Verifying that Apache Tomcat is running the BIRT report viewer	27
Changing the BIRT report viewer context root	28
Changing the BIRT report viewer location	28
Placing fonts on the application server	29
BIRT report viewer context parameters	29
Viewing a report using a browser	30
Using connection pooling	31
Setting up a report to use connection pooling	31
Using a jndi.properties file	31
Configuring a JNDI connection object	32
Index	35

About Installing and Deploying Eclipse BIRT

Installing and Deploying Eclipse BIRT introduces the currently available BIRT reporting packages, the prerequisites for installation, and the steps to install the packages and deploy BIRT technology on an application server.

Installing and Deploying Eclipse BIRT includes the following chapters:

- *About Installing and Deploying Eclipse BIRT*. This chapter provides an overview of this guide.
- *Chapter 1. Introducing Eclipse BIRT Report Designers*. BIRT provides a number of separate packages as downloadable archive (.zip or .tar.gz) files on the Eclipse website. 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 available packages and the prerequisites for those packages.
- *Chapter 2. Installing an Eclipse BIRT Report Designer*. BIRT provides two report designers as separate packages, which are downloadable archive (.zip) files on the Eclipse website. This chapter describes the steps required to install each of the available report designers on a Microsoft Windows system.
- *Chapter 3. Installing other Eclipse 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 7.0. From those instructions, a developer can infer how to deploy to other servers.

1

Introducing Eclipse BIRT Report Designers

This chapter contains the following topics:

- About Eclipse BIRT Designers
- BIRT components
- Eclipse BIRT packages
- Types of BIRT builds

About Eclipse BIRT Designers

There are two Eclipse designer applications that you can use to create BIRT reports:

- **BIRT Report Designer**
A tool that a report developer uses to build a BIRT report design and preview a report. BIRT Report Designer is a set of Eclipse plug-ins that includes BIRT Report Engine, BIRT Chart Engine, and BIRT Demo Database. This tool supports Java and JavaScript customization. BIRT Report Designer requires multiple Eclipse platform components and a Java Development Kit (JDK).
- **BIRT RCP Report Designer**
A simplified tool that a novice report developer can use to build a BIRT report design and preview a report. BIRT RCP (Rich Client Platform) Report Designer includes BIRT Report Engine, BIRT Chart Engine, and BIRT Demo Database without the additional overhead of the full Eclipse platform. This tool supports JavaScript customization, but does not support Java customization or debugging. It is provided primarily as a reference implementation of a Rich Client Platform application and is not the focus of this book.

BIRT components

BIRT Report Designer 4.3.1 includes the following key components:

- **Eclipse Software Development Kit (SDK) 4.3.1**
The SDK supports the development of plug-ins and extensions to the Eclipse platform, including BIRT. The SDK includes the core platform, the Java Development Tools (JDT), and the Plug-in Developer Environment (PDE).
- **Data Tools Platform (DTP) 1.11.1**
The DTP is a set of development tools used to develop plug-ins that access data sources and retrieve data.
- **Eclipse Modeling Framework (EMF) 2.9.1**
The EMF supports the development of BIRT charts. The EMF includes the Service Data Objects (SDO), which is a graph-structured data object that supports applying changes to a graph back to the data source.
- **Graphical Editing Framework (GEF) 3.9.1**
The GEF is an Eclipse plug-in that the BIRT Report Designer user interface requires. This framework provides a rich, consistent, graphical editing environment for an application running on the Eclipse platform.

- Eclipse Web Tools Platform (WTP) 3.5.1
The WTP is a set of Eclipse plug-ins that support deploying the BIRT report viewer to an application server. The package includes source and graphical editors, tools, wizards, and APIs that support deploying, running, and testing.

Eclipse BIRT packages

Eclipse BIRT provides the following packages. These packages do not include, but do require a Java Development Kit. Eclipse BIRT is supported by Java 1.6 JDK and higher.

- Report Designer Full Eclipse Install (All-in-One)
Contains BIRT and the Eclipse Integrated Development Environment (IDE). This all-in-one installation is the easiest way to install BIRT and is available in both Windows and Linux archive formats.
- Report Designer
Contains only BIRT for installing in an existing Eclipse Integrated Development Environment (IDE).
- RCP Report Designer
Contains a simplified version of BIRT without the Eclipse IDE.
- BIRT Software Development Kit (SDK)
Contains the source code for the BIRT plug-ins and documents including Javadoc and descriptions of extension points.
- Report Engine
Contains the run-time version of BIRT for installing in a J2EE application server or as an OSGi run-time.
- Chart Engine
Contains the stand-alone library that supports embedding a chart in a Java application.
- BIRT Web Tools Integration
Contains the plug-ins required to use the BIRT Web Project Wizard in a Web Tools Project, including the source code.
- BIRT Source Code
Contains the BIRT source code for a specific build. All source code is in a plug-in format ready to import into a workspace to build BIRT. These plug-ins are the required libraries for a standard BIRT installation. Additional libraries may be necessary. For example, this package does not include the Data Tools Platform (DTP) source code.

- **BIRT Samples**
Contains sample reports and charts, plus application examples that use the Chart, Report Engine, and Design Engine APIs.
- **BIRT Test Suites**
Contains automated tests for BIRT used by BIRT source-code developers. Download this package to use as a basis for testing a customized version of BIRT.
- **BIRT Demo Database**
Contains the package for defining and loading the demonstration database into Apache Derby and MySQL, including SQL and data files. The demonstration database package is a convenient way to install the Classic Models, Inc. database schema and data in the Apache Derby and MySQL systems. The package does not include any BIRT software. The Report Designer and the RCP Report Designer packages include the demonstration database for Apache Derby.

The demonstration database supports the following Apache and MySQL versions:
 - Apache Derby version 5.1 or higher
 - MySQL Connector/J version 3.1 or MySQL client version 4.x

Types of BIRT builds

The Eclipse BIRT download site makes several types of builds available for BIRT. The following list describes these builds:

- **Release build**
A production build that passes the complete test suite for all components and features. Use the release build to develop applications.
- **Milestone build**
A development build that provides access to newly completed features. The build is stable, but it is not production quality. Use this type of build to preview new features and develop future reporting applications that depend on those features.
- **Stable build**
A development build that is stable, but passes a reduced test suite. New features are in an intermediate stage of development. Use a stable build to preview new features.

Installing an Eclipse BIRT Report Designer

This chapter contains the following topics:

- Installing Eclipse BIRT Report Designer overview
- Installing BIRT Report Designer
- Installing BIRT RCP Report Designer
- Troubleshooting installation problems
- Installing a language pack
- Updating a BIRT Report Designer installation
- Updating a BIRT RCP Report Designer installation

Installing Eclipse BIRT Report Designer overview

Installing BIRT Report Designer adds a report design perspective to the Eclipse Integrated Development Environment (IDE). To install a BIRT Report Designer, download an archive file from the Eclipse website and extract it in your existing Eclipse environment. BIRT Report Designer is available for various Linux and Microsoft Windows platforms. The following sections describe how to install BIRT Release 4.3.1 on a Microsoft Windows system.

Installing BIRT Report Designer

If you are new to Eclipse and BIRT, download and install BIRT Report Designer Full Eclipse Install (All-in-One) package to start developing and designing BIRT reports. This package includes the Eclipse Integrated Development Environment (IDE), BIRT Report Designer, and all other required Eclipse components.

You must first download and install Java JDK 1.6 or higher. The examples in this edition use JDK 1.7.

If you experience persistent core exception or OSGi framework errors, it is often due to a mismatch between your Java environment and your Eclipse environment. For example, errors occur if you have a 32-bit Java environment and a 64-bit Eclipse installation.

Complete the following procedure to download BIRT Designer All-in-One on a Windows or Linux system.

How to install BIRT Report Designer All-in-One

- 1 Using your browser, navigate to the main BIRT web page at:

<http://www.eclipse.org/birt/>

- 2 From BIRT Project, choose Download BIRT 4.3.1.
- 3 From BIRT Report Downloads, choose All-in-One.
- 4 On BIRT Report Downloads, select the Download Link that meets your requirements, for example, Windows 64-bit.

Eclipse downloads - mirror selection appears. This page shows all the sites that provide this download file.

- 5 Choose the download site that is closest to your location.

The BIRT Report Designer all-in-one archive file downloads to your system.

- 6 Extract the files in the archive to the default location, C:\ on your machine, creating C:\eclipse.

To open BIRT Report Designer, start Eclipse, then start BIRT Report Designer as described in the following procedure. BIRT Report Designer is a perspective within Eclipse.

How to open BIRT Report Designer

- 1 To start Eclipse, in C:\eclipse, run eclipse.exe.
- 2 Select the default workspace.
- 3 If necessary, close the welcome window. In the Eclipse Window menu, choose Open Perspective→Other→Report Design.

Eclipse displays the BIRT Report Designer perspective.

Installing BIRT RCP Report Designer

BIRT RCP Report Designer is a stand-alone report design application that enables report developers to produce reports in both web and PDF formats. This application uses the Eclipse Rich Client Platform (RCP) to provide a report design environment that is less complex than the full Eclipse platform. BIRT RCP Report Designer runs on Windows only.

The RCP Report Designer is provided primarily as a reference implementation of a Rich Client Platform application and is not the focus of this book.

To install BIRT RCP Report Designer, download and extract an archive file. The following examples use Release 4.3.1.

Complete the following procedure to download and install BIRT RCP Report Designer on a Windows system.

How to install BIRT RCP Report Designer

- 1 Using your browser, navigate to the main BIRT web page at:

<http://www.eclipse.org/birt/>

- 2 From BIRT Home, choose Download BIRT 4.3.1.
- 3 From BIRT Report Downloads, choose RCP Designer.

Eclipse downloads - mirror selection appears. This page shows all the sites that provide this download file.

- 4 Choose the download site that is closest to your location.

The BIRT RCP Report Designer archive downloads to your system.

- 5 Extract the archive file to a hard drive location that you specify.

The extraction creates a directory named birt-rcp-report-designer-4_3_1 at the location that you specify.

To test the installation, start BIRT RCP Report Designer as described in the following procedure.

How to test the BIRT RCP Report Designer installation

- 1 Navigate to the `birt-rcp-report-designer-4_3_1` directory.
- 2 To run BIRT RCP Report Designer, double-click `BIRT.exe`. BIRT RCP Report Designer appears.

Troubleshooting installation problems

Installing a BIRT report designer is a straightforward task. If you extract the archive file to the appropriate location and the required supporting files are also available in the expected location, your BIRT report designer will work. One of the first steps in troubleshooting an installation problem is confirming that all files are in the correct location.

Verify that the `/eclipse/plugins` directory contains JAR files whose names begin with `org.eclipse.birt`, `org.eclipse.emf`, and `org.eclipse.gef`. The following sections describe troubleshooting steps that resolve two common installation errors.

Avoiding cache conflicts after you install a BIRT report designer

Eclipse caches information about plug-ins for faster start-up. After you install or upgrade BIRT Report Designer or BIRT RCP Report Designer, using a cached copy of certain pages can lead to errors or missing functionality. The symptoms of this problem include the following conditions:

- The Report Design perspective does not appear in Eclipse.
- You receive a message that an error occurred when you open a report or use the Report Design perspective.
- JDBC drivers that you installed do not appear in the driver manager.

The solution is to remove the cached information. The recommended practice is to start either Eclipse or BIRT RCP Report Designer from the command line with the `-clean` option.

To start Eclipse, use the following command:

```
eclipse.exe -clean
```

To start BIRT RCP Report Designer, use the following command:

```
BIRT.exe -clean
```

Specifying a Java Virtual Machine when starting a BIRT report designer

You can specify which Java Virtual Machine (JVM) to use when you start a BIRT report designer. This specification can be important, particularly for users on Linux, when path and permission problems prevent the report designer from locating an appropriate JVM to use.

On Windows and Linux systems, you can either start a BIRT report designer from the command line or create a command file or shell script that calls the appropriate executable file with the JVM path. Alternatively, specify the command-line option in the `eclipse.ini` file in the BIRT installation folder.

How to specify which JVM to use when you start a BIRT report designer

The example in this section uses BIRT Report Designer on a Windows system. On the command line, type a command similar to:

```
eclipse.exe -vm "C:\Program Files\Java\jre6\bin\java.exe"
```

Installing a language pack

All BIRT user interface components and messages are internationalized through the use of properties files. BIRT uses English as the default language, but supports other languages via language packs that contain the required properties files. The Eclipse community has translated BIRT 4.3.1 and provides language packs for many languages, including the following:

Albanian, Arabic, Basque, Bulgarian, Catalan, Chinese (Simplified), Chinese (Traditional), Czech, Danish, Dutch, English (Australian), English (Canadian), Estonian, Finnish, French, German, Greek, Hebrew, Hindi, Hungarian, Indonesian, Italian, Japanese, Klingon, Korean, Kurdish, Malayalam, Mongolian, Norwegian, Persian, Polish, Portuguese, Portuguese (Brazilian), Romanian, Russian, Serbian, Slovak, Slovenian, Spanish, Swedish, Thai, Turkish, Ukrainian, and Pseudo Translations

The following instructions explain how to download and install a language pack for BIRT 4.3.1 on Windows.

How to download and install a language pack

To download and install a language pack, perform the following steps:

1 Using your browser, navigate to the BIRT language pack web page at:

<http://www.eclipse.org/babel/downloads.php>

2 From Babel Language Pack Zips, choose Kepler.

3 Select and download the compressed file for each language that you need.

- 4 Extract files in the language pack archive into the Eclipse installation directory.
For example, if C:/eclipse is your Eclipse directory, extract files into C:/eclipse.
- 5 Start Eclipse and choose Window→Preferences→Report Design→Preview.
- 6 Select the language of choice from the drop-down list in Choose your locale.
- 7 Restart Eclipse.

If Windows is not running under the locale you need for BIRT, start Eclipse using the `-nl <locale>` command line option, where `<locale>` is a standard Java locale code, such as `es_ES` for Spanish as spoken in Spain. A list of locale codes is available at the following URL:

<http://www.oracle.com/technetwork/java/javase/locales-137662.html>

Updating a BIRT Report Designer installation

Because BIRT Report Designer is a Java-based application, updating an installation typically requires replacing the relevant files. Eclipse supports the update process for BIRT Report Designer by providing the Update Manager. BIRT RCP Report Designer is a stand-alone product, so you must replace the existing version with a newer version.

This section describes the steps required to update the following BIRT packages:

- Report Designer
- RCP Report Designer

You can use the Eclipse Update Manager to find and install newer major releases of BIRT Report Designer.

How to update a BIRT Report Designer installation using the Update Manager

- 1 In Eclipse, choose Help→Check for Updates.
- 2 In Available Updates, choose Select All then choose Next.
- 3 In Update Details, choose Next.
- 4 In Review Licenses, accept the license agreement terms and choose Finish.
- 5 When the update completes, restart your computer.

How to update BIRT Report Designer manually

- 1 Back up the workspace directory if it is in the eclipse directory structure.
- 2 To remove the BIRT files, use one of the following techniques:

- To prepare for a new all-in-one installation, remove the entire eclipse directory.
- To prepare for only a BIRT Report Designer installation, remove only the BIRT components.
 - 1 Navigate to the eclipse\features directory.
 - 2 Delete all JAR files and subdirectories with birt in their names.
 - 3 Navigate to the eclipse\plugins directory.
 - 4 Delete all JAR files and subdirectories with birt in their names.
- 3 Download and install BIRT Report Designer as described earlier in this book.
- 4 Restore the workspace directory, if necessary.
- 5 Restart BIRT Report Designer with the -clean option:
`eclipse.exe -clean`

Updating a BIRT RCP Report Designer installation

Unlike BIRT Report Designer, BIRT RCP Report Designer is a stand-alone application. To update this application, you delete the entire application and reinstall a newer version. If you created your report designs and resources in the birt-rcp-report-designer-<version> directory structure, you must back up your workspace directory and any resources that you want to keep before you delete BIRT RCP Report Designer. After you install a newer version of the application, you can copy your files back to the application directory structure.

As a best practice, do not keep your workspace in the birt-rcp-report-designer-<version> directory structure. Keeping your workspace in a different location enables you to update your installation more easily in the future.

How to update BIRT RCP Report Designer

- 1 Back up the workspace directory and any other directories that contain report designs, libraries, and other resources, if they are in the birt-rcp-report-designer-<version> directory structure.
- 2 Delete the birt-rcp-report-designer-<version> directory.
- 3 Download and install BIRT RCP Report Designer as described earlier in this book.
- 4 Restore the directories that you backed up in step 1, if necessary.
- 5 Restart BIRT RCP Report Designer with the -clean option:
`BIRT.exe -clean`

Installing other Eclipse BIRT packages

This chapter contains the following topics:

- Eclipse BIRT packages overview
- Installing Chart Engine
- Installing BIRT Data Tools Platform Integration
- Installing BIRT Demo Database
- Installing Report Engine
- Installing BIRT Samples
- Installing BIRT Source Code
- Installing BIRT Web Tools Integration
- Configuring report previews

Eclipse BIRT packages overview

Beyond the BIRT Report Designer packages, BIRT provides a number of other separate packages as downloadable archive files on the Eclipse website. Some of these packages are stand-alone modules, others require an existing Eclipse or BIRT environment, and still others provide additional functionality to report developers and application developers. This chapter describes the steps required to install the BIRT packages shown in the following list:

- Chart Engine
- Data Tools Platform (DTP) Integration
- Demo Database
- Report Engine
- Samples
- Source Code
- Web Tools Integration

This chapter also covers online help and preview configuration.

Installing Chart Engine

Chart Engine supports adding charting capabilities to a Java application. An application can use Chart Engine without using the BIRT reporting functionality or Report Engine. Chart Engine integrates into an existing Eclipse platform on Microsoft Windows, UNIX, or Linux. You can also install Chart Engine on an existing J2EE application server. To use Chart Engine, you use its public API, org.eclipse.birt.chart.

Both BIRT Report Designer and BIRT RCP Report Designer include all the components of Chart Engine. If you are using a BIRT report designer, you do not need to install BIRT Chart Engine separately.

How to install BIRT Chart Engine

On the BIRT website, perform the following operations:

- 1 Navigate to BIRT Downloads for build 4.3.1.
- 2 In More Downloads, choose full BIRT 4.3.1 Download Page.
- 3 In the Chart Engine section, choose the Chart Engine archive file:
`birt-charts-4_3_1.zip`
- 4 Extract all files from the archive to a location of your choice.

- 5 Start Eclipse from the command line with the `-clean` option to remove cached information.

The archive extraction process creates the following subdirectories in the extraction directory:

- **ChartRuntime**
This directory contains the plug-ins and libraries that an Eclipse platform requires to run, render, and edit charts.
- **ChartSDK**
This directory contains the plug-ins and libraries from the ChartRuntime directory plus the SDK that you need to create your own charting applications. It also includes examples, source code, and a Web Tools Platform (WTP) extension to support charts in web applications.
- **DeploymentRuntime**
This directory contains the libraries that you need to run your charting application in a non-Eclipse environment such as on an application server.

The Chart Engine download file also includes extensive Frequently Asked Questions (FAQ) and examples illustrating how to use Chart Engine. After extracting the archive, you can find the FAQ at the following location:

```
<CHART_ENGINE>/DeploymentRuntime/ChartEngine/docs/Charts_FAQ.doc
```

The examples are in a JAR file located at:

```
<CHART_ENGINE>/ChartSDK/eclipse/plugins  
/org.eclipse.birt.chart.examples_<version>.jar
```

Installing BIRT Data Tools Platform Integration

This package includes the minimal set of Data Tools Platform (DTP) plug-ins that BIRT Report Designer requires. If you install the BIRT Report Designer package in an existing Eclipse installation, you can install this BIRT DTP Integration package instead of the full DTP platform.

How to install BIRT DTP Integration

On the BIRT website, perform the following operations:

- 1 Navigate to BIRT Downloads for build 4.3.1.
- 2 In the Framework section, choose DTP version 1.11.1 release.
- 3 Extract all files from the `ntp_1.11.1.zip` archive file to the directory that contains your Eclipse directory.

DTP feature and plug-in files extract to the `eclipse\features` and `eclipse\plugins` directories.

- 4 Start Eclipse from the command line with the `-clean` option.

To test the BIRT DTP Integration package, open the Report Design perspective in Eclipse, as described in the following procedure.

How to test the BIRT DTP Integration installation

- 1 Start Eclipse.
- 2 From the Eclipse Window menu, choose Open Perspective➤Other. In Open Perspective, choose Report Design. Eclipse displays the BIRT Report Designer perspective.

Installing BIRT Demo Database

The BIRT Demo Database package provides the Classic Models, Inc. database that this book uses for example procedures. The database is provided in the following formats:

- Apache Derby
- MySQL

BIRT Report Designer and BIRT RCP Report Designer include this database in Apache Derby format, as the Classic Models, Inc. sample database data source. Install BIRT Demo Database if you want to use the native drivers to access this data source.

How to install BIRT Demo Database

On the BIRT website, perform the following operations:

- 1 Navigate to BIRT Downloads for build 4.3.1.
- 2 In More Downloads, choose full BIRT 4.3.1 Download Page.
- 3 In the Demo Database section, choose the Demo Database archive file:
`birt-database-4_3_1.zip`
- 4 Extract all files from the `birt-database` archive to a location of your choice.
Extracting creates a directory, `ClassicModels`, which contains the BIRT Demo Database in Apache Derby and MySQL formats.

To test the BIRT Demo Database, first connect to the database with the native database client tool or a Java application.

How to access BIRT Demo Database using a database client tool

Perform one of the following sets of tasks, based on your preferred database:

- Apache Derby database
Connect to the database in the derby subdirectory of ClassicModels.
- MySQL
 - 1 Navigate to the mysql subdirectory of ClassicModels.
 - 2 Create a database to use or edit create_classicmodels.sql to uncomment the lines that create and select the classicmodels database.
 - 3 Use the mysql command line interface to run create_classicmodels.sql.
 - 4 Review load_classicmodels.sql to determine if you can use the script on your platform without editing. Use the mysql command line interface to run load_classicmodels.sql.

Next, connect to the database from BIRT Report Designer or BIRT RCP Report Designer.

How to access BIRT Demo Database from a BIRT report designer

Connect to the database using BIRT Report Designer or BIRT RCP Report Designer.

- 1 To access the Classic Models, Inc. database in Apache Derby or MySQL format, first add the driver JAR files to a BIRT report designer installation.
- 2 In any report design, create a data source on the database. In the same report design, create a data set on the data source.

Installing Report Engine

Report Engine supports adding reporting capabilities to a Java application. BIRT Report Engine integrates into an existing Eclipse platform on Microsoft Windows, UNIX, or Linux. You can also install report engine components on an existing J2EE application server. To support quick deployment of reporting functionality to an application server, Report Engine includes a web archive (.war) file.

How to install BIRT Report Engine

On the BIRT website, perform the following operations:

- 1 Navigate to BIRT Downloads for build 4.3.1.
- 2 In Deployment, choose Runtime, then download the following file:
`birt-runtime-4_3_1.zip`
- 3 Extract all files from the BIRT runtime archive to a suitable directory.
- 4 For each new Java project, add the jars in the ReportEngine\lib subdirectory to your build path.

To test the installation, run the Report Engine report generation command line example. This example uses a batch (.bat) file on a Windows system and a shell script (.sh) file on a UNIX or Linux system. This file takes the parameters shown in Table 3-1.

Table 3-1 Parameters for the genReport script

Parameter	Valid for mode	Values
Execution mode -m		Valid values are run, render, and runrender. The default is runrender.
Target encoding -e	render, runrender	A valid encoding. The default is utf-8.
Output format -f	render, runrender	Valid values are HTML and PDF. The default value is HTML.
Report parameters file -F	run, runrender	Path to the parameter file. This file contains lines with the format: <parameter name>=<value>
Locale -l locale	run, runrender	A valid locale string. The default locale is en.
Output file name -o	render, runrender	The full path of the output file. The default value is the name of the report design with an extension based on the output format, .html for an HTML file and .pdf for a PDF file.
Report parameter -p "parameter name=value"	run, runrender	If you provide parameter values with the -p parameter, these values override the values in the report parameters file specified by -F.
HTML format -t	run, runrender	Valid values are HTML and ReportletNoCSS. HTML is the default. This format wraps the HTML output in an <HTML> tag. ReportletNoCSS does not wrap the HTML output in an <HTML> tag.
Report design file	All modes	The full path of the report design file. This parameter must be the last parameter on the command line.

How to test the BIRT Report Engine installation

- 1 From the command line, navigate to the directory where you installed BIRT Report Engine.
- 2 Navigate to the ReportEngine subdirectory.
- 3 To run the genReport script, run the appropriate file for your operating system:

- On a Windows platform, run `genReport.bat`.
- On a UNIX or Linux platform, run `genReport.sh`.

Enclose the value for a command line parameter in quotes. For example, the following Windows platform command uses the value, Hello, for the parameter, sample, to generate an HTML file from the report design, test.rptdesign:

```
genReport -p "sample=Hello"  
          "C:\birt-runtime-4_3_1\WebViewerExample\test.rptdesign"
```

`genReport` generates the required output file.

- 4 Open the output file. In this example, the file is `C:\birt-runtime-4_3_1\WebViewerExample\test.html`.

For more information about setting up the BIRT Report Engine, see Chapter 4, “Deploying a BIRT report to an application server.”

Installing BIRT Samples

BIRT Samples provides examples of a BIRT report item extension and of charting applications. The report item extension integrates into BIRT Report Designer and BIRT Report Engine.

How to install BIRT Samples

On the BIRT website, perform the following operations:

- 1 Navigate to BIRT Downloads for build 4.3.1.
- 2 In More Downloads, choose full BIRT 4.3.1 Download Page.
- 3 In the Samples section, download the Samples archive file:

```
birt-sample-plugins-4_3_1.zip
```

- 4 Extract all files from the archive to the directory that contains your Eclipse directory.

Installing BIRT Source Code

This package includes the source code for all BIRT plug-ins. You can examine this code to see how BIRT generates reports from designs. You can also import this source code into a workspace to build a custom BIRT installation.

How to install BIRT Source Code

- 1 Navigate to BIRT Downloads for build 4.3.1.

- 2 In More Downloads, choose full BIRT 4.3.1 Download Page.
- 3 In the BIRT Source Code section, download the BIRT Source Code archive file:

`birt-source-4_3_1.zip`

- 4 Extract all files from the archive to a new workspace directory.
Extracting creates the build files and BIRT features and plugins directories in that workspace directory.

To test the BIRT Source Code package, import the source code projects into your workspace.

How to test the BIRT Source Code installation

- 1 Start Eclipse.
- 2 Set the Java preferences for BIRT.
 - 1 From the Eclipse main menu, choose Window→Preferences.
 - 2 Expand Java, select Compiler. Make the following selections:
 - Set Compiler Compliance Level to 1.6.
 - Deselect Use default compliance settings.
 - Set Generated .class files compatibility to 1.6.
 - Set Source compatibility to 1.6.
 - 3 Choose OK.
- 3 From the Eclipse main menu, choose File→Import.
- 4 In Import—Select, expand General and select Existing Projects into Workspace. Choose Next.
- 5 In Import—Import Projects, select Select root directory, then type or browse to your workspace directory.
The BIRT features and plug-ins appear in Projects.
- 6 Choose Finish.
Eclipse builds the BIRT projects.

If the projects do not build correctly, check that you installed the prerequisites for BIRT Report Designer, as described in Chapter 1, “Introducing Eclipse BIRT Report Designers.” If you have not installed the BIRT Report Designer Full Eclipse Install, download this package and extract any JAR files that the build requires. Add any libraries that Eclipse does not find to the build paths of specific projects to resolve other build errors.

Installing BIRT Web Tools Integration

This package includes the minimal set of BIRT plug-ins that the Eclipse Web Tools Platform (WTP) requires to build a BIRT web project using the BIRT Web Project Wizard. This package also includes the source code for these plug-ins.

How to install BIRT Web Tools Integration

On the BIRT website, perform the following operations:

- 1 Navigate to BIRT Downloads for build 4.3.1.
- 2 In More Downloads, choose full BIRT 4.3.1 Download Page.
- 3 In the BIRT Web Tools Integration section, choose the BIRT Web Tools Integration archive file:

`birt-wtp-integration-sdk-4_3_1.zip`

- 4 Extract all files from the archive to the directory that contains your Eclipse directory.

Extracting creates the BIRT features and plug-ins in the `eclipse\features` and `eclipse\plugins` directories.

To test the BIRT Web Tools Integration package, create a BIRT web project in Eclipse.

How to test the BIRT Web Tools Integration installation

- 1 Start Eclipse.
- 2 From the Eclipse main menu, choose File→New→Project.
- 3 In New Project—Select a wizard, expand Web, select Dynamic Web Project. Choose Next.
- 4 In New Project—Dynamic Web Project, make the choices that you need for your BIRT web project, then choose Finish.

If you do not have the Java EE perspective open, Eclipse displays the following message:

This kind of project is associated with the Java EE perspective. Do you want to open this perspective now?

Choose Yes.

Configuring report previews

Adobe PDF and other formats change frequently. For testing report designs, the recommended preview setting is to always use external browsers.

How to configure BIRT preview to use external browsers

- 1** Start Eclipse.
- 2** From the Eclipse main menu, choose Window→Preferences.
- 3** In Preferences, choose Report Design→Preview.
- 4** In Preview, choose Always use external browsers.
- 5** Choose OK.

You can also specify which browser is launched using Window→Preferences→General→Web browser.

Deploying a BIRT report to an application server

This chapter contains the following topics:

- BIRT report deployment overview
- About application servers
- Setting up the BIRT report viewer
- Viewing a report using a browser
- Using connection pooling

BIRT report deployment overview

One way to view a BIRT report on the web is to deploy the BIRT report viewer to an application server, such as Apache Tomcat, IBM WebSphere, JBoss, or BEA WebLogic.

BIRT Report Engine includes the BIRT report viewer as a web archive (.war) file and as a set of files and folders. Deploying the BIRT report viewer requires copying files from BIRT Report Engine, which you must install separately from BIRT Report Designer.

This chapter provides information about deploying the BIRT report viewer both by using the WAR file and by using the set of files and folders.

About application servers

The instructions in this chapter specifically address deploying a BIRT report to Apache Tomcat version 7.0. Although the information in this chapter is specific to this version of Tomcat, a BIRT report can also be deployed to other versions of Tomcat and to other application servers.

Deploying to Tomcat

There are only minor differences between the requirements for deploying to Tomcat version 7.0 and deploying to other versions of Apache Tomcat. Apache Tomcat 7.0 runs Java 6 or later by default, which is also the recommended Java level for BIRT 4.3.1

Deploying to other application servers

Most application servers require a web archive (WAR) file that contains everything application requires, including a web.xml file describing the application and various deployment preferences. BIRT Report Engine includes a WAR file appropriate to Tomcat. Typically, the WAR file requires no modification. In some cases, developers who have experience with other application servers can modify the web.xml file to reflect the requirements of their environments. The section on mapping the report viewer folders, later in this chapter, discusses setting the web.xml parameters.

Deployment to JBoss may require copying axis.jar and axis-ant.jar from WEB-INF/lib to the following directory:

```
jboss/server/default/lib
```

This step is not necessary for all versions of JBoss, but if there are difficulties with a JBoss deployment, copying these files can resolve the problem.

Setting up the BIRT report viewer

You must place the BIRT report viewer in a location where Apache Tomcat can access it. Typically, this location is the `$TOMCAT_INSTALL/webapps` directory. On restarting Apache Tomcat, the application server automatically recognizes and starts the BIRT report viewer application if the BIRT report viewer is in this folder.

Installing the BIRT report viewer

The BIRT report viewer files provide core functionality to run, render, and view BIRT reports. To use additional JDBC drivers that are not part of the standard BIRT packages, you must install these drivers as well as the BIRT report viewer itself. If you install the BIRT report viewer as a WAR file, you must include the JDBC drivers in the WAR file.

The following instructions assume that you have installed the BIRT Report Engine from the BIRT website, that your web application directory is `$TOMCAT_INSTALL/webapps`, and that your BIRT run-time installation directory is `$BIRT_RUNTIME`.

How to install the BIRT report viewer from the BIRT Report Engine WAR file

The steps to install the BIRT report viewer from the WAR file differ depending upon whether you need to include additional JDBC drivers for your reports. If there are no additional drivers, install the WAR file from the BIRT Report Engine installation. If you use additional JDBC drivers, you must pack them into the WAR file before you deploy it.

- To install the BIRT report viewer from the BIRT Report Engine WAR file, copy the BIRT Report Engine WAR file, `birt.war` to the Tomcat applications folder, `$TOMCAT_INSTALL/webapps`, as illustrated by the following DOS command:

```
copy $BIRT_RUNTIME/birt.war $TOMCAT_INSTALL/webapps
```

Then, restart Apache Tomcat.

- To install the BIRT report viewer with additional JDBC drivers, perform the following steps:

- 1 Create a temporary directory and navigate to that directory.
- 2 Unpack the BIRT Report Engine WAR file into the temporary directory, using a command similar to the following one:

```
jar -xf $BIRT_RUNTIME/birt.war
```

- 3 Copy the JAR files for your JDBC drivers to the following folder in the temporary directory:

```
WEB-INF/platform/plugins
  /org.eclipse.birt.report.data.oda.jdbc_<version>/drivers
```

- 4 Repack the BIRT Report Engine WAR file from the temporary directory into a new birt.war file, using a command similar to the following one:

```
jar -cf birt.war *
```

This command creates birt.war in the temporary directory.

- 5 Copy the new birt.war file to the Tomcat applications folder, \$TOMCAT_INSTALL/webapps, as illustrated in the following DOS command:

```
copy birt.war $TOMCAT_INSTALL/webapps
```

- 6 Restart Apache Tomcat.

How to install the BIRT report viewer from the BIRT Report Engine viewer folder

To install the BIRT report viewer as an application in a file system folder, use the WebViewerExample folder in the BIRT Report Engine installation.

- 1 Navigate to \$TOMCAT_INSTALL/webapps.
- 2 Create a subdirectory named birt.
- 3 Copy the web viewer example directory and all its subdirectories to this new folder, as illustrated by the following DOS command:

```
xcopy /E "$BIRT_RUNTIME/WebViewerExample"
  $TOMCAT_INSTALL/webapps/birt
```

- 4 If the BIRT reports need additional JDBC drivers, add the JAR files for the JDBC drivers to the following directory:

```
$TOMCAT_INSTALL/birt/WEB-INF/platform/plugins
  /org.eclipse.birt.report.data.oda.jdbc_<version>/drivers
```

- 5 Restart Apache Tomcat.

Testing the BIRT report viewer installation

To test the installation of the BIRT report viewer described in earlier sections, type the following URL in a web browser address field:

```
<server_name>:<port>/birt/
```

<server_name> is the name of the application server and <port> is the port that the application server uses.

Tomcat opens the JavaServer Page (JSP), index.jsp. This file exists in both the WAR file and in the BIRT report viewer root directory. A link on this page runs

the simple BIRT report design file, test.rptdesign. If the BIRT report viewer is installed correctly, Tomcat uses index.jsp to process the report design and generate and render the report that it describes. The first time you run the report, Tomcat compiles the JSP files that comprise the viewer, so there is a delay before the report appears in the web browser.

Verifying that Apache Tomcat is running the BIRT report viewer

If there are problems accessing the BIRT report viewer, use the Tomcat manager to verify that the BIRT report viewer is running on Apache Tomcat. Running the Tomcat manager requires a manager's account. If a Tomcat manager account does not exist, create one by adding the following line to \$TOMCAT_INSTALL/conf/tomcat-users.xml:

```
<user name="admin" password="tomcat" roles="manager" />
```

Having a manager's account available, first open the Tomcat main page, which for a typical Apache Tomcat installation is <http://localhost:8080>, as shown in Figure 4-1.

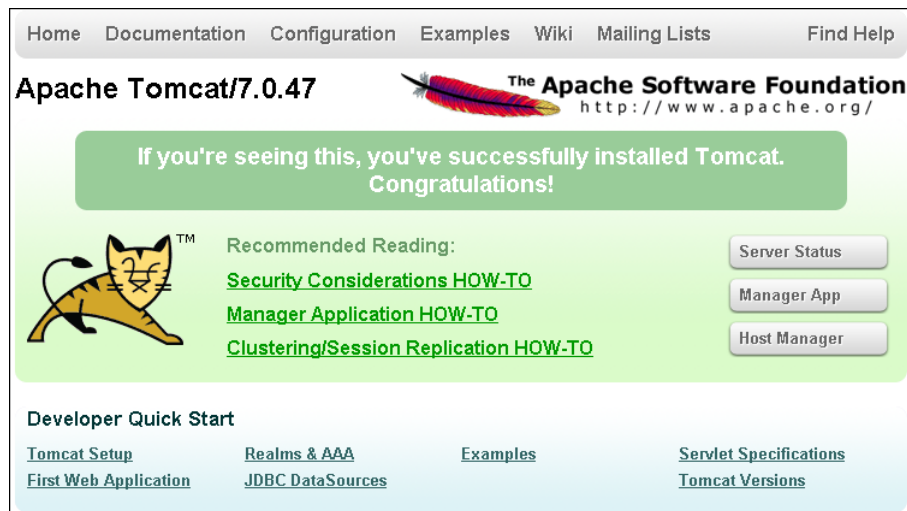




Figure 4-1 Apache Tomcat home page

On the Tomcat main page, choose Manager App. In the manager login window, type the user name and password of the manager account defined in the tomcat-users.xml file. When the BIRT report viewer application is running, the Running status for Eclipse BIRT Report Viewer is true, as shown in Figure 4-2.

Tomcat Web Application Manager

Message: OK

Manager

[List Applications](#) [HTML Manager Help](#) [Manager Help](#) [Server Status](#)

Applications

Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/birt	None specified	Eclipse BIRT Report Viewer	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

Figure 4-2 Running status for the BIRT report viewer

Changing the BIRT report viewer context root

By default, the context root of the URL for a web application is the path to the application directory or the WAR file. The default WAR file for the BIRT report viewer is `birt.war`, so the default URL to access a BIRT report from Apache Tomcat is similar to the following one:

```
http://localhost:8080/birt/run?__report=myReport.rptdesign
```

To change the BIRT context root, change the name of the `/birt` directory or the WAR file in `$TOMCAT_INSTALL/webapps`. Next, restart Apache Tomcat. In the URL to access your BIRT report, specify the name that you chose. For example, if you chose `reports`, the URL to access a BIRT report becomes:

```
http://localhost:8080/reports/run?__report=myReport.rptdesign
```

The URL examples in this section use a relative path to access the report design. The `BIRT_VIEWER_WORKING_FOLDER` parameter sets the path to access a report design as a relative path.

Changing the BIRT report viewer location

To place the BIRT report viewer in a location other than `$TOMCAT_INSTALL/webapps`, add a context mapping entry to the `server.xml` file in

\$TOMCAT_INSTALL/conf by adding the following line to server.xml just above the </host> tag near the end of the file:

```
<Context path="/birt_context" docBase="BIRT_Path"/>
```

birt_context is the context root for the BIRT report viewer application. *BIRT_Path* is the absolute path to the BIRT report viewer directory.

Save the changes to server.xml and restart Apache Tomcat to make the changes active.

Placing fonts on the application server

BIRT Report Engine requires certain TrueType fonts to display a PDF report. BIRT searches for fonts in the common font directories for Windows and Linux. The directories that BIRT searches on a Windows system include:

- /windows/fonts for drives A through G
- /WINNT/fonts for drives A through G

and on a Linux system include:

- /usr/share/fonts/default/TrueType
- /usr/share/fonts/truetype

If PDF reports appear to be missing content, place the necessary fonts in any of the directories in the preceding list. Alternatively, specify your own font search path in the environment variable BIRT_FONT_PATH.

BIRT report viewer context parameters

To determine the locations for report designs, images in reports, and log files, the BIRT report viewer uses context parameters defined in the web.xml file. The path provided as the value for any of these parameters can be relative or absolute. A relative path is relative to the root folder of the BIRT report viewer application. A path to a writable location for a BIRT report viewer that is deployed as a WAR file must be an absolute path.

By default, the relative path for report designs is relative to the BIRT report viewer's root folder. Place all report designs in this folder or use the full path to the report design in the URL. Using a relative path is not convenient for deployment of the BIRT report viewer in a WAR file as changes to report designs would require repackaging the WAR file. To set a different location for report designs, change the BIRT_VIEWER_WORKING_FOLDER parameter in the BIRT report viewer application's web.xml file.

Other context parameters determine other aspects of the behavior of the BIRT report viewer, such as the default locale and the level of detail in the log files.

How to set the location for report designs

- 1 Navigate to `$TOMCAT_INSTALL/webapps`.
- 2 Open `web.xml` in a code editor by performing one of the following steps, based on your deployment configuration:
 - If you use a WAR file to deploy the BIRT report viewer, extract `WEB-INF/web.xml` from `birt.war` into a temporary location.
 - If you use a folder to deploy the BIRT report viewer, navigate to `<context root>/WEB-INF`.

- 3 Locate the following element:

```
<context-param>
  <param-name>BIRT_VIEWER_WORKING_FOLDER</param-name>
  <param-value></param-value>
</context-param>
```

- 4 Change the `param-value` element, so that it includes the absolute path to the folder for the report designs, similar to the following code where *Report_Folder* is the absolute path to the folder for the report designs:

```
<context-param>
  <param-name>BIRT_VIEWER_WORKING_FOLDER</param-name>
  <param-value>Report_Folder</param-value>
</context-param>
```

- 5 Save `web.xml` and close the editor.
- 6 If you use a WAR file to deploy the BIRT report viewer, replace `WEB-INF/web.xml` in `birt.war` with the file just modified.
- 7 Copy the report designs into the folder specified in the `param-value` element for `BIRT_VIEWER_WORKING_FOLDER`.
- 8 Restart Apache Tomcat.

Viewing a report using a browser

After deploying the BIRT report viewer to your J2EE container, you can use the two available BIRT report viewer servlets to access your BIRT reports using a web browser. To view a BIRT report using a browser, use a URL in one of the following formats, where *parameter_list* is a list of URL parameters:

```
http://localhost:8080/birt/run?parameter_list
http://localhost:8080/birt/frameset?parameter_list
```

The `run` and `frameset` servlets display reports in two different ways. The `run` servlet displays the report as a stand-alone web page or a PDF file. If the report

requires parameters, specify them in the URL. The frameset servlet displays a page in the browser with a page navigation toolbar and buttons to do the following tasks:

- Display a table of contents.
- Display a parameters dialog.
- Display a dialog for exporting data.
- Display a dialog for exporting the report in various formats.
- Print the report.
- Print the report to a document file on the server.

Using connection pooling

BIRT provides support for connection pooling. For a Tomcat application server with a connection pool configured, BIRT reports can be set up to use a connection from the connection pool when connecting to a JDBC database. A BIRT JDBC data source uses the JNDI URL property to access the connection pool service on the web application server to get a connection from the pool.

Setting up a report to use connection pooling

Use BIRT Report Designer to configure reports to use connection pooling. The BIRT JDBC data source wizard requires configuring a direct-access connection as well as the JNDI URL. The reason for this requirement is that some JNDI service providers do not support client-side access. During design time, such JDBC drivers use the direct-access JDBC connection. The JDBC data-set query builder uses the direct JDBC connection to obtain its metadata.

In BIRT Report Designer, only the design functions directly related to a data-source design, such as Test Connection and Preview Results of a data set, attempt to use a JNDI name path. If the JNDI connection fails for any reason, the data source reverts to using the JDBC driver direct-access URL.

Similarly, at report run time, such as during report preview, the JDBC run-time driver attempts to look up its JNDI data source name service to get a pooled JDBC connection. If such look-up is not successful for any reason, the JDBC driver uses the direct-access URL to create a JDBC connection.

Using a jndi.properties file

Each individual JNDI application on the web application server uses its own environment settings stored in the JVM system properties. The JNDI reads the following standard JNDI properties from the system properties:

```
java.naming.factory.initial
java.naming.factory.object
java.naming.factory.state

java.naming.factory.control
java.naming.factory.url.pkgs
java.naming.provider.url
java.naming.dns.url
```

To simplify the task of setting up the JNDI initial context environment for an individual JNDI application, the JNDI feature supports the use of a `jndi.properties` resource file. Install this file in the `drivers` subfolder of the `oda.jdbc` plug-in located at the following path:

```
WEB-INF\platform\plugins
  \org.eclipse.birt.report.data.oda.jdbc_<version>\drivers
```

This file contains a list of key-value pairs in the properties file format, `key=value`. The key is the name of the property, and the value is a string, for example, `java.naming.factory.object=jnp://localhost:1099`.

Here is an example of a JNDI resource file used with JBoss application server:

```
java.naming.factory.initial=
  org.jnp.interfaces.NamingContextFactory
java.naming.provider.url=jnp://localhost:1099
java.naming.factory.url.pkgs=org.jboss.naming:org.jnp.interfaces
```

The JDBC run-time driver looks for the `jndi.properties` file in the web application's folder tree. If the driver does not find the file or has a problem reading from it, the initial context uses the default behavior, as defined by `javax.naming.Context`, to locate any JNDI resource files. Configuring the classpath for classes referenced by the environment properties is necessary.

Configuring a JNDI connection object

The JNDI URL property for the JDBC data source supports retrieving a JDBC connection from a pool when BIRT reports are deployed to a web application server. More information about configuring connection pooling on Tomcat is available at:

<http://tomcat.apache.org/tomcat-6.0-doc/jndi-resources-howto.html>

How to configure a JNDI connection object

The following example assumes you already have deployed the BIRT report viewer to a Tomcat 7.0 application server in the folder, `$TOMCAT_INSTALL/webapps/birt`, as described earlier in this chapter.

- 1 Install the JDBC Driver. Make an appropriate JDBC driver available to both Tomcat internal classes and the web application, for example, by installing the

driver's JAR files into the following library directory in the Tomcat application server home folder:

```
$CATALINA_HOME/common/lib
```

- 2 Declare the resource requirements in the BIRT report viewer's WEB-INF/web.xml file. For example, add the following entry to set up a JNDI service for a MySQL format database with the name, MySqlDB:

```
<resource-ref>
  <description>Resource reference to a factory for
    java.sql.Connection</description>
  <res-ref-name>jdbc/MySqlDB</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <res-auth>Container</res-auth>
</resource-ref>
```

- 3 Configure the Tomcat resource factory as a Resource element in the BIRT report viewer's META-INF/context.xml file, similar to the following lines:

```
<Context>
  <Resource name="jdbc/MySqlDB" auth="Container"
    type="javax.sql.DataSource" maxActive="5" maxIdle="-1"
    maxWait="10000" username="root" password="password"
    driverClassName="com.mysql.jdbc.Driver"
    url="jdbc:mysql://localhost:3306/classicmodels"
    description="MySQL DB"/>
</Context>
```

- 4 Make the JNDI URL in your report design match the Tomcat resource factory, similar to the following line:

```
java:comp/env/jdbc/MySqlDB
```

- 5 Open the report design using BIRT Report Designer. Edit the data source. In Edit Data Source, in JNDI URL, type the URL, as shown in Figure 4-3.

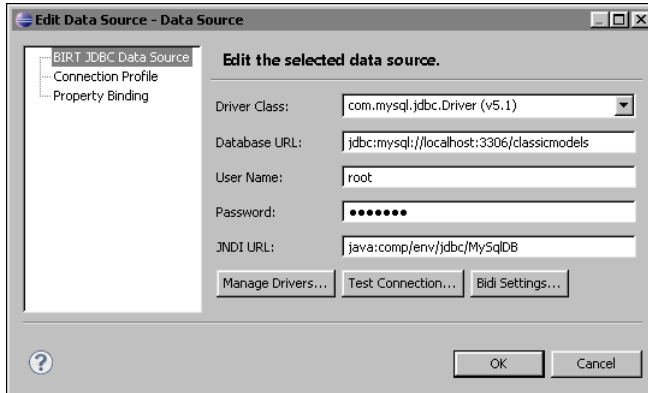


Figure 4-3 Setting the JNDI URL for a JDBC data source

- 6 Copy the report design to the BIRT report viewer root folder.
- 7 Restart the Tomcat service.
- 8 Run the report using a URL similar to the following one:

`http://localhost:8080/birt/run?__report=myJNDIReport.rptdesign`

The report uses a connection from the connection pool to connect to the database on a MySQL server.

Index

Symbols

" (double quotation mark) character
command line arguments and 19

A

absolute paths 29
access rights
 See also privileges
accessing
 BIRT packages 14
 connection pool services 31
 demo database 16, 17
 report viewer 25, 27, 28
 reports 28
 Tomcat Manager 27
ACL files
 See also access control lists
ACLs. *See* access control lists
ACS. *See* Caching service
adding
 display names. *See* display names
Administrative operations
 See also administration operations
administrators
 See also administration operations
aggregate data. *See* aggregation
aggregate functions. *See* aggregation
 functions
aging rules. *See* archiving rules
AIS. *See* Integration service
Apache Derby databases 16, 17
Apache Tomcat manager 27
Apache Tomcat server connections 31, 32
Apache Tomcat servers 24, 27
application programming interfaces (APIs)
 See also Information Delivery API
application server connections 31, 32
application servers 3, 24
applications
 accessing report viewer for 28
 building user interfaces for. *See* user
 interfaces

 deploying 24
 developing IDAPI. *See* IDAPI applications
 developing RSSE. *See* RSSE applications
 developing web. *See* web applications
 setting up JNDI connections for 31, 32
archive files 6, 7, 8
 See also jar files; war files
 BIRT Demo Database 16
 BIRT packages in 14
 chart engine 14
archive rules. *See* archiving rules
ArchiveRule objects
 See also archiving rules
arguments. *See* command line arguments;
 parameters
attributes
 See also properties
autoarchiving. *See* archiving operations
Axis servers. *See* Apache Axis environments
axis.jar 24
axis-ant.jar 24

B

Babel Language Packs 9
backing up program files 11
BEA WebLogic servers 24
beans. *See* JavaBeans
BIRT Data Analyzer. *See* Data Analyzer
BIRT Data Tools Platform Integration
 package 15
BIRT Demo Database 16
BIRT Demo Database package 4, 16
BIRT design files
 See also design files
BIRT designer components 2
BIRT documentation 15
BIRT iHub. *See* iHub System
BIRT packages 3, 14
BIRT RCP Report Designer
 accessing sample database for 16, 17
 adding charting functionality to 14
 building reports and 2
 installing 7–8

- specifying JVM for 9
- starting 8
- testing installations for 8
- troubleshooting 8
- updating 10, 11
- BIRT Report Designer
 - accessing sample database for 16, 17
 - adding charting functionality to 14
 - building reports and 2
 - installing 6
 - specifying JVM for 9
 - starting 7
 - testing installations for 7
 - troubleshooting 8
 - updating 10–11
- BIRT Report Designer Full Eclipse Install software 3, 6
- BIRT report files
 - See also* report files
- BIRT reports
 - See also* reports
- BIRT Samples package 4, 19
- BIRT Source Code package 3, 19
- BIRT Test Suites package 4
- BIRT Viewer
 - See also* report viewer
- BIRT Web Tools Integration package 21
- BIRT_FONT_PATH variable 29
- BIRT_VIEWER_WORKING_FOLDER
 - parameter 28, 29
- BIRT.exe 8
- birt.war 25
- browsers. *See* web browsers
- build files 20
- builds 4
- Business Intelligence and Reporting Tools.
 - See* BIRT

C

- cache conflicts 8
- cache database. *See* Caching service database
- calculated columns
 - See also* computed columns
- changing
 - URL context roots 28
- character data. *See* strings
- character encoding. *See* encoding
- character encryption. *See* encryption
- character strings. *See* strings
- chart engine
 - installing 14–15
- chart engine archive 14
- chart engine documentation 15
- Chart Engine package 3
- chart objects
 - See also* charts
- chart package 14
- chart wizard launcher
 - See also* chart builder
- charting application sample plug-in 19
- charting applications
 - adding reporting functionality to 14
 - developing 15
- ChartRuntime directory 15
- charts
 - See also* Flash charts; HTML5 charts
 - developing. *See* charting APIs
- Charts_FAQ.doc 15
- ChartSDK directory 15
- Classic Models, Inc. sample database 16
- ClassicModels directory 16
- classpaths
 - JNDI resource files and 32
- clean command line option 8, 11
- client applications. *See* applications
- code
 - importing 3
 - installing 19–20
- code archives 20
- code editors 30
- column headers
 - See also* column names
- column headings
 - See also* column names
- comma-separated values files. *See* CSV files
- command line applications 18
- command line arguments 19
- completion notices
 - See also* notifications
- computed columns
 - See also* calculated columns
- conditions. *See* filter conditions; search conditions

- configuration files 30
- configuring
 - connection pooling services 31, 32
 - report viewer 28, 29
- connection definition files. *See* database connection definition files
- connection definition files
- connection handles. *See* ConnectionHandle element
- connection pooling 31
- connections
 - application servers and 31, 32
 - JDBC data sources and 31, 32
 - JNDI service providers and 31, 32
 - setting properties for. *See* connection properties
- consolidator application. *See* log consolidator application
- context mapping 28, 29
- context parameters 29
- context root 28
- context-param element 30
- create_classicmodels.sql 17
- creating
 - Classic Models, Inc. database 17
 - display names. *See* display names
 - IDAPI applications. *See* applications
- CSS files
 - See also* cascading style sheets

D

- data
 - aggregating. *See* aggregation
 - extracting. *See* data extraction operations
 - localizing. *See* locales
- Data Analyzer
 - adding toolbars. *See* Data Analyzer toolbars
- data analyzer component
 - See also* Data Analyzer
- data charts viewer
 - See also* charts
- data cubes. *See* cubes
- data elements
 - See also* data items
- data fields
 - See also* columns
- data filters. *See* filters
- data items
 - See also* data
- data repositories
 - See also* volumes
- data rows. *See* rows
- data set fields. *See* fields
- data sets
 - retrieving from JDBC data sources 31
- data sorters. *See* sorters
- Data Tools Platform 2
- database connection properties. *See* connection properties
- database drivers. *See* drivers
- database schemas. *See* schemas
- databases
 - See also* data sources
- DCD. *See* database connection definitions
- default locale 18
- deleting
 - cached information 8
 - program files 10, 11
- demo database 16
- Demo Database archive 16
- Demo Database package 4, 16
- dependent files. *See* file dependencies
- deploying
 - applications 24
 - report viewer 24, 29
 - reports 24
- DeploymentRuntime directory 15
- Derby databases 16, 17
- design files
 - setting location of 29
- design perspective 6, 7
- design tools 2
- designers 2
- designs
 - deploying to web servers 26
 - installing report viewer and 29
- developing
 - charts. *See* charting APIs
 - IDAPI applications. *See* IDAPI applications
 - RSSE applications. *See* RSSE applications
- diagnostic information
 - See also* Ping operations
- directories

- accessing report designs and 28
- displaying reports and 25
- extracting chart engine package and 15
- installing language packs and 10
- installing report viewer and 25, 26, 28
- unpacking program archives and 8
- updating designers and 11
- viewing PDF documents and 29

directory paths. *See* paths

display formats. *See* formats

displaying

- HTML output 31
- PDF documents 29, 30
- reports 27, 30, 31

distributed iHub System. *See* clusters

documentation v, 15

- See also* help collections

documents

- See also* reports

double quotation mark (") character

- command line arguments and 19

.dov files. *See* data object values files

download operations

- See also* downloading

downloading

- See also* download operations
- BIRT Full Eclipse Install 6
- BIRT RCP Report Designer 7
- BIRT Report Designer packages 14
- chart engine archive 14
- demo database archive 16
- JDK software 6
- language packs 9
- report engine archive 17
- Samples archive 19
- source code 20
- Web Tools Integration archive 21

driver manager 8

drivers

- connecting to JDBC data sources and 32
- designing reports and 31
- installing report viewer and 25, 26
- locating JNDI resource files for 32
- using nonstandard 25

drivers directory 26

DTP Integration package 15

duplicating. *See* copying

E

- e command line option 18
- Eclipse 8, 10
- Eclipse BIRT packages 3, 14
- Eclipse Modeling Framework 2
- Eclipse Software Development Kit 2
- elements. *See* report elements; XML elements
- e-mail
 - sending attachments with. *See* attachments
 - setting notification options for. *See* notifications
- EMF software 2
- environment settings 31
- errors 8
- events
 - handling. *See* event handlers
- example charting applications 15
- example database 16
- execution mode parameter 18
- execution requests. *See* ExecuteReport operations
- extensible markup language. *See* XML

F

- F command line option 18
- f command line option 18
- fields
 - See also* columns
- file attributes
 - See also* file properties
- file IDs
 - See also* FileId element
- file names 18
- file paths. *See* paths
- file systems 26
- files
 - See also* report files
 - downloading designer 6, 7
 - extracting program archive 6, 7, 8
 - mapping to report viewer 29
 - naming. *See* file names
 - removing program 10, 11
 - setting properties for. *See* file properties
 - updating designers and 10
- finding data. *See* search operations
- folder paths. *See* paths

- font files 29
- fonts 29
- formats
 - See also* output formats
- frameset servlet 31
- Full Eclipse Install 6–7
- Full Eclipse Install package 3, 6
- functions
 - See also* callback functions; methods

G

- GEF plug-in 2
- genReport script 18
- Graphical Editing Framework plug-in 2
- graphical user interfaces. *See* user interfaces
- graphics elements
 - See also* images
- graphs. *See* charts
- groups
 - See also* notification groups; resource groups
- GUI components
 - See also* user interfaces

H

- header elements (SOAP messages)
 - See also* SOAP headers
- HTML format parameter 18
- HTML reports
 - displaying 31
- HTML value 18
- hyperlinks
 - See also* URLs
- hypertext markup language. *See* HTML code
- hypertext transfer protocol. *See* HTTP

I

- IBM WebSphere servers 24
- IDAPI applications
 - See also* Information Delivery API
- iHub clusters. *See* clusters
- iHub repository. *See* volumes
- iHub services
 - See also* specific iHub service
- image files 29
- importing

- source code 3
- source code projects 20
- index.jsp 26
- Information Console Security Extension
 - See also* IPSE applications
- Information Delivery API
 - See also* IDAPI applications
- input file IDs. *See* InputFileId element
- input file names. *See* InputFileName element
- input messages
 - See also* requests
- installation
 - BIRT Full Eclipse Install 6–7
 - BIRT RCP Report Designer 7–8
 - BIRT Report Designer 6
 - chart engine 14–15
 - demo database 16
 - DTP Integration package 15
 - language packs 9–10
 - report viewer 25–27, 28
 - Samples package 19
 - source code 19–20
 - testing 7, 8
 - troubleshooting 8–9
 - TrueType fonts 29
 - upgrades and 10
 - WTP plug-ins 21
- Integration Technology. *See* iHub Integration Technology
- interfaces
 - See also* user interfaces
- iPortalSecurityAdapter class
 - See also* IPSE applications
- iServer System. *See* iHub System

J

- J2EE application servers 3
- J2EE applications 30
- J2EE environments 14, 17
- .jar files 8
 - accessing demo database and 17
 - creating JNDI connections and 32
 - deploying to web servers and 24
 - installing report viewer and 26
- Java applications
 - adding charting capabilities to 14

- adding reporting capabilities to 17
- Java code 24
- Java RSSE framework
 - See also* RSSE applications
- JBoss servers 24
- JDBC connections 32
- JDBC data sets 31
- JDBC data sources 31
- JDBC driver manager 8
- JDBC drivers
 - adding nonstandard 25
 - configuring JNDI service providers for 31, 32
- JDK software 3, 6
- JNDI connection objects 32
- JNDI connection properties 31, 32
- JNDI name paths 31
- JNDI resource files 32, 33
- JNDI service providers 31, 32
- JNDI URL property 31, 32
- jndi.properties file 32
- jobs
 - failing. *See* failed jobs
 - pending. *See* pending jobs
 - print operations and. *See* print jobs
 - sending notifications for. *See* notifications
- JSPs 27
- JVMs 9

L

- l locale command line option 18
- language packs 9–10
- libraries
 - running charting applications and 15
- Lightweight Directory Access Protocol. *See* LDAP servers
- links (Information Console)
 - See also* hyperlinks
- Linux servers 9, 14, 17
 - See also* UNIX systems
- load_classicmodels.sql 17
- locale codes 10
- Locale parameter 18
- locales
 - installing language packs for 9–10
- localization 9

- log files 29
 - tracking error information and. *See* error log files
 - tracking usage information and. *See* usage log files
- Login operations
 - See also* SystemLogin operations

M

- m command line option 18
- mail. *See* e-mail
- mapping to report viewer 28, 29
- MDS. *See* Message Distribution service
- messages. *See* e-mail
- metadata
 - building JDBC data sets and 31
 - metadata schemas. *See* schemas
- methods
 - See also* functions
- Microsoft NET environments. *See* .NET environments
- Microsoft Windows. *See* Windows systems
- milestone build 4
- missing designer functionality 8
- monitoring tools
 - See also* performance monitoring
- multilingual reports. *See* locales
- mysql command line interface 17
- MySQL databases 16, 17

N

- names
 - See also* user names
- naming restrictions. *See* case sensitivity
- nl command line option 10
- nodes. *See* cluster nodes
- non-native reports. *See* third-party reports
- nonstandard drivers 25
- notifications
 - sending attachments with. *See* attachments

O

- o command line option 18
- object IDs
 - See also* ObjectId element

- on-demand report generation. *See*
 - synchronous jobs
- online analytical processing servers. *See*
 - OLAP servers
- online documentation v
- online help
 - See also* help
- Open Perspective command 7
- opening
 - configuration files 30
 - Tomcat Manager 27
- operations
 - administration. *See* Administrate operations
 - archiving files and. *See* archiving operations
 - login. *See* Login operations
 - searching. *See* search operations
 - updating files and. *See* update operations
- output
 - formatting. *See* output formats
 - viewing HTML 18, 31
- output file name parameter 18
- output format parameter 18
- output messages
 - See also* responses
- overriding report parameters 18

P

- p command line option 18
- packages 3, 14
- page-level security
 - See also* page security application
- parameter files 18
 - See also* data object values files; report object value files
- parameter values files. *See* data object values files; report object value files
- parameters
 - See also* report parameters
 - configuring report viewer and 28, 29
 - defining a dynamic filter. *See* dynamic filter parameters
 - viewing reports and 30
- param-value element 30
- paths

- font files 29
- JNDI resource files 32
- report design files 28
- report viewer configurations 29
- PDF documents
 - displaying 29, 30
 - installing fonts for 29
 - missing content in 29
- permissions. *See* privileges
- perspectives 6
- pick lists. *See* selection lists
- plug-in extensions. *See* extensions
- plug-ins 8
 - downloading source code for 20
 - installing 21
- PMD. *See* Process Management Daemon
- PPT formats. *See* PowerPoint formats
- preferences (users). *See* user preferences
- print jobs
 - See also* printing
- print requests. *See* print jobs
- printer settings. *See* printer options
- printing requests. *See* print jobs
- product upgrades 10
- projects
 - importing 20
- properties
 - JNDI connection 31, 32
 - .properties files 9
- purging. *See* deleting

Q

- queries
 - retrieving data with 31

R

- RCP Report Designer package 3
 - See also* BIRT RCP Report Designer
- records
 - See also* rows
- referencing
 - report designs 30
- relative paths 28, 29
- release build 4
- removing. *See* deleting
- report components. *See* components

- report design engine classes
 - See also* Design Engine API
- report design file parameter 18
- report design files
 - setting location of 29
- Report Design perspective 5, 7
- report design tools 2
- Report Designer Full Eclipse Install package 3, 6
- Report Designer package 3
- report designers 2
- report designs
 - deploying to web servers 26
 - installing report viewer and 29
- report documents
 - See also* reports
- Report Encyclopedia. *See* volumes
- report engine
 - adding fonts for 29
 - installing 17
 - testing installations for 18
- report engine archive 17
- Report Engine package 3
- report engine report generation example 18
- report execution requests. *See* ExecuteReport operations
- report explorer. *See* ReportExplorer components
- report files
 - See also* files; specific report file type naming. *See* file names
- report generation example 18
- report item extension sample plug-in 19
- report objects. *See* reports
- report parameter files 18
 - See also* data object values files; report object value files
- report parameters
 - See also* parameters
 - adding to URLs 30
 - overriding 18
 - restricting values for. *See* cascading parameters
- Report Server Security Extension
 - See also* RSSE applications; RSSE API
- report servers. *See* iHub servers
- report viewer
 - accessing 25, 27, 28
 - changing context root for 28
 - checking status of 27
 - deploying 24, 29
 - displaying reports and 27, 30
 - displaying web pages and 28
 - installing 25–27, 28
 - mapping to 28, 29
 - referencing report designs for 30
 - running on application servers 24, 25
 - setting context parameters for 30
 - starting 25
 - testing installations for 26
 - troubleshooting Apache servers for 27
- report viewer servlets 30
- reporting system. *See* iHub System
- ReportletNoCSS value 18
- reports
 - accessing 28
 - deploying 24
 - displaying 27, 30, 31
 - selecting language for 10
 - sending as attachments. *See* attachments
 - testing report viewer for 26
- repositories
 - See also* volumes
- requests
 - See also* SOAP messages
- resource factory 33
- resource files 32
- resources 11
 - JNDI connections and 33
- responses
 - See also* SOAP messages
- result sets
 - See also* queries; search results
- .rov files. *See* report object value files
- RPCs. *See* remote procedure calls
- rptdesign format
 - See also* report design files
- rptdocument format
 - See also* report document files
- RSSE applications
 - See also* Report Server Security Extension
- registering external users for. *See* external user registration
- rules. *See* archiving rules

run requests. *See* report generation requests
run servlet 30

S

sample charting applications 15
sample database 16
Samples archive 19
Samples package 19
scheduled jobs
 See also jobs
schemas
 See also WSDL schemas
SDO Runtime component 2
search criteria. *See* search conditions
search operations
 See also searching
 setting conditions for. *See* search conditions
search paths 29
searching
 See also search operations
security credentials. *See* credentials
security roles. *See* roles
server connections 31, 32
servers 24
 See also iHub servers
Service Data Objects component 2
services
 See also iHub services; web services
settings. *See* properties
simple object access protocol. *See* SOAP
SOAP endpoints
 See also SOAP ports
SOAP requests. *See* requests
SOAP responses. *See* responses
Software Development Kit 2
 See also SDK package
Software Development Kit package 3
sort fields. *See* sort columns
source code
 importing 3
 installing 19–20
source code archives 20
source code editors 30
spreadsheets. *See* Excel spreadsheets
SQL statements

JDBC data sources and 31
SQL statements. *See* queries
stable build 4
stand-alone web pages 30
starting
 BIRT RCP Report Designer 8
 BIRT Report Designer 7
 Eclipse 8, 10
 report viewer 25
subdirectories. *See* subfolders
system administrators
 See also administrators
system schemas. *See* schemas

T

-t command line option 18
tab-separated values files. *See* TSV files
target encoding parameter 18
tcpmon utility. *See* TCPMonitor
temporary files. *See* transient files
temporary reports. *See* transient reports
test.rptdesign 27
testing
 BIRT installations 7, 8
 Demo Database installations 16
 DTP Integration installation 16
 report engine installations 18
 report viewer installations 26
 source code package 20
 Web Tools Integration archive 21
text strings. *See* strings
Tomcat manager accounts 27
Tomcat server connections 31, 32
Tomcat servers 24, 27
toolbars 31
transactions
 See also Transaction operations
troubleshooting installation 8–9
TrueType fonts 29
types. *See* data types

U

UI elements
 See also user interfaces
Uniform Resource Locators. *See* URLs
universal hyperlinks. *See* hyperlinks

Universal Resource Identifiers. *See* URIs

UNIX platforms 9

UNIX systems 14, 17

unpacking BIRT archives 6, 7, 8

Update Manager 10

updating

report designers 10–11

upgrades 10

URLs

accessing reports and 28, 30

adding report parameters to 30

JNDI connections and 31, 33

localized reports and 10

report viewer and 26

user groups. *See* groups

user IDs

See also UserId element

V

values

See also data

overriding parameter 18

version names

See also VersionName element

view parameters

See also ViewParameter element

viewing

HTML output 31

PDF documents 29, 30

reports 27, 30, 31

viewing parameters. *See* view parameters

viewing preferences. *See* viewer preferences

viewing service. *See* View service

Vista computers. *See* Windows systems

volume administrators. *See* administrators

volume schemas. *See* schemas

W

.war files

report viewer deployment 29, 30

report viewer installations 25, 28

Tomcat servers and 24

report engine installations 17

web applications

See also applications

accessing report viewer for 28

adding charting capabilities to 15

installing report viewer for 25, 26

web browsers 30

web pages

displaying 27, 30

web servers 24

web service applications

See also IDAPI applications

Web Service Description Language. *See*

WSDL

web services messaging framework. *See*

SOAP

Web Tools Integration archive 21

Web Tools Integration package 3, 21

Web Tools Platform 3

Web Tools Platform extension 15

web.xml 24

webapps directory 25

WebLogic servers 24

WebSphere servers 24

WebViewerExample folder 26

Windows platforms 9, 10

Windows systems 14, 17

workspace directory 11

WSDL documents

See also WSDL files

WSDL elements

See also XML elements

WSDL2Java package

See also code emitter

WTP plug-ins 3, 21

X

XML attributes. *See* attributes

XML code

See also code

XML reports. *See* XML documents

XML schemas

See also WSDL schemas