

ActuateOne™

One Design
One Server
One User Experience

Deploying to a BIRT iServer System

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

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

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

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

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

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

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

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

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

Document No. 110812-2-430360 July 28, 2011

Contents

About *Deploying to a BIRT iServer System* v

Part 1

Introduction to deploying reports and information objects

Chapter 1

Deployment overview 3

Deployment overview 4

 Deploying reports 4

 Deploying information objects 4

About accessing other types of data sources 5

Using a connection configuration file 5

Part 2

Deploying BIRT reports

Chapter 2

Deploying BIRT reports to BIRT iServer 9

About deploying BIRT reports 10

Publishing a report to Actuate BIRT iServer 10

Publishing a report resource to Actuate BIRT iServer 14

Deploying Java classes used in BIRT reports 15

Installing a custom JDBC driver 17

Installing custom ODA drivers and custom plug-ins 17

Configuring the BIRT design cache 18

Chapter 3

Connecting to data sources 21

About data source connections 22

About data source connection properties 22

 Using a connection configuration file 22

 Changing a configuration file 23

 Specifying the location of the connection configuration file 24

 Encrypting the connection properties 24

 Configuring a cluster to use a connection configuration file 25

 Using a connection profile 25

Externalizing the connection profile properties on the iServer	26
Understanding externalization precedence	26
Referencing the external connection profile	27
Chapter 4	
Configuring fonts in BIRT iServer	29
About configuring fonts	30
Understanding font configuration file priorities	30
Understanding how BIRT engine locates a font	31
Understanding the font configuration file structure	32
<font-aliases> section	32
<composite-font> section	33
<font-paths> section	33
Chapter 5	
Working with BIRT encryption	35
About BIRT encryption	36
About the BIRT default encryption plug-in	36
About acdefaultsecurity.jar	37
About encryption.properties	37
About META-INF/MANIFEST.MF	37
About plugin.xml	37
Deploying encryption plug-ins to iServer	38
Chapter 6	
Using custom emitters	39
About custom emitters	40
Deploying custom emitters to iServer	41
Rendering in custom formats	41
Part 3	
Deploying other reports and information objects	
Chapter 7	
Deploying e.reports, spreadsheet reports, and information objects . . .	49
Deploying e.reports	50
Publishing a report object executable (.rox) file	50
Defining a database connection	52
Making fonts available to e.reports	52
Specifying fonts and parameters for PDF generation	53
Synchronizing the localemap.xml files	53
Deploying spreadsheet reports	54

Deploying a database driver 54
Installing fonts 54
Deploying a callback class 54
 Deploying classes using a JAR file 54
 Deploying classes by creating new directories 55
Publishing a report with a VBA template 55
Deploying information objects 55
Index 59

About Deploying to a BIRT iServer System

Deploying to a BIRT iServer System provides information for volume administrators, report developers, and data modelers regarding deploying reports and information objects to an iServer Encyclopedia volume.

Deploying to a BIRT iServer System includes the following parts and chapters:

- *About Deploying to a BIRT iServer System*. This chapter provides an overview of this guide.
- *Part 1. Introduction to deploying reports and information objects*. This part describes the general procedures for deploying reports and information objects to an Encyclopedia volume.
- *Chapter 1. Deployment overview*. This chapter describes ODA drivers and the connection configuration file.
- *Part 2. Deploying BIRT reports*. This part describes how to deploy BIRT reports to an Encyclopedia volume.
- *Chapter 2. Deploying BIRT reports to BIRT iServer*. This chapter describes how to publish BIRT reports and report resources to an Encyclopedia volume, how to deploy Java classes, and how to install JDBC and ODA drivers and plug-ins.
- *Chapter 3. Connecting to data sources*. This chapter describes how to use a connection profile, how to define environment variables, and how to connect to various data sources.
- *Chapter 4. Configuring fonts in BIRT iServer*. This chapter describes how to configure fonts for use with BIRT reports.
- *Chapter 5. Working with BIRT encryption*. This chapter describes how to deploy encryption plug-ins.
- *Chapter 6. Using custom emitters*. This chapter describes how to deploy custom emitters.

- *Part 3. Deploying other reports and information objects.* This part describes how to deploy e.reports, spreadsheet reports, and information objects.
- *Chapter 7. Deploying e.reports, spreadsheet reports, and information objects.* This chapter describes the specific procedures for deploying e.reports, spreadsheet reports, and information objects to an Encyclopedia volume, including defining database connections and making fonts available to reports.

Part One

**Introduction to deploying reports
and information objects**

Deployment overview

This chapter contains the following topics:

- Deployment overview
- About accessing other types of data sources
- Using a connection configuration file

Deployment overview

Actuate users work with reports and information objects when they log in to an iServer Encyclopedia volume using Information Console. The volume administrator is responsible for working with the developer to deploy reports and information objects to the Encyclopedia volume and make them available to users.

Deploying reports

Actuate offers several report designers including:

- BIRT Designer Professional
- BIRT Spreadsheet Designer
- BIRT Studio
- e.Report Designer Professional

BIRT Studio is a web-based report design tool for business users. With BIRT Studio, a user designs a report and saves the report design in the Encyclopedia volume. It is not necessary to publish the report design to the Encyclopedia volume. For more information about designing reports with BIRT Studio and integrating BIRT Studio with iServer, see *Using BIRT Studio - iServer Edition*.

The other report designers reside on the desktop. A report developer designs a report in one of these report designers and publishes the report executable to the Encyclopedia volume. The volume administrator then performs the following tasks:

- Places the report executable in the appropriate folder
- Schedules a job to generate the report document in the appropriate format, for example PDF
- Assigns privileges on the report document for users and roles
- Notifies users that the report document is available

For more information about these tasks, see *Managing an Encyclopedia Volume*.

Before the volume administrator makes reports available to users, he may need to perform additional tasks, for example, define database connections and make fonts available to reports. These tasks are discussed in the following chapters.

Deploying information objects

An information object can be used as a data source in any type of report design. To deploy an information object, you must publish the information object project to the appropriate Encyclopedia volume. If you want users to be able to execute

an information object without having execute privilege on its sources, apply the trusted execute privilege to the information object.

About accessing other types of data sources

A report or information object developer can access a variety of data sources using predefined data drivers. To access other types of data, you can create a custom data driver, known as an open data access (ODA) driver.

An ODA driver supports both design-time and run-time functionality. The report or information object designer uses the driver to build a connection to the data source, retrieve parameter and data row definitions, and compile these definitions for use at run time. At run time, BIRT iServer loads the driver. Then, the driver creates the connection and data source instance and fetches the requested data.

Each ODA driver supports one type of connection and can support multiple instances of that connection type. Each driver can support multiple data sources. The driver must be installed on the system where you design the report or information object and on BIRT iServer.

For more information about installing ODA drivers on BIRT iServer, see *Configuring BIRT iServer*.

Using a connection configuration file

Connection configuration files enable deployment of reports and information objects to a production environment. You can use a connection configuration file to specify which data connections to use in the design environment. You can use the same data connections or specify different connections for use when BIRT iServer runs the report or information object in the production environment. By specifying both design-time and run-time connections, you do not have to change the report design or information object when deploying to a production environment. For more information about connection configuration files, see the documentation for the report or information object designer and *Configuring BIRT iServer*.

Part **Two**

Deploying BIRT reports

Deploying BIRT reports to BIRT iServer

This chapter contains the following topics:

- About deploying BIRT reports
- Publishing a report to Actuate BIRT iServer
- Publishing a report resource to Actuate BIRT iServer
- Deploying Java classes used in BIRT reports
- Installing a custom JDBC driver
- Installing custom ODA drivers and custom plug-ins
- Configuring the BIRT design cache

About deploying BIRT reports

This chapter describes how to run and distribute BIRT reports in the Actuate business reporting system. To be successful in deploying BIRT reports to Actuate BIRT iServer, you need to understand the environment in which the reports run.

Actuate Information Console provides a central location from which business users can access, run, and view reports. You can use Actuate Information Console to run report executables, and to manage, generate, view, and print report documents.

Actuate Information Console is available in iServer. iServer is a report server that can store and manage the scheduling, versioning, and archiving of large numbers of reports. iServer uses the Actuate BIRT option to run and distribute reports.

BIRT Designer Professional, the tool that you use to develop BIRT reports, has built-in capabilities that facilitate the deployment process. BIRT Designer Professional integrates with iServer in several important ways to support performing the following tasks:

- Publish a report design to an Encyclopedia volume.
- Publish a resource to an Encyclopedia volume.
- Install a custom JDBC driver for use by BIRT reports running in the iServer environment.

The following sections describe these capabilities.

Publishing a report to Actuate BIRT iServer

The purpose of publishing a report to Actuate BIRT iServer is to make it accessible to a large number of users. You can schedule re-running the report to include updates from the data sources. You can choose who can access part or all of the report.

Actuate BIRT Designer Professional provides tools for easy deployment of reports, templates and their resources to iServer. The designer connects directly to an iServer and deploys the reports to selected iServer folders. The designer embeds an iServer Explorer view for managing the iServer connections. Using the iServer Explorer you can create iServer connection profiles to store the connection properties to a specific Encyclopedia volume. The iServer Explorer view with an iServer profile is shown in Figure 2-1.

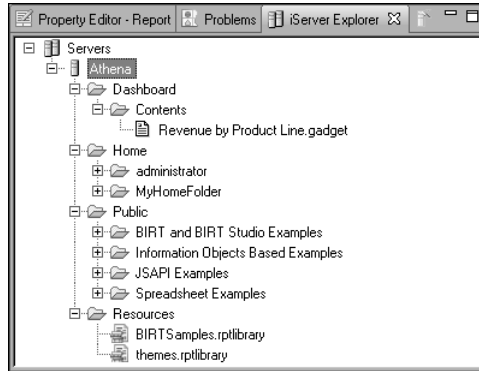


Figure 2-1 iServer Explorer view

How to create an iServer profile

- 1 In BIRT Designer Professional, open iServer Explorer. If you do not see the iServer Explorer view, choose **Windows**→**Show View**→**iServer Explorer**.
- 2 In iServer Explorer, right-click **Servers**, and choose **New iServer Profile** as shown in Figure 2-2.



Figure 2-2 Creating an iServer profile

- 3 In **New iServer Profile**, specify the connection information. Figure 2-3 displays an example of connection properties used for an iServer with name **Athena**.
 - 1 In **Profile name**, type a unique name that identifies the profile.
 - 2 In **iServer**, type the name or IP address of the computer on which Actuate BIRT iServer is installed.
 - 3 In **Port number**, type the number of the port to access iServer.
 - 4 In **Volume**, select the iServer Encyclopedia volume.
 - 5 In **User name**, type the user name required to access the volume.
 - 6 In **Password**, type the password required to access the volume.
- 4 Choose **Finish** to save the iServer profile. The iServer profile appears in the iServer Explorer as shown in Figure 2-1.

The next procedure describes the steps involved in publishing a report to iServer.

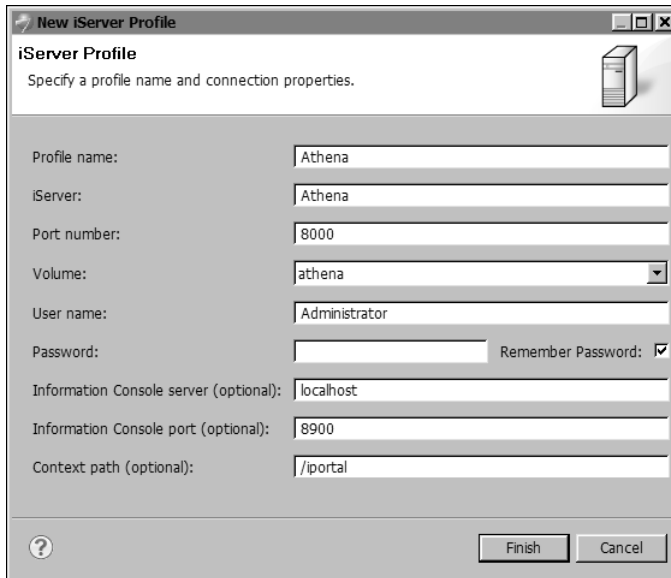


Figure 2-3 Setting properties for an iServer profile

How to publish a report design to iServer

- 1 Choose File→Publish Report to iServer.
- 2 On Publish Report Designs, select the report, as shown in Figure 2-4.

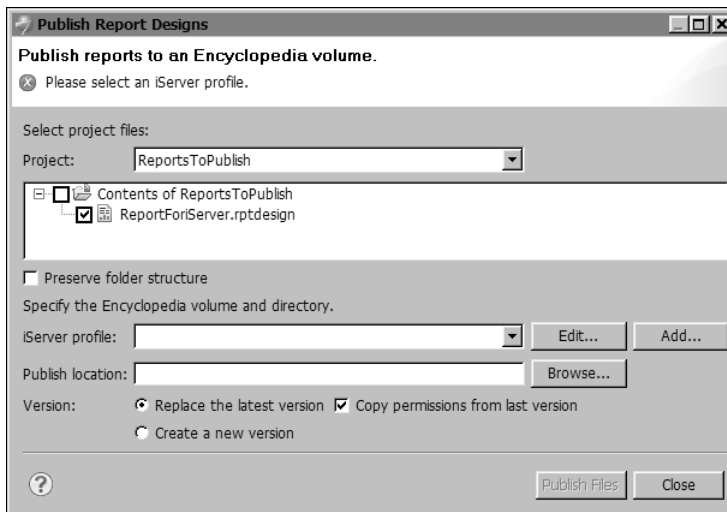


Figure 2-4 Selecting a report to publish

- 3 If there is no appropriate profile in the iServer profile list, create a new profile by choosing Add. Complete the information in New iServer Profile, as shown in Figure 2-3.
- 4 On Publish Report Designs, in Publish location, type or browse for the location in the Encyclopedia volume in which to publish the report design, as shown in Figure 2-5.

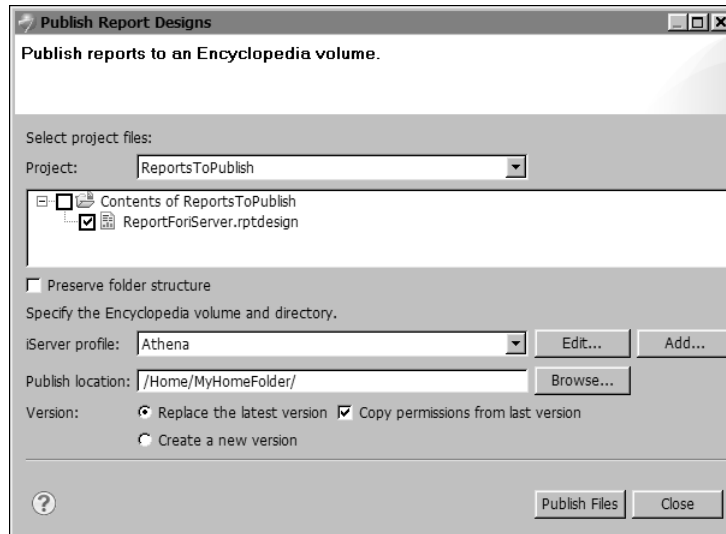


Figure 2-5 Selecting a server and location

- 5 Choose Publish Files. A new window, showing the file upload status, appears. On Publishing, wait until the upload finishes, then choose OK as shown in Figure 2-6.

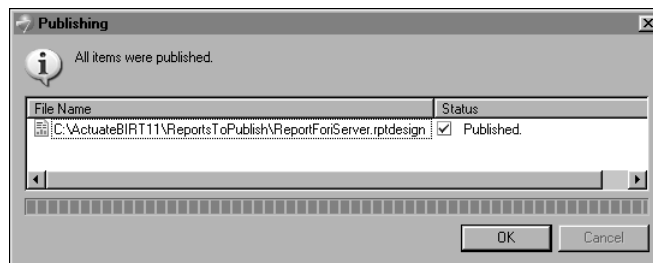


Figure 2-6 Confirming the report publishing

- 6 On Publish Report Designs, choose Close.

Publishing a report resource to Actuate BIRT iServer

In many cases, BIRT reports use additional files called resources. A BIRT resource is any of the following items:

- Static image that a BIRT report design uses
- Report library
- Properties file
- Report template
- CSS file
- External JavaScript file
- SWF file of a Flash object
- Java Event Handler class packaged as a Java Archive (JAR) file

You can publish BIRT resources from BIRT Designer Professional's local resource folder to iServer. By default, the Resource folder is the report design project folder. If you use shared resources with other developers and the resource files for your reports are stored in a different folder, you can change the default Resource folder. Choose **Windows**→**Preferences**→**Report Design**→**Resource** to set the resource folder.

In the Encyclopedia volume, by default, the Resource folder is set to /Resources. The /Public folder contains sample reports. The libraries and templates used by these sample reports are stored in /Resources folder.

If your resource folder in the Encyclopedia volume is different from the default, before publishing a resource, you need to specify the resource folder in the Encyclopedia volume.

How to change the Resource folder on an Encyclopedia volume

- 1 Open Management Console and log in to the Encyclopedia volume.
- 2 Create a folder to designate as a resource folder.
- 3 Choose Volume.
- 4 On Volume, choose Properties.
- 5 On Properties—General, in Resource folder, type or browse to the folder to which you want to publish BIRT resources.

How to publish a resource from the Resource folder to iServer

- 1 In BIRT Designer Professional, choose **File**→**Publish Resource to iServer**.

- 2 On Publish Resources, expand the BIRT Designer Professional Resource folder and select the resources that you want to publish.
- 3 Select the iServer profile, as shown in Figure 2-7.

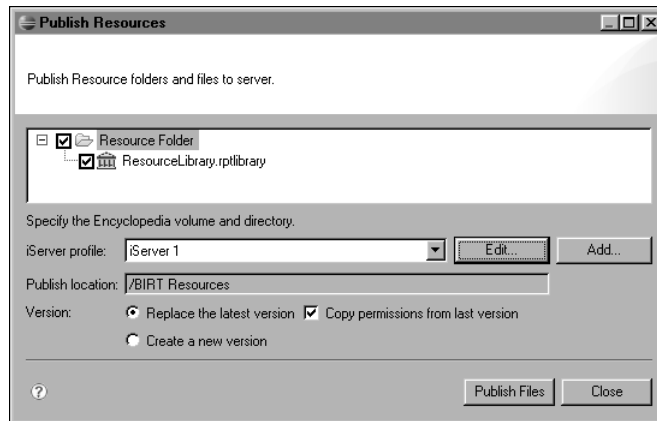


Figure 2-7 Publish Resources dialog

- 4 Choose Publish Files. A window, showing the file upload status, appears.
- 5 Choose OK when the upload finishes.
- 6 On Publish Resources, choose Close.

Deploying Java classes used in BIRT reports

A BIRT report design uses scripts to implement custom functionality. For example, you can use scripts to create a scripted data set or provide custom processing for a report element. When you deploy a BIRT report design to an Encyclopedia volume, you must provide iServer with access to Java classes that the scripts reference. You package these classes as JAR files that can be recognized and processed from an iServer Java Factory process. There are two ways to deploy Java classes:

- Deploy the JAR files to the Encyclopedia volume.
This method supports creating specific implementations for each volume in iServer. This method of deployment requires packaging the Java classes as a JAR file and attaching the JAR file as a resource to the report design file. You treat a JAR file as a resource in the same way as a library or image. Using this method, you publish the JAR file to iServer every time you make a change in the Java classes.

- Deploy the JAR files to the following iServer subdirectory:

```
$ACTUATE_HOME\iServer\resources
```

This method uses the same implementation for all volumes in iServer. Actuate does not recommend deploying JAR files to an iServer /resources folder because iServer must be restarted after deploying the JAR file. Another disadvantage of this deployment technique is that the JAR file deployed in the iServer /resources directory is shared across all volumes, which can cause conflicts if you need to have different implementations for different volumes. When using this method, you do not have to add the JAR file to the report design Resource property.

How to configure BIRT reports and deploy a JAR file to an Encyclopedia volume

- 1 Package the Java classes as a JAR file and copy the JAR file to the BIRT Designer Professional Resource folder.
- 2 Open your report design in BIRT Designer Professional.
- 3 In Outline view, select the root report design slot and select the Resources property group in the Property Editor window.
- 4 On Resources, in JAR files, choose Add and navigate through the Resource folder to select the JAR file as shown on Figure 2-8.

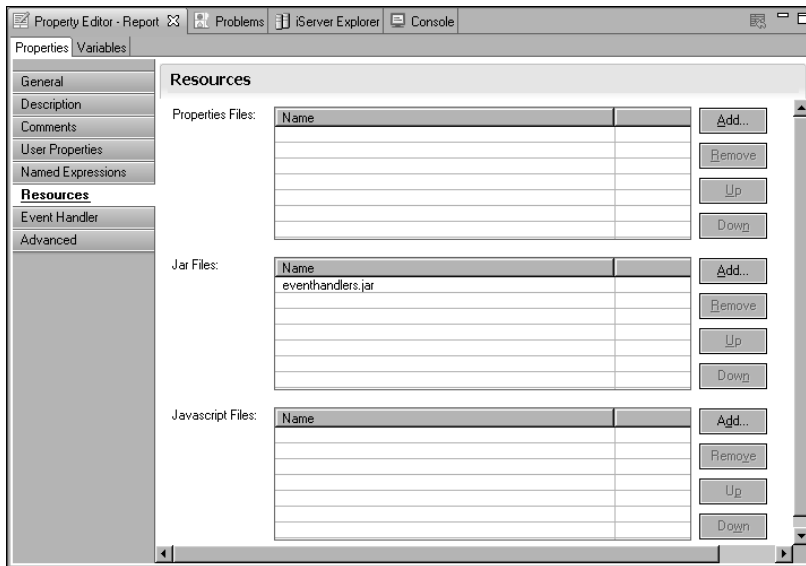


Figure 2-8 Add a JAR file as a resource to a report

When the report executes, the engine searches for the required classes and methods only in this JAR file.

- 5 Choose File→Publish Resource to iServer to publish the JAR file to the iServer. Select the server profile and the JAR file. The JAR file is stored in the Encyclopedia volume Resource folder.

- 6 Choose File→ Publish Report to iServer to publish your BIRT report design to the iServer.
- 7 Run the report from Information Console or Management Console.

How to deploy a JAR file to an iServer /resources folder

- 1 Copy the JAR file to the following iServer subdirectory:

```
$ACTUATE_HOME\iServer\resources
```

\$ACTUATE_HOME is the folder where Actuate products install. By default it is C:\Program Files\Actuate11 on Windows systems.

- 2 Publish the report to iServer.
- 3 Restart iServer.
- 4 Run the report from Information Console or Management Console.

Installing a custom JDBC driver

In order to run a BIRT application in the iServer environment when the BIRT application uses a custom JDBC driver, you must install the JDBC driver in the following location:

```
$ACTUATE_HOME\iServer\Jar\BIRT\platform\plugins  
  \org.eclipse.birt.report.data.oda.jdbc_<VERSION>\drivers
```

Installing custom ODA drivers and custom plug-ins

All custom ODA drivers and custom plug-ins for iServer must be installed in the following folder:

```
$ACTUATE_HOME\iServer\MyClasses\eclipse\plugins
```

By default, iServer loads custom plug-ins from this folder. If you need to deploy a custom plug-in to Actuate BIRT Designer Professional, you must copy the plug-in to:

```
$ACTUATE_HOME\MyClasses\eclipse\plugins
```

If your application uses a different location to store a custom plug-in you must set this location in each product link file. The Actuate products use link files to locate folders where custom plug-ins are deployed. The names of the link files are customPlugins.link and customODA.link. As the file names suggest customODA.link stores the path for custom ODA drivers and customPlugins.link is used for all plug-ins used by BIRT reports and BIRT Engine, such as custom emitters and flash object library plug-ins. Typically the link files are stored in a

\links subfolder of the Eclipse instance of the product. For BIRT Designer Professional, for example, the file is located in:

```
$ACTUATE_HOME\BRDPro\eclipse\links
```

You can change the path in customPlugins.link and deploy the plug-ins to the new location.

The locations of the link files for each product are listed in Table 2-1.

Table 2-1 .link files locations

Product	Paths of .link files
Actuate BIRT Designer Professional	\$ACTUATE_HOME\BRDPro\eclipse\links
Actuate BIRT iServer	\$ACTUATE_HOME\iServer\Jar\BIRT\platform\links

Configuring the BIRT design cache

By default, iServer caches a BIRT design, including access privileges. Caching benefits users who access the design concurrently. Users who request access to the same design share the cached design if they have the required privileges. Performance can improve because iServer does not have to repeatedly load the design. Generally, the fewer number of designs iServer needs to load, the better the response time.

By configuring the cache timeout, the administrator can control how long the design remains in the cache. iServer removes the design from the cache when the time elapses. Increasing the timeout increases the time the design stays in the cache. Decreasing the timeout purges the design sooner.

The administrator can also configure a limit on the number of designs in the cache. When the cache reaches the limit, caching stops. To disable caching, set this limit to zero.

How to configure the BIRT design cache

- 1 In Configuration Console, on Server Configuration Templates, expand Viewing Service, BIRT, and BIRT Content Caches and choose Design Cache.
- 2 In Cache timeout for BIRT designs, accept the default, 1800 seconds or 30 minutes, as shown in Figure 2-9. Alternatively, type another value.
- 3 In Maximum number of BIRT designs to cache, accept the default, 50, or type another value that limits the number of designs in the cache. To disable caching, set to zero.

Choose OK.

4 Restart iServer.

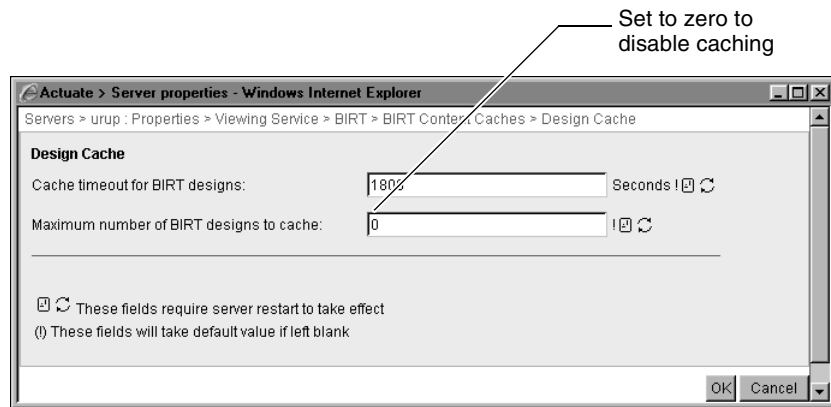


Figure 2-9 Configuring the BIRT design cache

3

Connecting to data sources

This chapter contains the following topics:

- About data source connections
- About data source connection properties

About data source connections

Actuate Customer Support publishes the Supported Products and Obsolescence Policy document that describes the data sources, drivers, operating systems, and other software requirements for connecting iServer to data sources. Actuate Supported Products and Obsolescence Policy, available on the Actuate Support site at the following URL, also contains information about the required patches:

<http://support.actuate.com/documentation/spm>

iServer connects to data sources when generating reports and using the Actuate Caching service (ACS). The report designer specifies data source connection information in the report design, or you specify it in an external connection configuration file. In most cases, iServer and the database run on different computers for load-balancing purposes, but this division is not mandatory. Running iServer on the database host can improve performance.

The iServer installation process installs and configures DataDirect Connect for ODBC drivers and JDBC drivers. You can also use third-party drivers to connect to data sources, but you must license, install, and configure them.

To connect BIRT reports to other JDBC data sources, place the .jar files for the custom database driver in:

```
AC_SERVER_HOME\iServer\Jar\birt\platform\plugins  
  \org.eclipse.birt.report.data.oda.jdbc<version>\drivers
```

About data source connection properties

Every report design specifies the connection properties required to connect to a data source. Typically, many report designs access the same data source. Rather than typing the same connection properties for each design, you can create a connection profile to reuse the same connection properties across multiple designs. The connection profile includes information about the database driver, a user ID, password, port, host, and other properties related to the type of data source.

Typically, you change database connection properties used in the development environment when you publish the reports to iServer. To change the connection properties dynamically when you design or deploy your reports, you can use one of two approaches, a connection configuration file or a connection profile. The following sections describe these two approaches.

Using a connection configuration file

A connection configuration file is an XML file, such as the one shown in Listing 3-1, in UTF-8 or ASCII encoding. The file specifies the data source

connection properties to use when iServer runs a report. Having the data source connection information for a report in an external file makes it convenient to modify. You change the connection information without altering the report design. You specify the location of the file using Configuration Console.

You can use an external connection configuration file to define a data source for the Actuate Caching service and for a data connection definition (.dcd) file, which contains information object connection properties for a data source. You can also use an external connection configuration file for connecting data sources to reports.

You can create an external connection profile to a data source used by a report. Changes to the profile are automatically picked up by the report. The settings in a connection configuration file override any connection configuration properties in the connection profile. The sample connection configuration file in Listing 3-1 externalizes the file path to the connection profile, C:\MyPath.

Listing 3-1 BIRT connection configuration file example

```
<oda-data-source
  extensionID="org.eclipse.birt.report.data.oda.jdbc" name="JDBC
  Data Source - SQL Server" id="783">
  <property name="odaDriverClass">com.actuate.jdbc.sqlserver.
    SQLServerDriver
  </property>
  <property name="odaURL">jdbc:actuate:sqlserver://DBSRV1-W2K
  </property>
</oda-data-source>
<ConnectOptions Type=".eclipse.birt.report.data.oda.jdbc_ JDBC
  Data Source - SQL Server ">
  <Property PropName="OdaConnProfileStorePath">C:\MyPath
  </Property>
</ConnectOptions>
```

In a BIRT report, the configuration key used to specify a data source is the unique ID of the ODA data source extension and data source name defined in the BIRT report design or library. You must concatenate the string as follows:

extensionID + "_" + data source name

For example, the key is org.eclipse.birt.report.data.oda.jdbc_SQL Server.

Changing a configuration file

The Factory process reads the configuration file for the configuration key values when the process starts. After changing a configuration file, you must restart Factory processes for changes to take effect. Only Factory processes that start after changes in the configuration file use the new information. To ensure that report executable files use updated configuration file information, confirm that no reports are active and stop Factory processes that are running before you change

the configuration file. After you change the file, iServer starts a Factory process for the next report request.

Specifying the location of the connection configuration file

There is no default location for the connection configuration file. To use a connection configuration file, you create the file and then specify its name and location using the ConnConfigFile parameter in Configuration Console.

From Server Templates—Advanced, choose Runtime from the list of property settings. Specify the location of the file using the Configuration file for database connections and search path parameter shown in Figure 3-1.

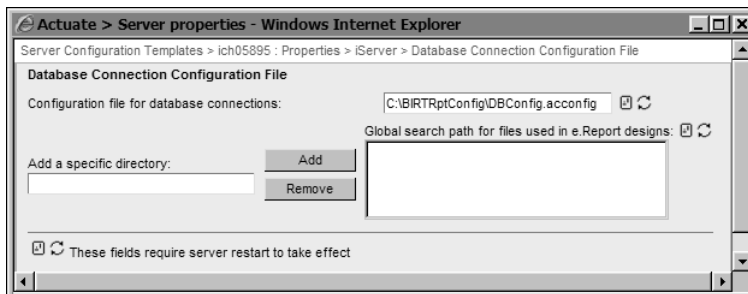


Figure 3-1 Specifying the location of a connection configuration file

On UNIX and Linux, the value of the parameter can be a path and file name only. On Windows, it can be either a path and file name or a URL. For example:

```
\\server1\configs\serverconfig.xml
```

or:

```
http://myserver/configs/testconfig.xml
```

If you do not specify a value for the configuration file parameter, iServer uses the data source connection properties in the report executable file.

Encrypting the connection properties

Actuate BIRT supports the encryption of connection properties in the connection configuration file. The encryption conversion is created in Actuate BIRT Designer Professional, using BIRT's encryption framework. The encryption user interface reads a user-specified configuration file, and writes the encrypted values for a specified property type to a new output file. The configuration file must have the file name extension .acconfig.

The run-time decryption processing runs in Actuate BIRT Designer Professional, iServer, and Actuate Java Component. You must deploy the encrypted version of a configuration file to the iServer or Actuate Java Component environments, and set up the database configuration for iServer.

For more information about the encryption conversion and the BIRT encryption mechanism, see *Using Actuate BIRT Designer Professional*.

Configuring a cluster to use a connection configuration file

If you have a cluster of iServers, each node must have access to the connection configuration file. The path can be a local absolute path on each machine and must be specified for each iServer. If you use a single copy of the file for a cluster, put it in a shared location, then specify the path to that shared location for all iServers in the cluster.

Using a connection profile

A connection profile contains all of the necessary information to allow a BIRT report to connect to a data source. BIRT supports using a connection profile when creating a data source in a report design. When the connection profile changes, the BIRT report picks up those changes. This behavior makes migration from a test to a production environment straightforward.

The connection profile is stored in the .metadata folder in the Report Designer workspace. The default name of the connection profile is ServerProfiles.dat. All connection profile properties can be bound to report parameters or expressions and changed when the report is generated. You can use the Export feature in Data Source Explorer to create a connection profile with a different name and store it at a different location.

When deploying reports that use connection profiles, you must deploy the connection profile to the correct folder in the file system on the iServer machine. For example, if a report uses a connection profile stored in a folder:

```
C:\ActuateBIRT11\ConnProfile\MySQL.dat
```

as shown in Figure 3-2, you must manually create the same folder structure on the iServer machine and copy the MySQL.dat file there.

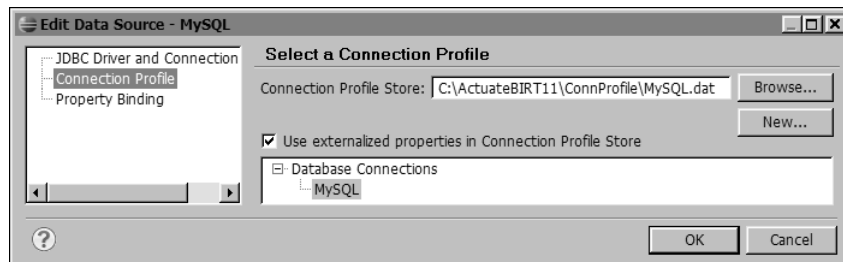


Figure 3-2 Connection profile properties in BIRT Designer Professional

If a BIRT report design uses a connection profile in the deployment environment, the report developer should use a connection profile in the development environment rather than specifying the data source connection properties in the

report design. Then, if iServer cannot find the connection profile or the connection defined in the connection profile fails, iServer does not run the report. If data source connection properties are specified in the report design and iServer cannot find the connection profile or the connection defined in the connection profile fails, however, iServer runs the report using the data source connection properties in the report design. Alternatively, if data source connection properties are specified in the report design, the report developer should remove them and specify a connection profile before deploying the report design to iServer.

For more information about how to create and manage connection profiles, see *BIRT: A Field Guide*.

Externalizing the connection profile properties on the iServer

The iServer database connection configuration file is used to externalize data source connection properties for a BIRT report design. As the connection profile store URL is the ODA data source property, `OdaConnProfileStorePath`, the file path to the Connection Profile can itself be externalized. When the report is deployed to the iServer and executed, the server reads the connection profile from the file path specified in the iServer's database connection configuration file. The file path specified in the report design is ignored.

Understanding externalization precedence

Data source properties in a report design can be externalized to the connection profile and to the iServer connection configuration file. In addition the Connection Profile Store URL itself can be externalized. The following precedence rules explain how iServer determines the final list of data source properties for report execution.

Data source properties in the iServer connection configuration file override the data source in the connection profile that overrides the data source connection properties in the report. The ascending order of precedence for iServer looks like this:

- Data source properties in the report design
- Data source properties in the connection profile
- Data source properties in the iServer connection configuration file

The following sample connection configuration file externalizes the file path to the connection profile and shows the required structure:

```

<Config>
<Runtime>
<ConnectOptions Type="org.eclipse.birt.report.data.oda.jdbc_SQL
  Server Data Source">
  <Property PropName="OdaConnProfileStorePath">
    C:\SqlServer.profile
  </Property>
</ConnectOptions>
</Runtime>
</Config>

```

The connection profile referenced by the BIRT report design is read when the report is executed in iServer. The path to the connection profile in the design has to be visible to iServer.

Referencing the external connection profile

The path to the external connection profile is stored in the BIRT report design. The ODA data source property, Connection Profile Store URL, holds this value. The path can be a relative or an absolute file path, or a URL. This path must be accessible by the iServer when the report is deployed. Actuate does not recommend the use of relative file paths. Typically, the location of the connection profile in both environments, BIRT Designer Professional and iServer, resolves to a different path. Absolute paths have the disadvantage that the absolute path used in the BIRT Designer Professional environment on Windows will not be available when the report is deployed to iServer on UNIX. On Unix, you can use relative paths with the use of soft links, but these links are not available on Windows.

When the file path to the connection profile is different in the design environment compared to the iServer deployment environment, there are some options to avoid having to change the report design file before deployment, as described in the following sections.

When specifying network paths in BIRT reports always use the Universal Naming Convention (UNC) to describe the path, instead of a mapped drive letter. Windows XP and later do not allow processes running as services to access network resources through mapped network drives. For this reason, a report that uses a mapped drive letter to access a resource runs in BIRT Designer Professional, but fails when the report runs on iServer, because the iServer processes cannot resolve the mapping address.

For example, a BIRT report uses a flat file Production.csv as a data source. The flat file is located on a shared network drive on a machine named ProductionServer. The UNC network path to the file is \\ProductionServer\e\$\Data and it is mapped as X:\ on your system. Using the path X:\ to define the data source HOME folder works only in BIRT Designer Professional. Using the UNC path \\ProductionServer\e\$\Data in the data source definition is the correct way to define network paths.

4

Configuring fonts in BIRT iServer

This chapter contains the following topics:

- About configuring fonts
- Understanding font configuration file priorities
- Understanding how BIRT engine locates a font
- Understanding the font configuration file structure

About configuring fonts

iServer supports rendering BIRT reports in different formats such as PDF, Microsoft Word, Postscript, and PowerPoint. The processes that do the conversion use the fonts installed on your system to display the report characters.

BIRT uses a flexible mechanism that supports configuring font usage and substitution. This mechanism uses font configuration files for different purposes that control different parts of the rendering process. The configuration files can configure the fonts used in specific operating systems, for rendering to specific formats, in specific locales only, or combinations of these parameters.

The plug-in folder, `org.eclipse.birt.report.engine.fonts`, contains the font configuration files. Table 4-1 shows the location of this folder in the supported BIRT environments.

Table 4-1 Locations of the font configuration file plug-in folder

Environment	Font configuration file folder location
BIRT Designer Professional	<code>\$Actuate<release>\BRDPro\eclipse\plugins</code>
iServer	<code>\$Actuate<release>\iServer\Jar\BIRT\platform\plugins</code>

Understanding font configuration file priorities

BIRT reports use five different types of font configuration files. The font configuration file naming convention includes information about the rendering format, the system platform, and the system locale, as shown in the following general format:

```
fontsConfig_<Format>_<Platform>_<Locale>.xml
```

The platform name is defined by the Java System property, `os.name`. The current Java Network Launch Protocol (JNLP) specification does not list the values for the `os` attributes. Instead it states that all values are valid as long as they match the values returned by the system property `os.name`. Values that only match the beginning of `os.name` are also valid. If you specify Windows and the `os.name` is Windows 98, for example, the operating system name is accepted as valid.

The following sample Java class code shows how to check the `os.name` property for the value on your machine:

```

class WhatOS
{
    public static void main( String args[] )
    {
        System.out.println( System.getProperty("os.name") );
    }
}

```

BIRT supports the following types of font configuration files, with increasing priority:

- For all rendering formats
 - These files have no format specifier in their names. These configuration files are divided into three sub-types:
 - The default configuration file:
 - `fontsConfig.xml`
 - Configuration files for a specific platform, for example:
 - `fontsConfig_Windows_XP.xml`
 - Configuration files for a specific platform and locale, for example:
 - `fontsConfig_Windows_XP_zh.xml`
 - `fontsConfig_Windows_XP_zh_CN.xml`
- For certain formats only
 - These files include the format specifier in their names. These configuration files are divided into two sub-types:
 - The default configuration file for a format, for example:
 - `fontsConfig_pdf.xml`
 - Configuration files for a format for a specific platform:
 - `fontsConfig_pdf_Windows_XP.xml`

Understanding how BIRT engine locates a font

The PDF layout engine renders the PDF, Postscript, and PowerPoint formats. The engine tries to locate and use the font specified at design-time. The PDF layout engine searches for the font files first in the fonts folder of the plug-in, `org.eclipse.birt.report.engine.fonts`. If the specified font is not in this folder, the BIRT engine searches for the font in the system-defined font folder. You can change the default load order by using the settings in the font configuration file.

When the required font for a character is not available in the search path or is incorrectly installed, the engine uses the fonts defined in the UNICODE block for

that character. If the UNICODE definition also fails, the engine replaces the character with a question mark (?) to denote a missing character. The font used for the ? character is the default font, Times-Roman.

The engine maps the generic family fonts to a PDF embedded Type1 font, as shown in the following list:

- cursive font styles to Times-Roman
- fantasy font styles to Times-Roman
- monospace font styles to Courier
- sans-serif font styles to Helvetica
- serif font styles to Times-Roman

Understanding the font configuration file structure

The font configuration file, `fontsConfig.xml`, consists of three major sections, `<font-aliases>`, `<composite-font>`, and `<font-paths>` sections.

<font-aliases> section

In the `<font-aliases>` section, you can:

- Define a mapping from a generic family to a font family. For example the following code defines a mapping from the generic type "serif" to a Type1 font family Times-Roman:

```
<mapping name="serif" font-family="Times-Roman"/>
```

- Define a mapping from a font family to another font family. This definition is useful if you want to use a font for PDF rendering which differs from the font used at design time. For example, the following code shows how to replace `simsun` with Arial Unicode MS:

```
<mapping name="simsun" font-family="Arial Unicode MS"/>
```

Earlier versions of BIRT Designer Professional use the XML element `<font-mapping>` instead of `<font-aliases>`. In the current release, a `<font-mapping>` element works in the same way as the `<font-aliases>` element. When a font configuration file uses both `<font-mapping>` and `<font-aliases>`, the engine merges the different mappings from the two sections. If the same entries exist in both sections, the settings in `<font-aliases>` override those in `<font-mapping>`.

<composite-font> section

The <composite-font> section is used to define a composite font, which is a font consisting of many physical fonts used for different characters. For example, to define a new font for currency symbols, you change font-family in the following <block> entry to the Times Roman font-family:

```
<composite-font>
...
<block name="Currency Symbols" range-start="20a0"
      range-end="20cf" index="58" font-family="Times Roman" />
...
</composite-font>
```

The composite fonts are defined by <block> entries. Each <block> entry defines a mapping from a UNICODE range to a font family name, which means the font family is applied for the UNICODE characters in that range. You cannot change the block name or range or index as it is defined by the UNICODE standard. The only item you can change in the block element is the font-family name.

You can find information about all the possible blocks at <http://www.unicode.org/charts/index.html>.

In cases when the Times Roman font does not support all the currency symbols, you can define the substitution character by character using the <character> tag, as shown in the following example:

```
<composite-font>
...
  <character value="?" font-family="Angsana New"/>
  <character value="\u0068" font-family="Times Roman"/>
...
</composite-font>
```

Note that characters are represented by the attribute, value, which can be presented two ways, the character itself or its UNICODE code.

You can find information about all the currency symbols from <http://www.unicode.org/charts/symbols.html>.

A composite font named all-fonts is applied as a default font. When a character is not defined in the desired font, the font defined in all-fonts is used.

<font-paths> section

If the section <font-paths> is set in fontsConfig.xml, the engine ignores the system-defined font folder, and loads the font files specified in the section, <font-paths>. You can add a single font path or multiple paths, ranging from one font path to a whole font folder, as shown in the following example:

```
<path path="c:/windows/fonts"/>  
<path path="/usr/X11R6/lib/X11/fonts/TTF/arial.ttf"/>
```

If this section is set, the PDF layout engine only loads the font files in these paths and ignores the system-defined font folder. If you want to use the system font folder as well, you must include it in this section.

On some systems, the PDF layout engine does not recognize the system-defined font folder. If you encounter this issue, add the font path to the <font-paths> section.

Working with BIRT encryption

This chapter contains the following topics:

- About BIRT encryption
- About the BIRT default encryption plug-in
- Deploying encryption plug-ins to iServer

About BIRT encryption

BIRT provides an extension framework to support users registering their own encryption strategy with BIRT. The model implements the JCE (Java™ Cryptography Extension). The Java encryption extension framework provides multiple popular encryption algorithms, so the user can just specify the algorithm and key to have a high security level encryption. The default encryption extension plug-in supports customizing the encryption implementation by copying the BIRT default plug-in, and giving it different key and algorithm settings.

JCE provides a framework and implementations for encryption, key generation and key agreement, and Message Authentication Code (MAC) algorithms. Support for encryption includes symmetric, asymmetric, block, and stream ciphers. The software also supports secure streams and sealed objects.

A conventional encryption scheme has the following five major parts:

- Plaintext, the text to which an algorithm is applied.
- Encryption algorithm, the mathematical operations to conduct substitutions on and transformations to the plaintext. A block cipher is an algorithm that operates on plaintext in groups of bits, called blocks.
- Secret key, the input for the algorithm that dictates the encrypted outcome.
- Ciphertext, the encrypted or scrambled content produced by applying the algorithm to the plaintext using the secret key.
- Decryption algorithm, the encryption algorithm in reverse, using the ciphertext and the secret key to derive the plaintext content.

About the BIRT default encryption plug-in

BIRT's default encryption algorithm is implemented as a plug-in named:

```
com.actuate.birt.model.defaultsecurity_<Release>
```

Table 5-1 shows the location of this plug-in folder in the supported BIRT environments.

Table 5-1 Locations of the default encryption plug-in folder

Environment	Default encryption plug-in folder location
BIRT Designer Professional	\$Actuate<Release>\BRDPro\eclipse\plugins

Table 5-1 Locations of the default encryption plug-in folder

Environment	Default encryption plug-in folder location
iServer	\$Actuate<Release>\iServer\Jar\BIRT\platform\plugins

More information about how to implement encryption in BIRT reports is available in *Using Actuate BIRT Designer Professional*.

The BIRT default encryption plug-in consists of the following main modules:

- acdefaultsecurity.jar
- encryption.properties
- META-INF/MANIFEST.MF
- plugin.xml

About acdefaultsecurity.jar

This JAR file contains the encryption classes. The default encryption plug-in also provides key generator classes that can be used to create different encryption keys.

About encryption.properties

This file specifies the encryption settings. BIRT loads the encryption type, encryption algorithm, and encryption keys from the encryption.properties file to do the encryption. The file contains pre-generated default keys for each of the supported algorithms.

About META-INF/MANIFEST.MF

META-INF/MANIFEST.MF is a text file that is included inside a JAR to specify metadata about the file. Java's default ClassLoader reads the attributes defined in MANIFEST.MF and appends the specified dependencies to its internal classpath. The encryption plug-in id is the value of the Bundle-SymbolicName property in the manifest file for the encryption plug-in. You need to change this property when you deploy multiple instances of the default encryption plug-in, as described later in this chapter.

About plugin.xml

plugin.xml is the plug-in descriptor file. This file describes the plug-in to the Eclipse platform. The platform reads this file and uses the information to populate and update, as necessary, the registry of information that configures the whole platform.

Deploying encryption plug-ins to iServer

If you deploy report designs to iServer, you also need to deploy the encryption plug-in to iServer. iServer loads all encryption plug-ins at start up. During report execution the server reads the encryptionID property from the report design file and uses the corresponding encryption plug-in to decrypt the encrypted property. Every time you create reports using an encryption plug-in, make sure you deploy the plug-in to iServer, otherwise report execution on the server fails.

How to deploy an encryption plug-in instance to iServer

1 Copy:

```
$ACTUATE_HOME\BRDPro\eclipse\plugins  
  \com.actuate.birt.model.defaultsecurity_<Release>_rsa
```

to:

```
$iServer\Actuate11\iServer\Jar\BIRT\platform\plugins
```

- 2** Publish the report design to iServer.
- 3** Restart iServer to load the encryption plug-in.
- 4** Log in to iServer using Information Console and run the report. iServer uses the encryption plug-in to decrypt the password.

Using custom emitters

This chapter contains the following topics:

- About custom emitters
- Deploying custom emitters to iServer
- Rendering in custom formats

About custom emitters

In Actuate BIRT Designer Professional or Interactive Viewer you can choose to render BIRT reports in several different formats as shown in Figure 6-1.

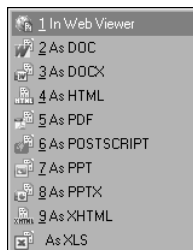


Figure 6-1 Rendering formats

Actuate provides report rendering for the following file formats:

- DOC - Microsoft Word document.
- DOCX - Microsoft Word document, introduced in Windows 7.
- HTML - Hypertext Markup Language document, a standard format used for creating and publishing documents on the World Wide Web.
- PDF - Created by Adobe, a PDF is a portable file format intended to facilitate document exchange.
- POSTSCRIPT - A page description language document for medium- to high-resolution printing devices.
- PPT - Microsoft PowerPoint document.
- PPTX - Microsoft PowerPoint document for Windows 7.
- XHTML - Extensible Hypertext Markup Language document, the next generation of HTML, compliant with XML standards.
- XLS - MS-Excel Document.

If you need to export your document to a format not supported by Actuate, for example CSV or XML, you must develop a custom emitter. Actuate supports using custom emitters to export reports to custom formats. After a system administrator places custom emitters in the designated folder in iServer, users are able to use them as output formats when scheduling BIRT report jobs or exporting BIRT reports. Custom emitters are also supported as e-mail attachment formats.

The iServer uses the custom emitter format type as a file extension for the output file when doing the conversion. When you develop custom emitters always use the same name for a format type and an output file extension type. Management

Console and Actuate Information Console display the options of each emitter for the user to choose when exporting a report.

Integrating and Extending BIRT, published by Addison-Wesley, provides detailed information about how to develop custom emitters in BIRT.

Deploying custom emitters to iServer

The custom emitters in BIRT are implemented as plug-ins and packaged as JAR files. To make them available to the Actuate products that support them, copy the emitters to the following folder:

```
Actuate<release>/MyClasses/eclipse/plugins
```

To deploy custom emitter to iServer copy the plug-ins to:

```
Actuate<release>/iServer/MyClasses/eclipse/plugins
```

The MyClasses folder appears at different levels on different platforms but it is always available in the product's installation folder.

Every time you deploy a custom emitter you need to restart the product. This ensures the emitter JAR is added to the classpath and the product can discover the new rendering format.

The following tools support custom emitters:

- Actuate BIRT Designer
- BIRT Interactive Viewer
- Information Console
- Management Console

Rendering in custom formats

After deploying the custom emitter you can see the new rendering formats displayed along with built-in emitters in the following places:

- Preview report in the Web Viewer in Actuate BIRT Designer
- Output page of schedule job in Management Console and Information Console
- Attachment Notification page of schedule job in Management Console and Information Console
- Export Content in Actuate BIRT Viewer and Actuate BIRT Interactive Viewer

The following examples show the deployment and usage of a custom CSV emitter. The CSV emitter renders a report as a comma separated values file. The JAR file name is org.eclipse.birt.report.engine.emitter.csv.jar. The custom format type is MyCSV.

To test the emitter functionality with Management Console or Information Console, you schedule any BIRT report design or report document from the examples in the Public folder. The examples that follow use the report from the sample Encyclopedia volume for an iServer:

Public/BIRT and BIRT Studio Examples/CustomList.rptdesign

How to deploy a custom emitter

The assumption in this example is that the Actuate products are installed in C:\Program Files\Actuate<Release> folder on Windows.

1 Copy org.eclipse.birt.report.engine.emitter.csv.jar to:

```
C:\Program Files\Actuate<Release>\iServer\MyClasses\eclipse
\plugins
```

2 Restart the product to make it load the new plug-in in its classpath:

- For BIRT Designer Professional, reopen the designer.
- For iServer, restart Actuate iServer <Release> from Start>Settings>Control Panel>Administrative Tools>Services, as shown in Figure 6-2.
- For a separately deployed Information Console, you must also restart Apache Tomcat for Actuate Information Console <Release>.

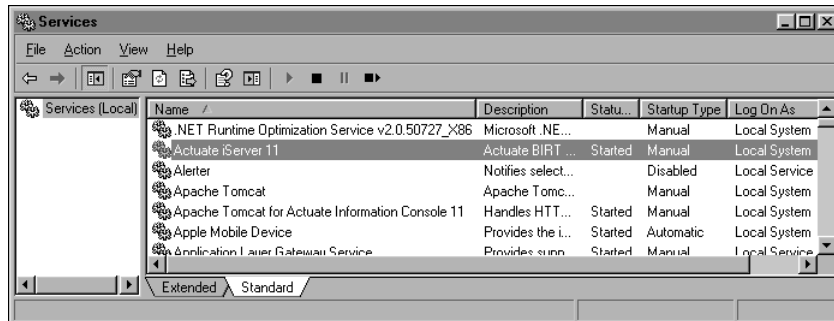


Figure 6-2 Services

The next few procedures show you how to export a BIRT report to the new MyCSV format in several different products.

How to export a BIRT report from BIRT Designer Professional

1 Open a BIRT report in Actuate BIRT Designer Professional.

- 2 Preview the report in the Web Viewer. The new MYCSV format appears in the list of formats as shown in Figure 6-3.

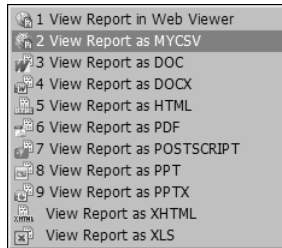


Figure 6-3 List of available formats in Web Viewer

- 3 Select the MYCSV option. A new window opens and displays the report in MYCSV format, as shown in Figure 6-4.

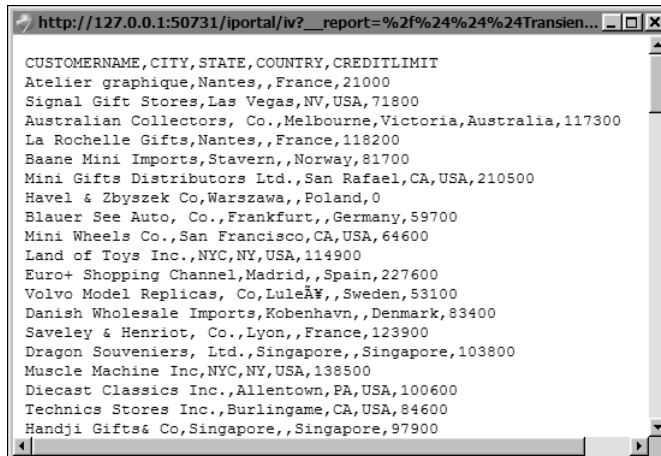


Figure 6-4 Exported content

How to export a BIRT report in Management Console

- 1 Open Management Console.
- 2 Navigate to the Public/BIRT and BIRT Studio Examples folder.
- 3 Click the blue arrow next to CustomerList.rptdesign and select the Schedule option from the menu.
- 4 On the Schedule page select the Output tab. The new MYCSV format appears in the list of the available formats as shown in Figure 6-5.

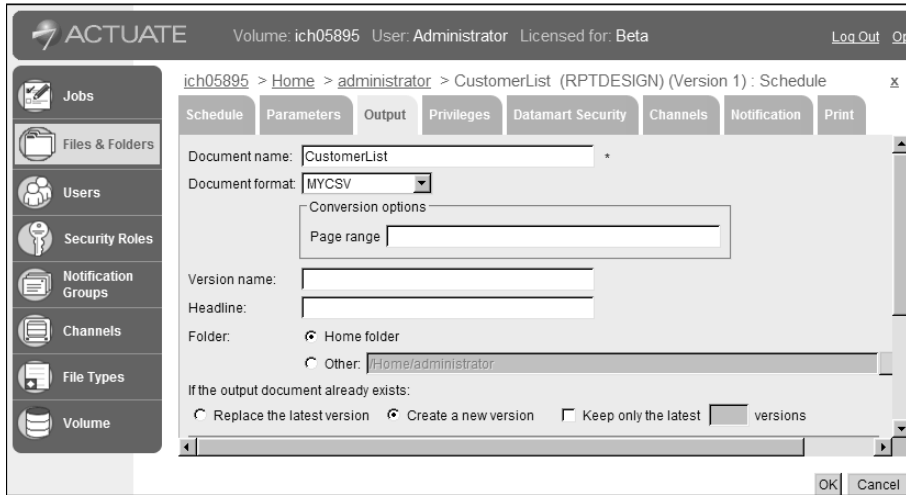


Figure 6-5 Output format in Management Console

- 5 Choose the Notification tab in the Schedule Job page. Select MYCSV format from the Format for the attached report's drop-down list as shown in Figure 6-6.

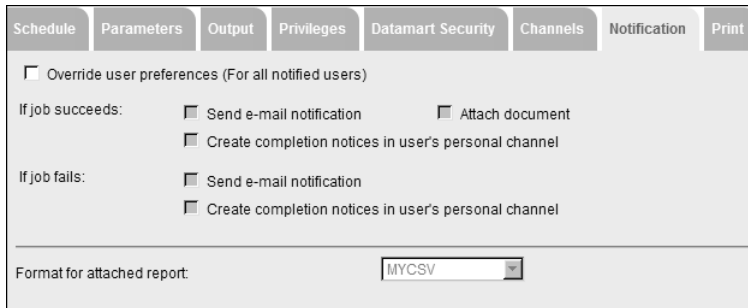


Figure 6-6 Notification tab in the Schedule Job page

- 6 Choose OK. The generated report is saved as CustomerList.MYCSV in the Encyclopedia volume. The report is also attached to the e-mail notification.

How to export a BIRT report from Information Console

Schedule a BIRT report to run by choosing Save As on the schedule page. The new MYCSV format appears in the Document Format list. You can also select to attach the output report to an e-mail notification as shown in Figure 6-7.

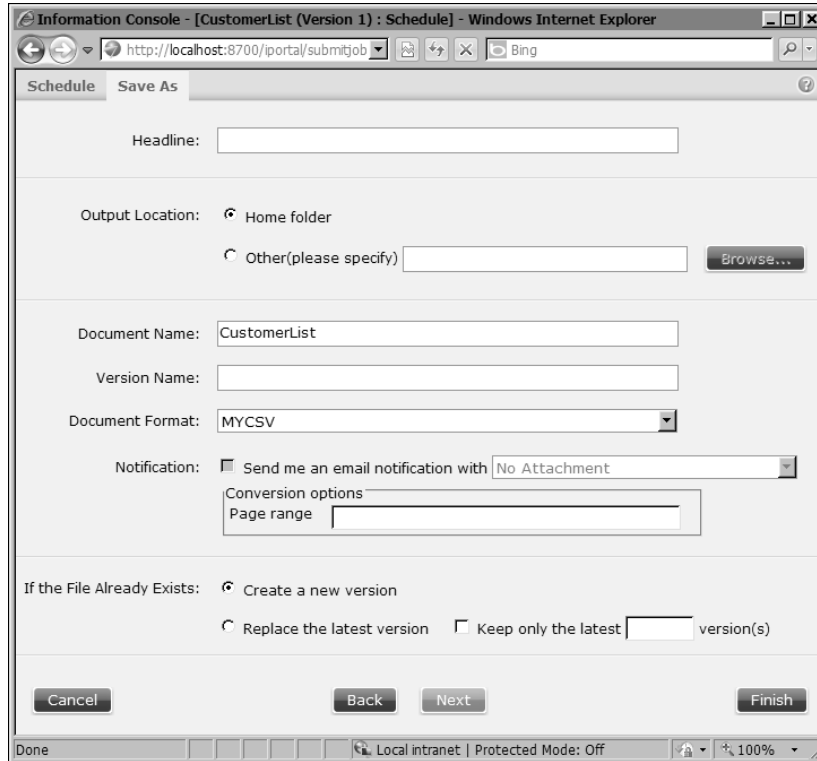


Figure 6-7 Save As tab in the Schedule Jobs page in Information Console

How to export a BIRT report from Actuate BIRT Viewer or Actuate BIRT Interactive Viewer

- 1 Open a BIRT report in Actuate BIRT Viewer or Interactive Viewer.
- 2 Select Export Content from the viewer menu. The MyCSV format appears in Export Formats as shown in Figure 6-8.

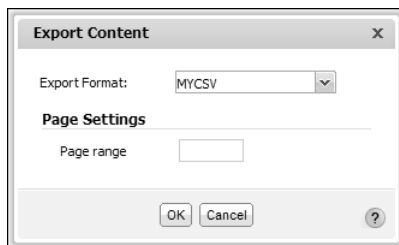


Figure 6-8 Export Content in Actuate BIRT Viewers

- 3 Choose OK. A File Download window appears as shown in Figure 6-9. You can choose to open or save the file. The suggested file name is CustomerList.mycsv.

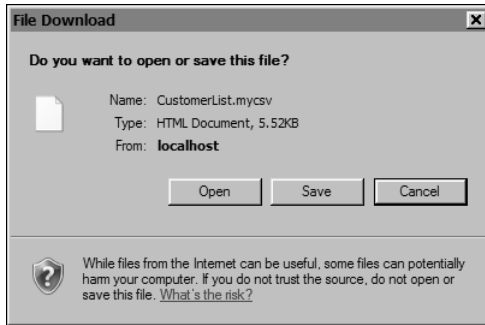


Figure 6-9 File Download

Part Three

**Deploying other reports and
information objects**

7

Deploying e.reports, spreadsheet reports, and information objects

This chapter contains the following topics:

- Deploying e.reports
- Deploying spreadsheet reports
- Deploying information objects

Deploying e.reports

To deploy an e.report, you must first publish the report object executable (.rox) file to the appropriate Encyclopedia volume using e.Report Designer Professional. You may also need to perform the following tasks:

- Define a database connection.
- Make fonts available to e.reports.
- Specify the fonts and parameters for PDF generation.
- Synchronize the localemap.xml files used in the development and deployment environments.

Publishing a report object executable (.rox) file

When you publish a report executable file, e.Report Designer Professional:

- Generates an executable file from the report design.
- Logs in to the Encyclopedia volume that your iServer profile specifies.
- Uploads the executable file to the destination folder your profile specifies.
- Performs other tasks your publication settings require, such as opening your destination folder, displaying the output, or copying properties from an existing version of the file.
- Displays the report in DHTML format in the layout window.

If the report that you publish uses parameters, the layout window first displays a Management Console page that supports providing parameter values, then running the report.

If the report design uses a report object library (.rol) file, you must also publish the library.

How to publish an executable file to an Encyclopedia volume

- 1 With a report design open in the design perspective in e.Report Designer Professional, choose File→Publish to Server. Publish and Preview Options appears.
- 2 In Publish and Preview Options, shown in Figure 7-1:
 - 1 In iServer profile information:
 - Select a profile from the drop-down list.
 - Select a destination folder.
 - 2 In Publish options, select Publish and preview.

- 3 In Version options, select Create a new version.

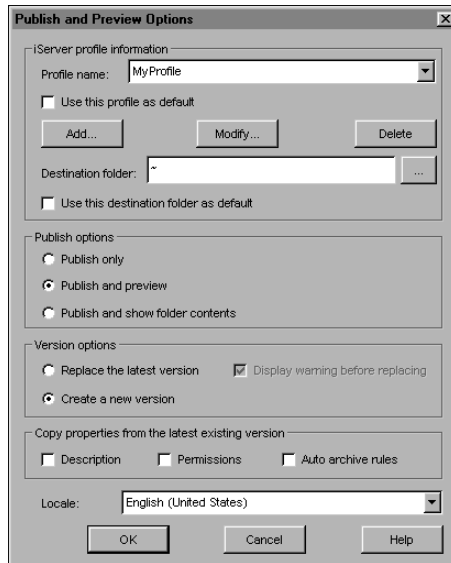


Figure 7-1 Publish and Preview Options

Choose OK. e.Report Designer Professional generates the report object executable (.rox) file.

If the report that you publish uses parameters, the Parameters page of Management Console appears, as shown in Figure 7-2.

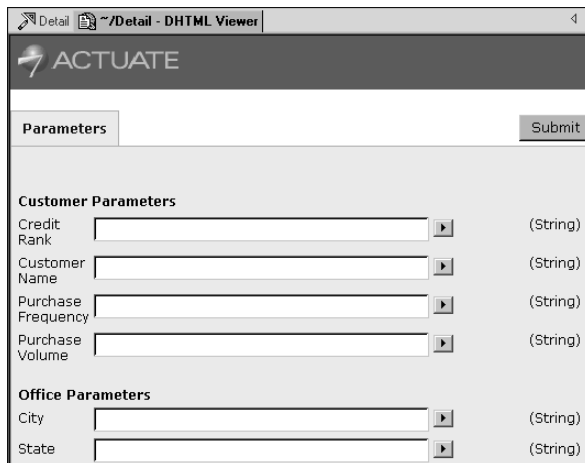


Figure 7-2 Parameters page

Select the parameters to use and choose Submit. The report appears in the DHTML Viewer as shown in Figure 7-3.

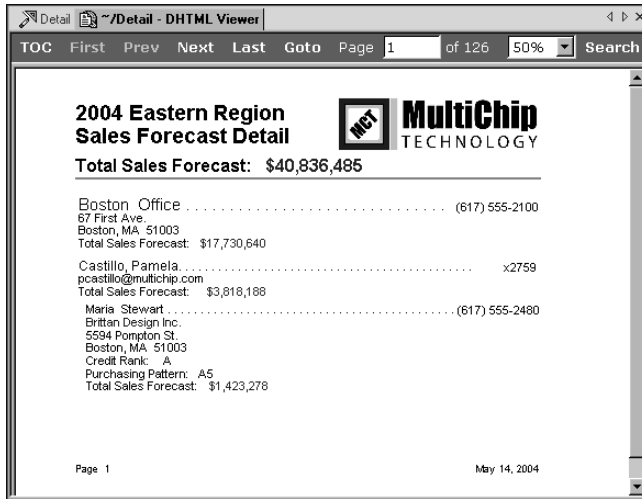


Figure 7-3 Published report in DHTML Viewer

If the report that you publish does not use parameters, the first page of the report appears in the DHTML Viewer.

Defining a database connection

Some native database connection types, such as Oracle, require the installation of a database client before you can connect to the database server. You must also define the database connection on iServer before you can successfully deploy a report. For more information about defining an iServer database connection, see *Configuring BIRT iServer*.

Making fonts available to e.reports

The iServer Factory process uses fonts and font information when generating and printing reports. When you design a report in e.Report Designer Professional, you can specify any combination of font properties for a control. Report users can see text in the fonts that you use only if their computers support your fonts. If the Factory process cannot find your fonts when displaying the report in the PDF Viewer or DHTML Viewer, it substitutes fonts. For example, when a report user views a report object instance (.roi) file, any unavailable fonts map to Windows fonts. Similarly, font substitution occurs if you use a font that is unavailable in the user's print environment.

There are three ways to make fonts available to a report:

- Use the fonts that `master_fonts.rox` defines.
Every report design can use the fonts this file makes available.
- Define fonts in a file named `customized_fonts.rox`.
Create this file. Fonts must be available in your environment before you can define a customized font file.
- Define fonts in the report object executable (`.rox`) file.
A font you use in the report design is embedded in the ROX that generates from the design, except for fonts in charts and images and fonts the user sets dynamically at run time.

Depending on the environment in which the Factory process is running, the Factory process looks for font information in your report object executable (`.rox`) file, a custom font file, a master font file, or the system's operating system. The order in which the Factory process searches for font information depends on the system and the system's configuration.

For more information about the `master_fonts.rox` and `customized_fonts.rox` files, see *Developing Reports using e.Report Designer Professional* and *Configuring BIRT iServer*.

Specifying fonts and parameters for PDF generation

You can specify fonts and parameters for PDF generation using a render profile. Render profiles reside in the `AcRenderConfig.xml` file. You can specify font families, including regular, bold, italic, and bold+italic styles. You can also specify parameters such as:

- Whether or not the PDF document includes a table of contents
- The document's author
- The DPI for drawings

For more information about render profiles, see *Developing Reports using e.Report Designer Professional* and *Configuring BIRT iServer*.

Synchronizing the `localemap.xml` files

e.reports use a file called `localemap.xml` to specify the formats for dates, times, currency, and numbers in supported locales. The `localemap.xml` file used to develop an e.report must be identical to the `localemap.xml` file used to deploy the e.report on iServer. For the locations of these files, see *Working in Multiple Locales using Actuate Basic Technology*.

If you are planning to deploy e.reports in more than one locale or in a locale other than English (US), see *Working in Multiple Locales using Actuate Basic Technology*.

Deploying spreadsheet reports

To deploy a spreadsheet report, you must publish the spreadsheet object executable (.sox) file to the appropriate Encyclopedia volume by choosing File→Publish Report in BIRT Spreadsheet Designer. The procedure for publishing an SOX file is very similar to the procedure for publishing an ROX file. You must also deploy the database driver and install any required fonts.

A spreadsheet report may include callback classes or a VBA template. These items require special handling when you publish the report.

Deploying a database driver

To access a database driver from iServer, place the driver archive (.jar) file in:

```
...\Actuate11\iServer\reportengines\engines\ess\lib
```

Installing fonts

The fonts used in a spreadsheet report must be installed on the report user's computer.

Deploying a callback class

When you publish a report that contains a callback class that does not use additional classes to Actuate BIRT iServer System, you must deploy the callback class in the BIRT Spreadsheet library directory:

```
...\Actuate11\iServer\reportengines\engines\ess\lib
```

If the callback class uses additional classes, you must deploy the callback class and the additional classes in either of two ways. The first method requires you to create a JAR file and the second requires you to create one or more new directories.

For information about debugging callback classes, see *Designing Spreadsheets using BIRT Spreadsheet Designer*.

Deploying classes using a JAR file

To deploy classes using a JAR file, you create a Java JAR file containing the callback class and any classes your callback class uses. All the classes must be in the same JAR file. The additional classes must be part of a package, as explained in *Designing Spreadsheets using BIRT Spreadsheet Designer*.

To deploy the JAR file to Actuate BIRT iServer System, place a copy of the JAR file in the BIRT Spreadsheet library directory:

```
...\Actuate11\iServer\reportengines\engines\ess\lib
```

Deploying classes by creating new directories

If you do not create a JAR file with which to deploy the additional classes that your callback class uses, you must create a directory under the BIRT Spreadsheet library directory on Actuate BIRT iServer System.

To locate your package, you must complete the following tasks:

- Create a directory under the BIRT Spreadsheet library directory on the BIRT iServer System machine. The default location is:

```
...\Actuate11\iServer\reportengines\engines\ess\lib
```

- Give the directory the same name as the package.
- Place copies of all the class files for that package in the new directory.

You can have more than one package, and you can access classes in more than one package, but each package must have its own directory.

Publishing a report with a VBA template

You can include VBA functionality in a spreadsheet report you publish to BIRT iServer System. To use VBA functionality, you must develop the VBA code in an Excel file, then use that file as a template for the spreadsheet report. You can use VBA to add custom features such as macros, custom calculations, auto shapes, or cell comments. You can also use VBA to add interactive features. For example, you can use VBA to enable a user to write data back to a database.

When you publish a spreadsheet report to an Encyclopedia volume, ensure that the template file is in the same directory as the SOD file and that the template name matches the name of the SOD file. BIRT Spreadsheet Designer includes the template file when it creates and uploads the spreadsheet report executable file. When users open the published and distributed report, the VBA functionality appears in the report.

Deploying information objects

To deploy an information object, you must publish the information object project to the appropriate Encyclopedia volume using the IO design perspective in BIRT Designer Professional.

When you create an information object project, you must specify a development location that includes the Encyclopedia volume and the folder in which the project resides. When you are ready to deploy the project, you should publish the project to a different location. The publish volume can be the same as the development volume, or you can publish to a different volume. If you publish to the same volume, the publish folder should be different from the development

folder. If you do not specify a project's default publish location when you create the project, you can do so in the project's Properties dialog.

You can publish a project or individual resources to an Encyclopedia volume. Data and cache connection definitions (DCDs), external procedures (EPRs), maps (SMAs), information objects (IOBs), and cache objects (ICDs) are not versioned in an Encyclopedia volume. When you publish a DCD, EPR, SMA, IOB, or ICD, you replace the existing file.

How to specify a default publish location for a project

- 1 In Navigator, select the project.
- 2 Choose File→Properties.
- 3 In Properties, select Actuate iServer.
- 4 In Publish Location, choose Add.
- 5 Complete the New iServer Profile dialog. Choose Finish.
- 6 Browse for the appropriate folder or accept the default folder, shown in Directory in Figure 7-4. Choose OK.

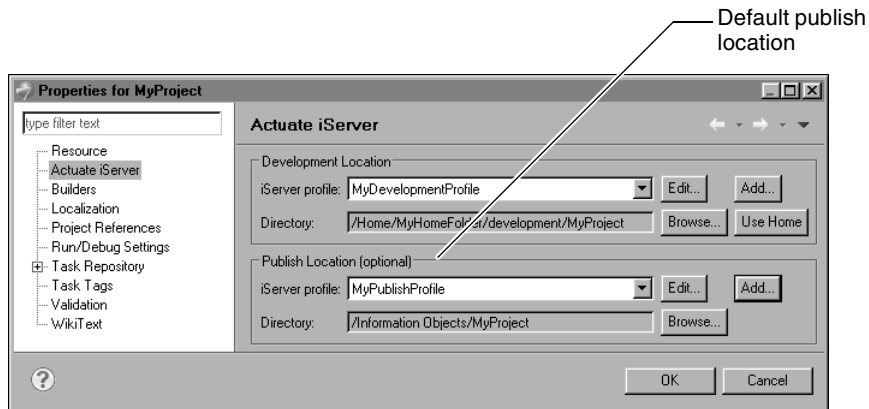


Figure 7-4 Default publish location for a project

How to publish a project

- 1 In Navigator, select the project.
- 2 Choose File→Publish Information Objects.
- 3 In Publish Information Objects:
 - 1 Select the project, a folder, or individual resources.
 - 2 In iServer profile, select a profile from the list. If you specified a default publish location for the project, the iServer profile and folder appear.

- 3 In Publish location, browse for the appropriate folder or accept the default folder, as shown in Figure 7-5.
- 4 If you want Information Object Designer to remember this location the next time you publish a project, select Remember this location.

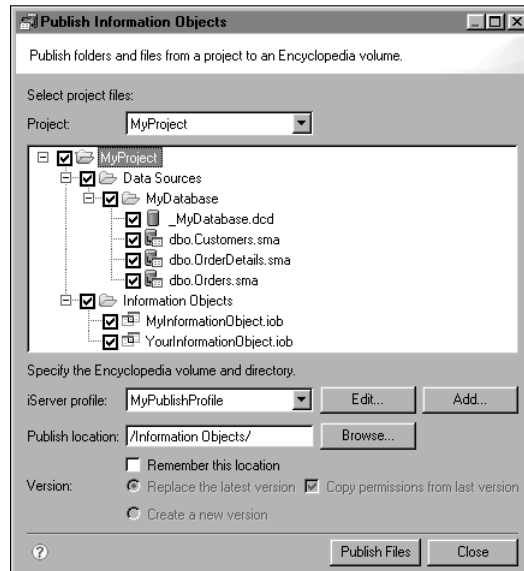


Figure 7-5 Selecting files to publish

- 5 Choose Publish Files. Publishing appears, as shown in Figure 7-6. Messages indicate whether the selected items are published. If an item is not published, a message gives the reason.

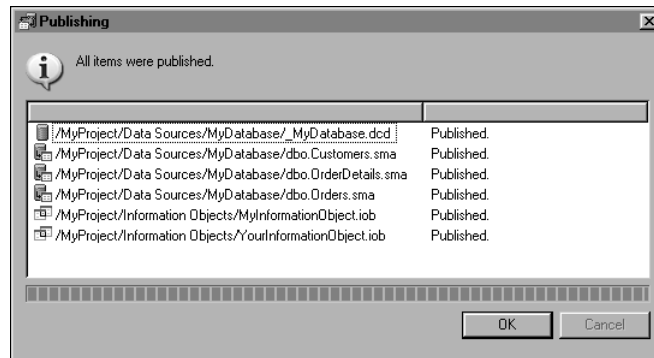


Figure 7-6 Publishing messages

- 4 When publishing is complete, choose OK in Publishing.
- 5 In Publish Information Objects, choose Close.

Index

Symbols

? (question mark) characters 32

A

absolute paths 27

accessing

connection profiles 25

custom emitters 41

custom ODA plug-ins 17

data 5, 22

database drivers 17, 54

font files 30

iServer Explorer 11

JAR files 16

Java classes 15

ODA data sources 5, 17

reports 10

resources 27

sample reports 14

Actuate Supported Products and
Obsolescence Policy 22

adding

connection profiles 22, 25

custom drivers 5

iServer profiles 10, 11, 13

administrators 4

algorithms (encryption) 36

applications 17, 42

attachments 40

authentication 36

B

BIRT applications 17

BIRT Content Caches properties

design cache 18

BIRT design cache 18

BIRT Designer Professional 10, 17, 40, 42, 55

BIRT Engine 17

BIRT reports 10

See also reports

BIRT Spreadsheet Designer 54, 55

BIRT Studio 4

BIRT Viewer 45

block element 33

Bundle-SymbolicName property 37

C

cache

setting timeout values for 18

cache connection definitions 56

cache objects 56

caching

BIRT designs 18

privileges 18

callback classes 54–55

changing

configuration files 23

connection profiles 23, 25

connection properties 22, 23, 25

resource folders 14

character sets 31, 33

character tag 33

classes 15, 16, 37, 54

classpaths 41

clusters 25

comma separated values file 42

composite fonts 33

composite-font element 33

Configuration Console 23, 24

configuration files

connecting to data sources and 5, 22, 26

editing 23

font usage and substitution in 30, 31, 32

rendering PDF reports and 53

setting default location for 24

updating 23

configuration keys 23

ConnConfigFile parameter 24

connection configuration files. *See*

configuration files; connections

connection definition files 23, 56

connection information 22, 23

See also connection properties

connection parameters 24

Connection Profile Store URL property 26, 27

- connection profiles
 - accessing data sources and 22, 25, 26
 - changing 25
 - creating 10, 11, 13, 25
 - deploying 25
 - externalizing 23, 26, 27
- connection properties 24
 - accessing data sources and 22
 - changing 22, 23, 25
 - externalizing 26–27
 - overriding 23
 - saving 10
- connections
 - accessing nonnative data sources and 5
 - changing 25
 - configuring 5, 22
 - designing reports and 5
 - externalizing 26–27
 - installing database clients for 52
 - publishing reports and 10
 - retrieving data and 22, 25
 - running reports and 5
 - setting default locations for 24
- copying
 - connection profiles 25
 - custom emitters 41
- creating
 - connection profiles 22, 25
 - custom drivers 5
 - information object projects 55
 - iServer profiles 10, 11, 13
 - reports 38
- CSV emitter 42
- currency symbols 33
- custom emitters 40, 41, 42
- customizing
 - data drivers 5
 - report designs 15

D

- data
 - accessing 5, 22
 - formatting 53
 - retrieving 22, 25
 - updating 10
- data connection definition files 23, 56

- data connections. *See* connections
- data sources 4, 5, 22
- database connection types 52
- database connections. *See* connections
- database drivers. *See* drivers
- databases 22, 52, 55
 - See also* data sources
- .dcd files. *See* data connection definition files
- decryption 36, 38
- default connections 24
- default encryption 36
- default font 32, 33
- default font configuration file 31
- deploying
 - callback classes 54–55
 - connection profiles 25
 - custom emitters 41, 42
 - custom ODA plug-ins 18
 - database drivers 54
 - encryption plug-in 37, 38
 - information objects 4, 55
 - Java classes 15, 16
 - report design files 15, 38
 - reports 4, 10, 25, 50
 - spreadsheet reports 54
- design cache 18
- Design Cache property 18
- design files. *See* designs
- designs 18
 - connecting to data sources and 22, 23, 25, 26
 - customizing 15
 - deploying 15, 38
 - publishing 4, 12
- DHTML formats 50
- DHTML Viewer 52
- directory paths
 - configuration files 24, 25
 - connection profiles 26, 27
 - font files 33
 - ODA drivers 17
- displaying reports 30, 41, 52
- DOC formats 40
- documentation v
- drivers
 - accessing 54
 - connecting to data sources and 5, 22

- installing custom JDBC 17
- installing ODA 5, 17
- drives, mapping 27

E

- e.Report Designer Professional 50
- e.reports 50, 53
 - See also* reports
- e.Spreadsheet reports. *See* spreadsheet reports
- Eclipse platform 37
- editing. *See* changing
- e-mail 40
- emitters 40, 41, 42
- encrypting 24
- encryption 24, 36, 37
- encryption classes 37
- encryption keys 36, 37
- encryption plug-in 36, 37, 38
- encryption plug-in descriptor file 37
- encryption settings 37
- encryptionID property 38
- Encyclopedia volumes
 - connecting to 10
 - deploying to 4, 15, 38, 50
 - publishing to 4, 10, 12, 14, 50–52, 55
- .epr files. *See* external procedures
- Excel spreadsheets 40, 55
- executable files
 - changing configurations and 23
 - connecting to data sources and 24
 - embedding fonts in 53
 - publishing 4, 50, 54
 - running spreadsheet reports and 55
- executing
 - applications 17
 - information objects 4, 5
 - reports 10, 27, 38, 44
- exporting reports 40, 42, 43, 44, 45
- external configuration files 23
- external connection profiles 23, 26, 27
- external procedures 56
- externalizing connections 26–27

F

- Factory service 23, 52, 53

- file names 24, 30, 40
- file paths. *See* directory paths
- files

- See also* specific type
- connecting to data sources and 23
- deploying reports and 50
- embedding fonts in 53
- generating reports and 40
- loading custom plug-ins and 17, 18
- publishing information object projects and 56
- publishing reports and 50
- sharing 18

folders

- accessing custom emitters and 41
- accessing custom plug-ins and 17
- accessing JDBC drivers and 17
- changing resource 14
- configuring fonts and 30, 31, 33
- deploying callback classes to 55
- deploying JAR files to 16, 17
- sharing resources and 14
- storing connection profiles and 25

- font configuration files 30, 32

- font files 31, 33

- font information 53

- font-aliases element 32

- font-mapping element 32

- font-paths element 33

fonts

- displaying reports and 30, 31
- embedding 53
- generating reports and 52, 53
- installing 54
- mapping 32, 33
- printing reports and 52
- substituting 31, 33, 52

- formats 53

- See also* output formats

G

- generating reports 22, 40, 41, 52, 53

H

- HTML formats 40

I

- .icd files. *See* cache objects
- Information Console 10, 41, 44
- Information Object Designer 57
- information object files 56
- information objects
 - configuring connections for 23
 - deploying 4, 55
 - publishing projects for 55, 56
 - running 4, 5
- installation
 - custom ODA plug-ins 17
 - fonts 54
 - JDBC drivers 17
 - ODA drivers 5, 17
- interactive features 55
- Interactive Viewer 40, 45
- .job files. *See* information object files

iServer

- accessing resources and 27
 - changing connection properties for 22
 - connecting to data sources and 22, 24, 25, 26, 27
 - creating profiles for 10, 11, 13
 - deploying to 4, 15, 38, 41, 50
 - displaying reports and 10, 30, 41
 - exporting reports and 40
 - installing custom drivers for 17
 - loading custom ODA plug-ins for 17
 - publishing to 4, 10, 12, 14, 50
- iServer Explorer 10, 11
- iServer Explorer command 11
- iServer Express
- See* iServer

J

JAR files

- accessing 16
- callback classes and 54
- custom emitters and 41, 42
- deploying 16, 17
- Java classes and 15
- JDBC data sources and 22
- Java classes 15, 16, 54
- Java Cryptography Extension 36
- JCE security extension 36

JDBC data sources 22

JDBC drivers 17, 22

L

- libraries 14, 50
- link files 17, 18
- load balancing 22
- loading
 - custom emitters 42
 - custom ODA plug-ins 17
 - encryption plug-in 37, 38
 - font files 31, 33
- locales 30, 53

M

- MAC algorithms 36
- Management Console 14, 40, 43
- map files (information objects) 56
- mapped network drives 27
- mapping fonts 32, 33
- Message Authentication Code algorithms 36
- metadata 37
- missing characters 31
- MyCSV format 42

N

- name property 30
- naming
 - font files 30
 - iServer profiles 11
 - output files 40
- networked environments 27
- New iServer Profile wizard 11
- nonnative data sources. *See* ODA data sources

O

- ODA data source property 26
- ODA data sources 5, 23
- ODA drivers 5, 17
- OdaConnProfileStorePath parameter 26
- ODBC drivers 22
- online documentation v
- open data access technology. *See* ODA data sources

- opening
 - report designs 18
- operating systems 22, 30
- optimizing
 - load operations 18
- os attributes 30
- output files 40
- output formats 30, 31, 40

P

- packages 15, 41, 55
- parameters
 - connecting to data sources and 24, 26
 - generating reports and 53
 - publishing reports and 50, 51
- patches 22
- paths
 - configuration files 24, 25
 - connection profiles 26, 27
 - font files 33
 - ODA drivers 17
- PDF formats 31, 32, 34, 40, 53
- PDF layout engine 31, 34
- PDF Viewer 52
- performance 22
 - sharing report files and 18
- platform-specific fonts 31
- plug-ins
 - custom emitters 41, 42
 - custom ODA drivers 17
 - encryption and 36, 37, 38
- PostScript formats 31, 40
- PowerPoint formats 31, 40
- printing 52
- privileges 5
 - caching 18
- production environments 5
- profiles
 - accessing data sources and 22, 25, 26
 - connecting to iServer and 10, 11, 13
 - creating 25
 - deploying 25
 - externalizing 23, 26, 27
 - generating reports and 53
- project folders 14
- projects (information objects) 4, 55, 56

- properties
 - accessing data sources and 22, 25
 - deploying encryption plug-in and 37, 38
 - externalizing 26–27
 - naming font files and 30
 - overriding 23
 - saving connection 10
- Public folder 14
- Publish and Preview Options dialog 50
- Publish Information Objects command 56
- Publish Information Objects dialog 56
- Publish location property 13
- publish locations 13, 56, 57
- Publish Report command 54
- Publish Report Designs dialog 12
- Publish Report to iServer command 12
- Publish Resource to iServer command 14, 16
- Publish Resources dialog 15
- Publish to Server command 50
- publishing
 - information object projects 4, 55, 56
 - report design files 4, 12
 - report executables 4, 50, 54
 - reports 10, 12, 50–52
 - resources 14, 56
 - spreadsheet reports 55
- Publishing dialog 13, 57

Q

- question mark (?) characters 32

R

- relational databases. *See* databases
- relative paths 27
- render profiles 53
- rendering formats. *See* output formats
- rendering reports. *See* generating reports
- report design cache 18
- report design files. *See* report designs
- report designs 18
 - connecting to data sources and 22, 23, 25, 26
 - customizing 15
 - deploying 15, 38
 - publishing 4, 12
- report emitters 40, 41, 42

- report executables. *See* executable files
- report files
 - sharing 18
- report files. *See* specific type
- report libraries 14
- report object executable files 50, 53
 - See also* executable files
- report object instance files 52
- report object library files 50
- report server. *See* iServer
- report templates 10, 14, 55
- reporting services 27
- reports
 - accessing nonnative data sources and 5, 17
 - accessing sample 14
 - connecting data sources to 23, 25, 26
 - creating 38
 - deploying 4, 10, 25, 50
 - displaying 30, 41, 52
 - exporting 40, 42, 43, 44, 45
 - generating output for 40
 - printing 52
 - publishing 10, 12, 50–52
 - running 10, 27, 38, 44
- Resource command 14
- Resource folder 14
- resource folders 14, 17
- resources 10, 14, 27, 56
- restarting applications 42
- .roi files. *See* report object instance files
- .rol files. *See* report object library files
- .rox files. *See* report object executable files
- running
 - applications 17
 - information objects 4, 5
 - reports 10, 27, 38, 44
 - third-party drivers 22

S

- sample reports 14
- saving connection properties 10
- scheduling reports 44
- scripts 15
- security 36
- sending e-mail attachments 40

- servers 52
 - See also* iServer
- services 27
- shared files 18
- shared resources 14
- .sma files. *See* map files
- software requirements 22
- .sox files. *See* spreadsheet object executable files
- spreadsheet object executable files 54
- spreadsheet reports 10, 54, 55
- Supported Products and Obsolescence Policy 22
- system administrators. *See* administrators
- system-defined fonts 34, 53

T

- templates 10, 14, 55
- testing custom emitters 42
- text 52
- text files 42
- third-party drivers 22
- timeout intervals
 - design cache 18

U

- updating
 - configuration files 23
 - connection profiles 25
 - data 10
- URLs
 - connection profiles 26, 27
 - specifying default connections and 24

V

- value attribute 33
- VBA code 55
- viewing reports 30, 41, 52
- volume administrators 4

X

- XHTML formats 40
- XLS formats 40