



**JASPERREPORTS SERVER
COMMUNITY PROJECT
INSTALLATION GUIDE**

RELEASE 4.7

<http://www.jasperforge.org>

Copyright © 2012 JasperSoft Corporation. All rights reserved. Printed in the U.S.A. JasperSoft, the JasperSoft logo, JasperSoft iReport Designer, JasperReports Library, JasperReports Server, JasperSoft OLAP, and JasperSoft ETL are trademarks and/or registered trademarks of JasperSoft Corporation in the United States and in jurisdictions throughout the world. All other company and product names are or may be trade names or trademarks of their respective owners.

This is version 0812-JSO47-28 of the *JasperReports Server Community Project Installation Guide*.

TABLE OF CONTENTS

Chapter 1 Introduction	9
1.1 Conventions	10
1.2 Java Version Supported	10
1.3 JasperReports Server Distributions	10
1.3.1 Installer Support	10
1.3.2 WAR File Binary Distribution Support	12
1.4 Release Notes	13
1.5 System Requirements	13
1.6 Support for Internationalization	13
Chapter 2 Installing JasperReports Server	15
2.1 Pre-Installation Steps	15
2.2 Starting the Installer	15
2.3 Accepting the License Agreement	16
2.4 Choosing an Installation Directory	16
2.5 Selecting a Tomcat Configuration	16
2.6 Selecting a PostgreSQL Configuration	17
2.6.1 Choosing the Bundled PostgreSQL	17
2.6.2 Choosing an Existing PostgreSQL on a Local Host	17
2.6.3 Using an Existing PostgreSQL on a Remote Host	18
2.6.4 Enabling Connections to a Remote Host	18
2.7 Installing Sample Data	19
2.8 Installing iReport Designer	19
2.9 Completing the Installation	19
2.10 Post-Installation Steps	20
2.10.1 Updates Made by the Installer During Installation	20
2.10.2 Installer Output Log File Location	20
2.10.3 Checking your Java JVM Options	21

Chapter 3	Starting and Stopping JasperReports Server	23
3.1	Start/Stop Menu — Windows	23
3.1.1	Start/Stop Menus — Bundled Tomcat and PostgreSQL	23
3.1.2	Additional Information about the Bundled Tomcat and PostgreSQL	23
3.1.3	Start/Stop Scripts — No Bundled Applications	24
3.2	Start/Stop Scripts — Linux	24
3.2.1	Manual Start/Stop	24
3.2.2	Auto Start/Stop with Bundled Tomcat and PostgreSQL	24
3.3	Start/Stop Apps — Mac OSX	25
3.3.1	Start/Stop Apps — Mac Dock	25
3.3.2	Start/Stop JasperReports Server — Mac Terminal Shell	25
3.4	Logging into JasperReports Server	25
3.5	Starting the Included Jaspersoft iReport Designer	26
3.6	JasperReports Server Log Files	27
Chapter 4	Uninstalling JasperReports Server	29
4.1	Windows	29
4.2	Linux	29
4.3	Mac OSX	30
4.4	Uninstall Survey	30
Chapter 5	Installing the WAR File Distribution	31
5.1	Applications Supported by the WAR File Distribution	31
5.1.1	Database and Application Server Support	31
5.1.2	Operating System Support for Bash Shell	32
5.2	Installing the WAR File Using js-install Scripts	32
5.3	Starting JasperReports Server	34
5.4	Logging into JasperReports Server	34
5.4.1	JasperReports Server Heartbeat	34
5.5	Installing the WAR File Manually	35
5.6	Troubleshooting Your JasperReports Server Configuration	36
5.6.1	JasperReports Server Startup Problems	36
5.6.2	Error Running a Report	36
5.6.3	Error Running js-install Scripts (js-install-ce.bat/sh)	36
5.6.4	Problem Connecting to a Cloud Database Instance	37
5.7	Running the Import and Export Utilities	38
5.7.1	Running Export from Buildomatic	38
5.7.2	Running Import from Buildomatic	38
5.7.3	Running the Import-Export Shell Scripts	39
Chapter 6	Manual Database Creation and Other Information	41
6.1	Manually Creating the JasperReports Server Database	41
6.1.1	PostgreSQL 8.x and 9.x	42
6.1.2	MySQL	42
6.2	Setting JVM Options for Application Servers	43

6.2.1	Tomcat and JBoss JVM Options	43
6.2.2	Changing JVM Options for Bundled Tomcat as a Windows Service	44
6.2.3	Changing JVM Options for Existing Tomcat as a Windows Service	45
6.2.4	Changing JVM Options for Bundled Tomcat on Linux	45
6.2.5	Changing GlassFish JVM Options	46
6.3	Locating and Changing Buildomatic Configuration Files	47
6.3.1	Regenerating Buildomatic Settings	47
6.3.2	Locating Buildomatic-Generated Property Files	47
6.3.3	Changing the JDBC Driver Deployed by Buildomatic	48
6.3.4	Buildomatic Location for JDBC Drivers	49
6.3.5	Buildomatic Location for JasperReports Server WAR File	49
6.3.6	Buildomatic Location for SQL Scripts	49
6.3.7	Buildomatic Location for Database Creation Scripts	49
6.3.8	Buildomatic Location for Sample Data Catalog ZIP Files	50
6.3.9	Hibernate Properties Settings	50
6.3.10	Database Connection Configuration Files	50
6.4	Configuring Report Scheduling	51
6.4.1	Mail Server Configuration Settings	52
6.4.2	Database Settings for the Quartz Driver Delegate Class	52
6.4.3	Settings for the Report Scheduler Web URI	53
6.4.4	Settings for the Quartz Table Prefix	53
6.4.5	Settings for Import-Export	53
6.5	Updating XML/A Connection Definitions	54
Chapter 7	Planning Your 4.7.0 Upgrade	55
7.1	Upgrading Themes in 4.7.0	55
7.1.1	Banner and Toolbar Modifications	56
7.1.2	Changes to IE Overrides	57
7.1.3	Images in JasperReports Server 4.7.0	57
7.2	Data Snapshots	63
7.2.1	Using Data Snapshots	63
Chapter 8	Upgrading from 3.7 - 4.5 to 4.7.0	65
8.1	Standard Upgrade Steps	65
8.2	Backing Up Your Previous JasperServer Instance	66
8.3	Exporting Old Repository Data	66
8.3.1	Using Buildomatic Scripts to Export Data	66
8.3.2	Using the js-export Script to Export Data	67
8.4	Preparing the JasperReports Server 4.7.0 WAR File Distribution	67
8.5	Configuring Buildomatic for Your Database and Application Server	67
8.5.1	Example Buildomatic Configuration	68
8.6	Upgrading to JasperReports Server 4.7.0	68
8.6.1	js-upgrade Test Mode	69
8.6.2	Output Log Location	69
8.6.3	Errors	69

8.7	Starting and Logging into JasperReports Server 4.7.0	69
8.7.1	Clearing Your Browser Cache	70
8.7.2	Logging into JasperReports Server	70
8.8	Additional Tasks to Complete the Upgrade	70
8.8.1	Handling JasperReports Server Customizations	70
8.8.2	Clearing the Application Server Work Folder	70
8.8.3	Clearing the Application Server Temp Folder	70
8.8.4	Clearing the Repository Cache Database Table	71
8.8.5	Updating the XML/A Connections (Optional)	71
8.8.6	Upgrading the Liferay Portal	71
8.9	Manual Upgrade Steps	71
Chapter 9	Upgrading from 4.5.0 to 4.7.0	73
9.1	Upgrade Steps Overview	73
9.2	Backing Up Your Old JasperReports Server Instance	73
9.2.1	Backing Up Your JasperReports Server WAR File	74
9.2.2	Backing Up Your Jasperserver Database	74
9.3	Preparing the JasperReports Server 4.7 WAR File Distribution	74
9.4	Configuring Buildomatic for Your Database and Application Server	74
9.4.1	Example Buildomatic Configuration	75
9.5	Upgrading to JasperReports Server 4.7.0	75
9.5.1	js-upgrade Test	76
9.5.2	Output Log Location	76
9.5.3	Errors	76
9.6	Starting and Logging into JasperReports Server 4.7.0	76
9.6.1	Clearing Your Browser Cache	76
9.6.2	Logging into JasperReports Server	77
9.7	Additional Tasks to Complete the Upgrade	77
9.7.1	Clearing the Application Server Temp Folder	77
9.7.2	Clearing the Repository Cache Database Table	77
9.8	Manual Upgrade Steps	77
Chapter 10	Upgrading JasperServer 3.5.0	79
Chapter 11	Changing Password Encryption in JasperReports Server	81
11.1	Backing Up Your JasperReports Server Database	81
11.2	Stopping Your Application Server	81
11.3	Running the Repository Export Utility	82
11.4	Specifying Encryption Settings in the JasperReports Server WAR	82
11.4.1	Specifying Encryption Settings - Reference Table	83
11.5	Specifying Encryption Settings for the Import Utility	83
11.6	Recreating the JasperReports Server Database	84
11.6.1	Dropping and Recreating the Database in PostgreSQL	84
11.6.2	Dropping and Recreating the Database in MySQL	84
11.7	Importing Your Repository Data	84

11.8	Starting the Application Server	84
11.9	Logging into JasperReports Server	85
Chapter 12	Configuring the Import-Export Utilities	87
12.1	Introduction	87
12.2	Import-Export Configuration Files	88
12.3	Changing Your Configuration Settings	88
12.3.1	Creating a default_master.properties File	88
12.3.2	Location of Properties Files and JDBC Driver	88
12.3.3	Checking the js.jdbc.properties File	89
12.3.4	Checking the js.quartz.properties File	89
12.4	Deploying a JDBC Database Driver	90
12.5	Running Import or Export	90
12.5.1	Import-Export Access Events Option	90
12.6	Configuring the Import-Export Utility for JasperServer 3.7	90
Appendix A	Troubleshooting	91
A.1	Binary Installer Freezes	91
A.1.1	Installer Log Files	91
A.1.2	Installer DebugTrace Mode	92
A.2	Error Running Buildomatic Scripts	92
A.2.1	Missing Java JDK	92
A.2.2	Forgot to Copy the File ant-contrib.jar	92
A.2.3	Older Apache Ant Version	93
A.3	Unable to Edit Files on Windows 7	93
A.4	Bash Shell for Solaris, IBM AIX, HP UX and FreeBSD	93
A.5	Linux Installer Issue with Unknown Host Error	94
A.6	Problem Starting JasperReports Server on the Mac	94
A.7	Database-related Problems	95
A.7.1	Installation Error Using PostgreSQL 8.1	95
A.7.2	Database Connectivity Errors	95
A.7.3	Maximum Packet Size in MySQL	96
A.7.4	Database Error after Changing Database Port Number	96
A.7.5	Case Sensitivity for Table and Column Names	96
A.7.6	PostgreSQL: Job Scheduling Error	97
A.7.7	Error Running Scheduled Report	97
A.7.8	Error Running a Report	97
A.8	Application Server-related Problems	98
A.8.1	Memory Issues Running Under Tomcat	98
A.8.2	Java Out of Memory Error	98
A.8.3	Configuration File Locations	98
A.8.4	Context.xml under Tomcat: Special Case	98
A.8.5	Tomcat 6 Installed Using apt-get	99
A.8.6	GlassFish Modifications	99
A.8.7	JBoss Modifications	100

A.8.8	Disabling User Session Persistence in Application Servers	101
A.8.9	Session Error Using JasperReports Server and Tomcat 7	102
A.9	Problems Importing and Exporting Data from the Repository	102
A.9.1	Exporting a Repository That Contains UTF-8	102
A.9.2	Importing Scheduled Jobs with Update Option	103
A.10	Problems with Upgrade	103
A.10.1	Include Audit Events on Upgrade	103

CHAPTER 1 INTRODUCTION

JasperReports Server Community Project builds on JasperReports as a comprehensive family of Business Intelligence (BI) products, providing robust static and interactive reporting, report server, and data analysis capabilities. These capabilities utilize common metadata and provide shared services, such as security, a repository, and scheduling. The server exposes comprehensive public interfaces enabling seamless integration with other applications and the capability to easily add custom functionality.

JasperReports Server provides the ability to:

- Efficiently and securely manage many reports.
- Interact with reports, including sorting, changing formatting, entering parameters, and drilling on data.
- Schedule reports for distribution through email and storage in the repository.

For business intelligence users, Jaspersoft offers Jaspersoft OLAP, which runs on the server.

Jaspersoft provides several other sources of information to help extend your knowledge of JasperReports Server:

- Our Ultimate Guides document advanced features and configuration. They also include best practice recommendations and numerous examples. The guides are available as downloadable PDFs. Community project users can purchase individual guides or bundled documentation packs from the Jaspersoft [online store](#). Commercial customers can download them freely from the [support portal](#).
- Our free [Business Intelligence Tutorials](#) let you learn at your own pace, and cover topics for developers, system administrators, business users, and data integration users. The tutorials are available online from the Professional Services section of our [website](#).
- Free samples installed with JasperReports, iReport, and JasperReports Server, are documented online: [JasperReports Samples Overview](#) and [JasperReports Samples Reference](#). For more information, visit our [community website](#).

This chapter contains the following sections:

- **Conventions**
- **Java Version Supported**
- **JasperReports Server Distributions**
- **Release Notes**
- **System Requirements**
- **Support for Internationalization**

1.1 Conventions

This document uses the following conventions when referring to file locations:

Convention	Description
<js-install>	The root directory where JasperReports Server will be installed. For manual installations, the directory where you unpack the WAR file distribution.
<glassfish>	The directory where GlassFish is installed.
<java>	The directory where java is installed.
<jboss>	The directory where JBoss is installed.
<postgresql>	The directory where PostgreSQL is installed. If you use the instance of PostgreSQL that is bundled by the installer, <postgresql> is located in the <js-install> directory.
<tomcat>	The directory where Apache Tomcat is installed. If you use the instance of Tomcat that is bundled by the installer, <tomcat> is located in <js-install>.

1.2 Java Version Supported

JasperReports Server supports Java 1.6 and 1.7. Versions earlier than Java 1.6 are not supported.

JasperReports Server is tested and certified using Oracle/Sun Java. Other Java versions, such as OpenJDK, are known to have issues running JasperReports Server.

1.3 JasperReports Server Distributions

There are two main distribution packages for JasperReports Server.

Distribution Package	Description
Installer	Runs on Windows, Linux, and Mac OSX (32 or 64 bit).
WAR File Distribution Zip	Used for manual installation on Windows, Linux, Mac, and other platforms.

The installer distribution package installs JasperReports Server, automatically configures the JasperReports Server database, and, if you choose the Install Sample Data option, installs sample data for working with tutorials.

The WAR file binary distribution contains the JasperReports Server web archive file as well as scripts to create and load the database. The WAR file distribution supports additional applications that are not supported by the installers.

1.3.1 Installer Support

The installers support the following operating systems (32 and 64 bit):

Platform	Versions supported
Linux	Red Hat Enterprise Linux 5.4, 5.5 Novell SUSE Linux Enterprise 10.2, 11.1
Mac OSX	10.5 (Leopard) 10.6 (Snow Leopard)
Windows	XP Windows 7

1.3.1.1 Installer Naming for 32-bit and 64-bit

Native 32- and 64-bit installers are supported. The 64-bit installer will put 64-bit versions of Java 6 and PostgreSQL 9 onto your system for increased speed and performance.

The installer file naming distinguishes the 32-bit installer from the 64-bit installer.

Installer Type	Naming
32-bit installer	jasperreports-server-cp-4.7.0-windows-x86-installer.exe jasperreports-server-cp-4.7.0-linux-x86-installer.run jasperreports-server-cp-4.7.0-osx-x86-installer.app.zip
64-bit installer	jasperreports-server-cp-4.7.0-windows-x64-installer.exe jasperreports-server-cp-4.7.0-linux-x64-installer.run jasperreports-server-cp-4.7.0-osx-x64-installer.app.zip
Note: x86 is shorthand referring to the 386, 486, and 586 CPU architecture.	

Note: You can install the 32-bit installer onto a 64-bit operating system, but we recommend that you install the 64-bit installer onto a 64-bit system. The 64-bit installer will not execute on a 32-bit system.

1.3.1.2 Installer Distribution Components

The installer is designed to get JasperReports Server up and running quickly. The server requires the Java environment, an application server, and database to run. The installer distribution bundles these components:

Component	Description
JasperReports Server Application	WAR file and configuration support scripts.
JasperReports Server Documentation	Found in the <js-install>/docs directory.
Apache Tomcat	Web application container. You can use the bundled version or an existing version.
iReport Designer	JasperReports Designer that communicates directly with JasperReports Server for editing, uploading, or executing reports on the server (optionally installed).
Java 1.6 Runtime	Runs the web application container.
PostgreSQL Database	Database server. You can use the bundled version or an existing version.

1.3.1.3 Installing with Existing Components

You can choose to deploy the bundled application or if you have existing components, the installer can deploy to these components. Both Apache Tomcat and the PostgreSQL database can be independently used as bundled or existing instances.

If you would like the installer to install Tomcat, choose the bundled Tomcat. If you already have Tomcat on your computer you can choose an existing Tomcat.



If you use an existing Tomcat, it must be on the local machine.

If you use an existing PostgreSQL, it can be on a local or remote machine. If it's on a remote Linux machine, configure PostgreSQL to allow remote connections as described in [2.6.4, “Enabling Connections to a Remote Host,” on page 18](#).

For information about specific versions of third party applications supported by the installer, refer to the JasperReports Server release notes in the root of the installation directory.

1.3.1.4 Running Components as Windows Services

The Windows installer installs PostgreSQL and Tomcat as Windows Services. Users can manage JasperReports Server under the Windows operating system using Services in the Control Panel:

Control Panel > System and Security > Administrative Tools > Services

The bundled PostgreSQL and Tomcat applications restart automatically when the host Windows system restarts. If you do not want to run these components to automatically restart, you can change the Startup Type from automatic to manual.

You can find the PostgreSQL and Tomcat services under the following names:

- jasperreportsPostgreSQL
- jasperreportsTomcat

You can also start JasperReports Server from the Windows Start menu.

1.3.2 WAR File Binary Distribution Support

Use the WAR file binary distribution package to install the JasperReports Server application if you cannot use the installer. The WAR file supports more applications than the installer. If you want to use a database other than PostgreSQL and an application server other than Apache Tomcat, install JasperReports Server using the WAR file.



For a complete list of applications supported by the WAR file distribution, refer to the release notes that are included in the root directory of the distribution.

The target database can be on a remote server. Using a remote PostgreSQL database on some Linux platforms requires a change to its configuration file, as described in section [2.6.4, “Enabling Connections to a Remote Host,” on page 18](#).

The application server should reside on the local machine.

There are js-install shell scripts (for Linux and Window) included in the WAR file distribution which automate much of the installation tasks by using a single properties file. These scripts are named:

- js-install-ce.bat
- js-install-ce.sh

The main contents of the WAR file binary distribution are:

Content Item	Description
JasperReports Server js-install scripts	Found at <js-install>/buildomatic/js-install-ce.bat and js-install-ce.sh.
JasperReports Server Database Scripts	SQL scripts for each supported database.
JasperReports Server Documentation	Guides for end users and administrators.
JasperReports Server Extra Samples	Web Service example applications, sample reports, custom data source examples, and other sample files.
JasperReports Server Standard Sample Data	Sample data that highlights JasperReports Server features.
JasperReports Server WAR file archive	All of the JasperReports Server class files and dependent jars.

1.3.2.1 About Bundled Apache Ant

The War File Distribution ZIP comes with a bundled version of Apache Ant so you do not need to download or install Ant. The buildomatic Ant scripts come with Windows and Linux batch scripts that are pre-configured to use the bundled version of Apache Ant. The buildomatic Ant scripts are called from the command line in the following manner:

```
Windows:          js-ant <target-name>
Linux and Mac OSX: ./js-ant <target-name>
```

The bundled Apache Ant is version 1.8.1. This version or higher is required if you want to run your own version of Ant.

The bundled Apache Ant has an additional jar that extends Ant functionality. This jar is: ant-contrib.jar. This jar enables conditional logic in Ant. If you are running your own Ant you should copy the ant-contrib.jar to your Ant/lib folder.



On Linux and Solaris, the js-ant commands may not be compatible with all shells. If you have errors, use the `bash` shell explicitly. For more information, see section [A.4, “Bash Shell for Solaris, IBM AIX, HP UX and FreeBSD,”](#) on [page 93](#).

1.4 Release Notes

Release notes are included with each distribution and with each new update to a distribution.

Not all applications are immediately supported when a new JasperReports Server version is released. For instance, some applications require additional testing beyond what is completed for the initial General Availability (GA) release. To find out exactly what applications are supported with a particular distribution refer to the release notes found in that distribution.

1.5 System Requirements

The following table contains the minimum and recommended resources for a full installation that includes PostgreSQL and an application server. The values are based on our own testing. You may find that JasperReports Server can run on systems with fewer resources or slower systems than stated in the minimum resources column. At the same time, it is possible to run out of resources with the recommended configuration. The success of your deployment depends on the intended load of the system, the number of concurrent users, the data sets, and whether the databases are installed on the same system as the JasperReports Server.

Resource	Footprint	Minimum	Recommended
Disk	~700MB	10GB free	40GB +
RAM		3GB	4GB +
Processor		1.5GHz single Pentium, UltraSparc II, or equivalent	2.5GHz + multi-core Pentium for Windows, Mac, and Linux

1.6 Support for Internationalization

JasperReports Server supports the full Unicode character set using UTF-8 encoding. It also depends on the underlying database and application server to support the UTF-8 character encoding. If you use the bundled Tomcat and PostgreSQL software, UTF-8 is configured by default. If you use any other software, refer to the *JasperReports Server Administrator Guide* for instructions about configuring software to support UTF-8.

CHAPTER 2 INSTALLING JASPERREPORTS SERVER

This chapter contains the following sections:

- **Pre-Installation Steps**
- **Starting the Installer**
- **Accepting the License Agreement**
- **Choosing an Installation Directory**
- **Selecting a Tomcat Configuration**
- **Selecting a PostgreSQL Configuration**
- **Installing Sample Data**
- **Installing iReport Designer**
- **Completing the Installation**
- **Post-Installation Steps**

2.1 Pre-Installation Steps

When you run the installation executable, you are given the option to install a bundled Apache Tomcat application server and PostgreSQL database or to use an existing Tomcat and PostgreSQL.



If you want to use an existing database instance, the database must be running at install time. If you want to use an existing Apache Tomcat, the Tomcat instance should be stopped.

If you choose to install the bundled Tomcat and database, both are installed on the same host with the server.

2.2 Starting the Installer

In Windows, the installer is an executable file that you can double-click to run. For example, double-click the following:

```
jasperreports-server-cp-4.7.0-windows-x86-installer.exe    (32 bit)
jasperreports-server-cp-4.7.0-windows-x64-installer.exe    (64 bit)
```

In Linux, the installer is a .run file; you can run it from the command line or from a graphical environment. To start the installer from the command line, open a bash shell, and enter the name of the installer file. For example:

```
./jasperreports-server-cp-4.7.0-linux-x86-installer.run    (32 bit)
./jasperreports-server-cp-4.7.0-linux-x64-installer.run    (64 bit)
```

In Mac OSX, the installer is a .zip file. Typically, after download, the installer will be found in your <user>/Downloads folder, and it will already be unpacked. After the download is complete, double-click the following:

```
jasperreports-server-cp-4.7.0-osx-x86-installer.app      (32 bit)
jasperreports-server-cp-4.7.0-osx-x64-installer.app      (64 bit)
```

Whether you run the installer from the command line or in a graphical environment, you are prompted for the same information. The following sections describe these prompts, and assume you are in a graphical environment. If you are installing from the command line, use your keyboard to specify the same details. For example, with the license text, instead of clicking **I accept the agreement**, you press **Y** and press **Enter**.

The welcome screen introduces the installer and allows you to continue or exit. Click **Next**.

Note: If you are installing a 32-bit installer onto a 64-bit operating system you will normally get a popup message reminding you that a 64-bit installer is available. You may continue the 32-bit installation if you choose to.

2.3 Accepting the License Agreement

You are prompted to read and accept the license agreement. Read the agreement, agree to the terms by clicking **I accept the agreement**, and click **Next**. On the command line, you must page through several screens of text to read the full agreement.

If you do not accept the agreement, you must exit the installer.

2.4 Choosing an Installation Directory

You are prompted for the directory where JasperReports Server is installed, referred to as the <js-install> directory. Accept the default or click **Browse** and select a different location, and click **Next**. On the command line, press Enter to accept the default. To choose a different directory location, enter that location at the prompt.

The default <js-install> directory depends on your operating system:

```
Windows:  C:\Program Files\jasperreports-server-cp-4.7.0
Linux:    <USER_HOME>/jasperreports-server-cp-4.7.0
Mac OSX  /Applications/jasperreports-server-cp-4.7.0
```



On Linux, choose a <js-install> path that's no more than 84 characters.

2.5 Selecting a Tomcat Configuration

JasperReports Server requires an application server in order to run. The installer is pre-configured to run with the Apache Tomcat server. When you run the installer, two options appear on **Setup — Please select the Tomcat configuration you wish to use**:

- ♦ **I wish to use the bundled Tomcat**

If you choose this option, the installer puts an instance of Tomcat 6 onto your system. Later, after choosing a bundled or existing database, you are prompted for the server port and shutdown port that Tomcat will use. Most users accept the default values that are displayed. Accept the default values or enter alternate values, then click **Next**.

- ♦ **I wish to use an existing Tomcat**

If you already have an instance of Tomcat on your system, you can choose this option. Later, after choosing a bundled or existing database, you are prompted for the location of Tomcat. You can browse to the folder where you installed Tomcat, such as C:\Apache Software Foundation\Tomcat 6.

After selecting a PostgreSQL configuration, you are prompted for Tomcat's server port and shutdown port. Accept the default values, 8080 and 8005 by default, or enter alternate values.

2.6 Selecting a PostgreSQL Configuration

JasperReports Server requires a database in order to run. The installer is pre-configured to run with the PostgreSQL database. There are two options available for your PostgreSQL database:

- **I wish to use the bundled PostgreSQL database**
- **I wish to use an existing PostgreSQL database**

2.6.1 Choosing the Bundled PostgreSQL

If you choose the option to install the bundled PostgreSQL, the installer puts PostgreSQL 9 onto your system. The default PostgreSQL port 5432 will be used. If the installer finds that port 5432 is already in use, you are prompted to pick an alternate port. In this case, choose an alternative port value. The installer sets the PostgreSQL administrator password to postgres and also creates a PostgreSQL database user with administrator privileges and credentials of jasperdb/password.

The following table summarizes the parameters set during installation of the bundled PostgreSQL:

Parameter	Default Value and Description
Binary Directory	The directory where the postgres and pgAdmin3 binaries are located.
Port	The port number that PostgreSQL uses (default is 5432). User must choose an alternate port if 5432 is in use.
IP or Host Name	The IP address or name of the machine where PostgreSQL is installed. The default value is 127.0.0.1.
PostgreSQL Administrative Password	Password of the database administrative user: postgres. The installer cannot handle special characters at the end of a password string. Incompatible characters include: & ; \$
Database User Name	Hard coded default: jasperdb - The installer creates this user which is used to connect to the JasperReports Server database
Database User Password	Hard coded default: password - The installer uses this password for the jasperdb user.

2.6.2 Choosing an Existing PostgreSQL on a Local Host

If you choose the option to use an existing PostgreSQL database, you are eventually prompted for the location of PostgreSQL and the port to use. If you have an instance of PostgreSQL installed locally, accept the default, which is 127.0.0.1, the localhost. Accept the default location for the PostgreSQL \bin directory, or click **Browse** to locate and select another location. You are also prompted for the default administrative account password of the PostgreSQL administrative user. The database administrative user account name postgres is used by default. Enter the database administrative user password and click **Enter**.



If the installer displays an error message saying FATAL: password authentication failed for user postgres, try re-entering the administrative password for your PostgreSQL database.

The following table summarizes the parameters set during the installation of an existing PostgreSQL:

Defaults Used	Hardcoded Default Values Used or Created
PostgreSQL Administrative User Name	postgres - The default administrative database user.
jasperserver Database User Name	jasperdb - The installer creates this database user which is used to connect to jasperserver database.
jasperserver Database User Password	password - The installer creates this password for the jasperdb database user.



To improve system security, Jaspersoft recommends that you change the default password for jasperdb as soon as possible. To change the jasperdb connection password in JasperReports Server, edit: <js-install>/apache-tomcat/jasperserver/META-INF/context.xml. (And delete, if it exists: <js-install>/apache-tomcat/conf/Catalina/localhost/jasperserver.xml.) Then, make the same change in PostgreSQL using pgAdmin III.

2.6.3 Using an Existing PostgreSQL on a Remote Host

If you are installing to a remote instance of PostgreSQL, you need the PostgreSQL client tools on your local machine. The version of client tools should match the remote PostgreSQL version. You can check the version of PostgreSQL instance by entering this command on the computer where it's installed:

```
psql --version
or
<path-to-postgresql-bin-folder>/psql --version
```

For instance: C:/Program Files/PostgreSQL/9.0/bin/psql --version

To verify that you can connect to the target remote PostgreSQL from the local installation machine:

1. If necessary, install PostgreSQL client tools on your local, JasperReports Server machine.
2. Using your local PostgreSQL client tools, enter this command:

```
psql -U postgres -h <remote-host> -d postgres
or
<path-to-postgresql-bin-folder>/psql -U postgres -h <remote-host> -d postgres
```

You might also need to enable connections as described in the next section.

2.6.4 Enabling Connections to a Remote Host

On some platforms, the default PostgreSQL installation doesn't allow remote connections (as a security feature). You need to enable remote connections as described in this documentation:

- The PostgreSQL configuration documentation on the PostgreSQL web site
- The \docs directory of your PostgreSQL installation

To enable connections from the installation machine to the remote PostgreSQL server:

1. Locate the following PostgreSQL host-based authentication (hba) configuration file on the remote PostgreSQL server instance:

Windows: C:\Program Files\PostgreSQL\9.0\data\pg_hba.conf

Linux: /var/lib/pgsql/data/pg_hba.conf

2. Add the IP address of your local JasperReports Server installation machine to this file. For example, to allow the local installation machine with address 192.168.12.10 to connect to the PostgreSQL server, add this entry to the pg_hba.conf file:

```
host all all 192.168.12.10/32 trust
```

3. Allow TCP/IP connections to the remote PostgreSQL server instance by making the following change to the postgresql.conf file on the remote machine:
 From: `listen_addresses = 'localhost'`
 To: `listen_addresses = '*'`
4. Restart PostgreSQL.
5. Using your local PostgreSQL client tools, verify that you can connect to the target remote PostgreSQL from the local installation machine, as described in section 2.6.3, “Using an Existing PostgreSQL on a Remote Host,” on page 18.

2.7 Installing Sample Data

JasperReports Server can be installed with sample databases and sample reports for evaluating its features. Included are:

- Sugar CRM data that simulates three years of operations for a fictitious company that relies on the SugarCRM open source application.
- Foodmart data that simulates three years of operations for a fictitious company.
- JasperReports Server repository resources such as Reports, OLAP Views, Ad Hoc Topics, Domains, Data Sources, and Input Controls.
- Jaspersoft strongly recommends that you install this data, unless you are not interested in testing or evaluating with the default sample data.

During installation, the following prompt appears:

Would you like to install sample databases and sample reports?

Click **Yes** to install the sample data, and click **Next**.

2.8 Installing iReport Designer

iReport Designer is the primary GUI-based JasperReports Library creation tool. The installer pre-configures iReport to communicate with JasperReports Server. iReport communicates with JasperReports Server through a plug-in and can retrieve an existing JasperReports Library from a JasperReports Server instance for editing, uploading, or execution.

During installation, the following prompt appears:

Would you like to install iReport?

If you would like to install iReport, click **Next** to accept the **Yes** option.

The components are now ready for installation. Click **Install** or **Next** to continue. Installation can take a number of minutes.

2.9 Completing the Installation

After the files have been installed, you see the final installation screen. There are several post-installation options:

- **Launch JasperReports Server Now** - If you choose to launch JasperReports Server from the installer, the installer exits and the application server starts if you chose the bundled Tomcat and PostgreSQL. A pause that lasts approximately 25 seconds occurs as the server starts up, then the login page appears in your system default browser. If you’re installing under Linux, do not close the terminal window running the start script. For information about logging in, see section 3.4, “Logging into JasperReports Server,” on page 25.



The **Launch JasperReports Server Now** check box option will only be displayed if you have chosen to install a bundled Tomcat and a bundled PostgreSQL. The menu based start/stop scripts only control the bundled applications that you chose to be installed. For more information, see Chapter 3, “Starting and Stopping JasperReports Server,” on page 23.

- **Opt-in for JasperServer Heartbeat** - When the heartbeat is enabled, the server sends anonymous system and version information to Jaspersoft using https. JasperReports Server heartbeat information helps Jaspersoft create better products by improving our understanding of customer installation environments. For more information, see section [5.4.1, “JasperReports Server Heartbeat,”](#) on page 34.

Make your choices, then click **Finish**.

You should now be ready to log into the server.

2.10 Post-Installation Steps

2.10.1 Updates Made by the Installer During Installation

This section lists the standard updates that the installer makes to your local environment if you install to existing applications. When the installation completes, you can check that the updates, or corresponding changes, were successful.

Updates made to the application server

If you installed to an existing Tomcat, the following modifications to the Tomcat environment were attempted:

File or Directory	Updates
Windows: bin/setenv.bat Linux and Mac OSX: bin/setenv.sh	Creates this file. Sets increased Java memory allocation values to JAVA_OPTS. For additional settings, refer to section 6.2, “Setting JVM Options for Application Servers,” on page 43.
Tomcat 5: common/lib Tomcat 6 and 7: lib	Adds PostgreSQL JDBC driver to this directory.

Updates made to the PostgreSQL database

If you installed to an existing PostgreSQL database, new schemas and users are created in your database instance:

PostgreSQL Updates	Description
Database <code>jasperserver</code> created	This is the JasperReports Server repository database. This database holds all of system information, such as users, roles, datasources, and report definitions.
Database user <code>jasperdb</code> created	The JasperReports Server application uses this user to connect to the database.
Sample database <code>foodmart</code> created	(optional) Database created if install sample data option was chosen.
Sample database <code>sugarcrm</code> created	(optional) Database created if install sample data option was chosen.

2.10.2 Installer Output Log File Location

The installer creates a log during installation that records information as the installation progresses. If you encounter any problems when you install JasperReports Server, it can be helpful to look at the installer log. You can find the installer log at `<js-install>/installation.log`.

2.10.3 Checking your Java JVM Options

For both the bundled Tomcat and the existing Tomcat, the installer attempts to set Java JVM options to help with memory allocation. You can double-check the values set to see that they are appropriate for your installation. If you installed a bundled version of Tomcat from the installer, these are the default Java JVM options for heap memory allocation:

Installer Type	Setting	File Location
32 bit (x86) Windows	-Xms512m -Xmx1280m -XX:MaxPermSize=256m	<js-install>/apache-tomcat/bin/service.bat
32 bit (x86) Linux and Mac OSX	-Xms512m -Xmx1280m -XX:MaxPermSize=256m	<js-install>/apache-tomcat/scripts/ctl.sh
64 bit (x64) Windows	-Xms1024m -Xmx2048m -XX:MaxPermSize=512m	<js-install>/apache-tomcat/bin/service.bat
64 bit (x64) Linux and Mac OSX	-Xms1024m -Xmx2048m -XX:MaxPermSize=512m	<js-install>/apache-tomcat/scripts/ctl.sh

CHAPTER 3 STARTING AND STOPPING JASPERREPORTS SERVER

This chapter contains the following sections:

- **Start/Stop Menu — Windows**
- **Start/Stop Scripts — Linux**
- **Start/Stop Apps — Mac OSX**
- **Logging into JasperReports Server**
- **Starting the Included Jaspersoft iReport Designer**
- **JasperReports Server Log Files**

3.1 Start/Stop Menu — Windows

This section describes different start and stop procedures depending on how you installed JasperReports Server: using the bundled Tomcat and PostgreSQL or using an existing Tomcat and PostgreSQL.

3.1.1 Start/Stop Menus — Bundled Tomcat and PostgreSQL

If you chose to install a bundled Tomcat and a bundled PostgreSQL with JasperReports Server, use the Windows Start menu items to start and stop JasperReports Server.

To start or stop JasperReports Server from the Windows Start menu:

- Click **Start > All Programs > JasperReports Server CP 4.7.0 > Start or Stop Services > Start Service.**
- Click **Start > All Programs > JasperReports Server CP 4.7.0 > Start or Stop Services > Stop Service.**

3.1.2 Additional Information about the Bundled Tomcat and PostgreSQL

The Windows Control Panel—Services lists entries for Tomcat and PostgreSQL, which are installed as Windows Services by the JasperReports Server installer. These services are listed as:

- jasperreportsTomcat
- jasperreportsPostgreSQL

By default, these services are started automatically when you reboot. Consequently, the JasperReports Server also automatically restarts. You can change the startup mode for the services from automatic to manual.

To prevent JasperReports Server from starting up automatically:

1. In the Control Panel—Services, select jasperreportsTomcat.

2. Right-click the jasperreportsTomcat service, and select properties.
3. Change the Startup type drop-down setting Automatic to Manual.

When JasperReports Server is running, the Windows Task Manager lists information about the processes running under the SYSTEM user name:

- postgres.exe
- tomcat6.exe

3.1.3 Start/Stop Scripts – No Bundled Applications

During installation, if you chose to install one bundled and one existing Tomcat or PostgreSQL, you can use the Windows start/stop scripts to start and stop only the bundled one.

For example, if you have an existing Tomcat and you install the bundled PostgreSQL, the scripts and menus specified in the previous section would start and stop the PostgreSQL application. To start and stop the existing Tomcat, you would use the management scripts provided by the Tomcat application.



JasperReports Server needs to have database and application servers started in this order:

- First, start the database server.
- Next, start the application server.

3.2 Start/Stop Scripts – Linux

This section describes different start and stop procedures depending on how you installed JasperReports Server: using the bundled Tomcat and PostgreSQL or using an existing Tomcat and PostgreSQL.

3.2.1 Manual Start/Stop

You typically start and stop JasperReports Server at the Linux command line. Run the following commands in a Linux shell.

Start JasperReports Server:

```
cd <js-install>
./ctlscript.sh start
```

Stop JasperReports Server:

```
cd <js-install>
./ctlscript.sh stop
```

To start and stop individual components:

```
cd <js-install>
./ctlscript.sh start|stop postgresql
./ctlscript.sh start|stop tomcat
```

3.2.2 Auto Start/Stop with Bundled Tomcat and PostgreSQL

To have JasperReports Server automatically start when you reboot your Linux server, you need to install the JRS database and application server as services. If you have installed JasperReports Server using the binary installer with the bundled Tomcat and bundled PostgreSQL options, an example jasperserver service script can be found in the following location:

```
<js-install>/scripts/linux/jasperserver
```

Edit this script and set permissions as described in the <js-install>/scripts/linux/readme file in the same location.

Once installed, these services are started automatically when you reboot. Consequently, the JasperReports Server also automatically restarts.

3.3 Start/Stop Apps – Mac OSX

After you complete the Mac OSX installation, you typically find JasperReports Server installed in the following location:

```
/Applications/jasperreports-server-cp-4.7.0
```

When JasperReports Server is running, you can see the names of the Java and PostgreSQL processes in the Activity Monitor.

To start JasperReports Server, locate this folder in Finder and double-click the following app:

```
jasperServerStart.app
```

To stop JasperReports Server, locate this folder in Finder and double-click the following app:

```
jasperServerStop.app
```

The Mac lists the following information in the Activity Monitor:

- java
 - or
 - org.apache.catalina.startup.Bootstrap
- postgres

3.3.1 Start/Stop Apps – Mac Dock

Using Finder, move the following apps into the Mac Dock to start, stop, and login to JasperReports Server:

- jasperServerStart.app
- jasperServerStop.app
- jasperServerLogin.app

3.3.2 Start/Stop JasperReports Server – Mac Terminal Shell

To start and stop JasperReports Server using the Mac terminal shell:

1. Open a Terminal shell (Finder > Go > Utilities > Terminal Icon).
2. Navigate to the <js-install> folder. For instance: /Applications/jasperreports-server-cp-<ver>
3. To start PostgreSQL, Tomcat, and JasperReports Server, enter:


```
./ctlscript.sh start
```
4. To shutdown PostgreSQL, Tomcat, and JasperReports Server, enter:


```
./ctlscript.sh stop
```
5. To start and stop individual components:

```
cd <js-install>
./ctlscript.sh start|stop postgresql
./ctlscript.sh start|stop tomcat
```

3.4 Logging into JasperReports Server

To log into JasperReports Server on any operating system:

1. Start JasperReports Server.
2. Open a supported browser: Firefox, Internet Explorer, Chrome, and Safari.
3. Log into JasperReports Server by entering the startup URL in your browser's address field. The URL depends upon your application server. If you installed the default, bundled Tomcat use:

```
http://<hostname>:8080/jasperserver
```

- <hostname> is the name or IP address of the computer hosting JasperReports Server.

- ♦ 8080 is the default port number for the Apache Tomcat application server. If you used a different port when installing your application server, specify its port number instead of 8080.

The login page appears.

4. Log in using the following credentials:

User ID	Password	Description
jasperadmin	jasperadmin	Administrator

If you installed the sample data, the sample end-user is also created. This user is a non-administrative user who have fewer system privileges than an administrative user.

User ID	Password	Description
joeuser	joeuser	Sample end-user



When you complete the evaluation or testing of your JasperReports Server instance, change the administrator password (jasperadmin) and remove any sample end-users. Leaving the default passwords and end-users in place weakens the security of your installation.

To log into JasperReports Server on Windows:

On Windows, you can launch the login page from the desktop of the JasperReports Server host computer by clicking **Start > All Programs > JasperReports Server CP 4.7.0 > JasperReports Server Login**.

To log into JasperReports Server on Mac OSX:

On Mac OSX, you can launch the login page by going to Finder and clicking the following script:

```
/Applications/<js-install>/jasperServerLogin
```

For example: `/Applications/jasperreports-server-cp-4.7.0/jasperServerLogin`

To use the Dock to log into JasperReports Server:

From Finder, you can drag the `/Applications/<js-install>/jasperServerLogin.app` to the Dock to handle logging into JasperReports Server using your default system browser.

3.5 Starting the Included Jaspersoft iReport Designer

This section describes how to start the iReport Designer that you installed using the JasperReports Server installer.

Windows:

Click **Start > All Programs > JasperReports Server CP 4.7.0 > Start iReport Designer**.

Linux:

Enter the following commands in a terminal window:

```
cd <js-install>
ireport/bin/iReportLoader.sh
```

Mac OSX:

1. From Finder, double-click the `iReport-mac.dmg` file in the root of the JasperReports Server installation. A new window appears.
2. Double-click the iReport Designer application.

3.6 JasperReports Server Log Files

Log files contain important information about JasperReports Server operations. If your application server is Tomcat, JBoss, or GlassFish, the log output goes to one of the following files:

Tomcat: <tomcat>/webapps/jasperserver/WEB-INF/logs/jasperserver.log

JBoss: <jboss>/server/default/deploy/jasperserver.war/WEB-INF/logs/jasperserver.log

GlassFish: <glassfish>/domains/domain1/autodeploy/jasperserver.war/WEB-INF/logs/jasperserver.log

You can configure the log outputs and logging levels in the log4j.properties file in the WEB-INF folder.

To change the logging levels while you are running JasperReports Server:

1. Browse to http://<hostname>:8080/jasperserver/log_settings.html.

The Log Settings page appears.

2. Change logging levels using the drop-down menus.

Changes to logging levels affect only the current session of JasperReports Server. Logging levels revert to default settings as defined in the properties files at the next startup.

For more information about system logging, see the *JasperReports Server Administrator Guide*.

CHAPTER 4 UNINSTALLING JASPERREPORTS SERVER

This chapter contains the following sections:

- [Windows](#)
- [Linux](#)
- [Mac OSX](#)
- [Uninstall Survey](#)

4.1 Windows

To uninstall JasperReports Server on Windows 7:

Click **Start > All Programs > JasperReports Server CP 4.7.0 > Uninstall JasperReports Server**.

To uninstall JasperReports Server on Windows XP:

1. Click **Start > All Programs > JasperReports Server CP 4.7.0 > Uninstall JasperReports Server**.
Typically a popup window asks which user account to run as.
2. Uncheck the check box that says:
Protect my computer and data from unauthorized program activity



The uninstaller will not execute if **Protect my computer and data from unauthorized program activity** is checked.

4.2 Linux

Under Linux, the `<js-install>` folder includes an executable that removes JasperReports Server from the host.

To uninstall JasperReports Server:

1. From the command line, log in as the root user (or any user with sufficient privileges).
2. Enter the following commands:

```
cd <js-install>  
./uninstall
```

3. Respond Y or yes to the prompt that asks if you want to remove JasperReports Server from this computer.

4.3 Mac OSX

To use Finder to uninstall JasperReports Server:

1. Navigate to the <js-install> folder.
For example: /Applications/jasperreports-server-cp-4.7.0
2. Click the uninstall.app to launch the uninstaller.

4.4 Uninstall Survey

After running the uninstaller, you are prompted to take an uninstall survey from Jaspersoft. Survey answers are anonymous and help Jaspersoft improve the products we make. When you click **Yes**, the survey launches on the Jaspersoft web site in a new browser window. Select all the reasons that led you to uninstall JasperReports Server, or enter a short explanation if none match. Thank you for your feedback.

CHAPTER 5 INSTALLING THE WAR FILE DISTRIBUTION

In addition to the installer binaries, you can install the JasperReports Server application using the stand-alone WAR file distribution. If you do not want to use the installer or your target configurations, such as your database or application server, are unsupported by the installer, use the WAR file distribution. Download the WAR file distribution from [Jaspersoft technical support](http://support.jaspersoft.com) (<http://support.jaspersoft.com>) or contact your sales representative. The WAR file distribution comes in a file named `jasperreports-server-4.7.0-bin.zip` in the compressed ZIP format.

This chapter contains the following sections:

- [Applications Supported by the WAR File Distribution](#)
- [Installing the WAR File Using js-install Scripts](#)
- [Starting JasperReports Server](#)
- [Starting JasperReports Server](#)
- [Logging into JasperReports Server](#)
- [Installing the WAR File Manually](#)
- [Troubleshooting Your JasperReports Server Configuration](#)
- [Running the Import and Export Utilities](#)

5.1 Applications Supported by the WAR File Distribution

5.1.1 Database and Application Server Support

The instructions in this and subsequent chapters support the following configurations:

Database	Application Server
PostgreSQL MySQL	Apache Tomcat JBoss GlassFish

For version information about these databases and application servers refer to the release notes in the root of the unpacked distribution ZIP.

5.1.2 Operating System Support for Bash Shell

JasperReports Server is a Java Web Application. Therefore, it supports all operating system platforms where Java is fully supported. However, for the `js-install` shell scripts (described in the section below), the default shell required is the bash shell. Here is a list of shells required:

Operating System	Required Shell for <code>js-install</code> scripts	System Default Shell	Script to Run
Windows	CMD shell	CMD shell	<code>js-install-ce.bat</code>
Linux	Bash shell	Bash shell	<code>js-install-ce.sh</code>
Solaris	Bash shell	Korn shell (ksh)	<code>js-install-ce.sh</code>
IBM AIX	Bash shell	Korn shell (ksh)	<code>js-install-ce.sh</code>
HP UX	Bash shell	Posix shell (posix/sh)	<code>js-install-ce.sh</code>
FreeBSD	Bash shell	C shell (tcsh)	<code>js-install-ce.sh</code>

5.2 Installing the WAR File Using `js-install` Scripts

Follow the steps in this procedure to install JasperReports Server using WAR file distribution. The `js-install` shell scripts, supported on Windows, Linux, and Mac OSX, do most of the work for you.

To meet prerequisites for installing the WAR file:

1. Install the Oracle/Sun Java JDK 1.6 or 1.7. There are known bugs with other Java implementations such as OpenJDK.
2. Create and set the `JAVA_HOME` system environment variable.
3. Locate or install one of the following application servers:
 - Apache Tomcat 5.5, 6, or 7
 - JBoss 5.1 or 7.1
 - Glassfish 2.1 or 3.0 using the default domain (domain1)

If you use GlassFish 3.1.0 or a custom domain, see section [A.8.6, “GlassFish Modifications,” on page 99](#).
4. Locate or install PostgreSQL or MySQL database.



The target database can be on a remote server. The application server should reside on the local machine.

If you would like to run a pre-install validation test, you can run `js-install-ce.bat test` or a similar command. For more information about how to perform a validation test in your environment, see section [5.6.3.1, “js-install Script Test Mode,” on page 37](#).

To install the WAR file using `js-install` scripts:

The scripts are intended for the bash shell.



If installing to non-linux Unix platforms such as HP-UX, IBM AIX, FreeBSD, or Solaris the bash shell is required for using the `js-install` scripts.

1. Extract all files from `jasperreports-server-cp-4.7.0-bin.zip`. Choose a destination, such as `C:\Jaspersoft` on Windows, `/home/<user>` on Linux, or `/Applications` on Mac OSX.

The directory, `jasperreports-server-cp-4.7.0-bin`, appears in the file location you choose.

2. Copy the <database>_master.properties file for your database from sample_conf and paste it to buildomatic:
 - ♦ Copy from — <js-install>/buildomatic/sample_conf/
 - ♦ Paste to — <js-install>/buildomatic
 For example, copy postgresql_master.properties to <js-install>/buildomatic.
3. Rename the file you copied to default_master.properties.
4. Edit the default_master.properties file to add the settings for your database and application server. [Table 5-1](#) lists sample property values for each supported database.

Table 5-1 Sample Values for the default_master.properties File

Database	Sample Property Values
PostgreSQL	<pre>appServerType=tomcat6 [tomcat7, tomcat5, jboss, jboss7, glassfish2, glassfish3, skipAppServerCheck*] appServerDir=c:\\Program Files\\Apache Software Foundation\\Tomcat 6† dbUsername=postgres dbPassword=postgres dbHost=localhost</pre>
MySQL	<pre>appServerType=tomcat6 [tomcat7, tomcat5, jboss, jboss7, glassfish2, glassfish3, skipAppServerCheck*] appServerDir=c:\\Program Files\\Apache Software Foundation\\Tomcat 6† dbUsername=root dbPassword=password dbHost=localhost</pre>

* When the property appServerType is set to skipAppServerCheck, buildomatic skips any application server validation.

† Backslashes in paths must be doubled in properties files, for example appServerDir=c:\\Apache Software Foundation\\Tomcat 7.

Valid values for appServerType in the default_master.properties file are:

- ♦ tomcat5
- ♦ tomcat6
- ♦ tomcat7
- ♦ jboss (use for JBoss 5.1)
- ♦ jboss7
- ♦ glassfish2
- ♦ glassfish3

If Linux Tomcat is installed using apt-get, see section [A.8.5, “Tomcat 6 Installed Using apt-get,” on page 99](#).

5. Run the js-install scripts.
 - a. Start your database server.
 - b. Stop your application server.
 - c. Open Command Prompt as Administrator on Windows or open a terminal window on Linux and Mac OSX.
 - d. Run the appropriate commands:

Commands	Description
cd <js-install>/buildomatic	
js-install-ce.bat (Windows) ./js-install-ce.sh (Linux and Mac OSX)	Installs JasperReports Server, sample data, and sample databases (foodmart and sugarcrm)
js-install-ce.bat minimal (Windows) ./js-install-ce.sh minimal (Linux and Mac OSX)	Installs JasperReports Server, but does not install sample data and sample databases

If you encounter errors during the js-install script execution, see section 5.6.3, “Error Running js-install Scripts (js-install-ce.bat/sh),” on page 36.

- Set Java JVM Options, as described in 6.2, “Setting JVM Options for Application Servers,” on page 43. This step is required.



To view the output log, look in this location: <js-install>/buildomatic/logs/js-install-<date>-<number>.log

5.3 Starting JasperReports Server

To run JasperReports Server:

- First, start your application server using one of these commands:

Tomcat:	Windows	<tomcat>/bin/startup.bat
	Linux and Mac OSX	<tomcat>/bin/startup.sh
JBoss:	Windows	<jboss>/bin/run.bat
	Linux and Mac OSX	<jboss>/bin/run.sh
GlassFish:	Windows, Linux, and Mac OSX	asadmin start-domain domain1

- Start JasperReports Server as described in Chapter 3, “Starting and Stopping JasperReports Server,” on page 23.

To view the JasperReports Server application logs, see section 3.6, “JasperReports Server Log Files,” on page 27.

5.4 Logging into JasperReports Server

After JasperReports Server starts up, login by going to this URL:

http://<hostname>:8080/jasperserver

Example:

http://localhost:8080/jasperserver

http://jasperserver.example.com:8080/jasperserver

The login page appears after compiling the necessary JSP file.

Use one of the following sets of credentials to log into JasperReports Server:

User ID	Password	Description
jasperadmin	jasperadmin	Administrator

If you logged in successfully, your JasperReports Server home page appears.



When you complete the evaluation or testing of your JasperReports Server instance, change the administrator password (jasperadmin) and remove any sample end-users. Leaving the default passwords and end-users in place weakens the security of your installation.

Refer to the *JasperReports Server User Guide* to begin adding reports and other objects to the server.

5.4.1 JasperReports Server Heartbeat

After initially logging into JasperReports Server, you are asked to opt-in to the JasperReports Server Heartbeat.

To opt-in, click **OK**. To opt-out, click the check box to remove the check and click **OK**.

The heartbeat helps Jaspersoft understand customer installation environments to improve our products. If you choose to enable the heartbeat, an HTTPS call at server startup time sends information like this to Jaspersoft:

- Operating System type and version
- JVM type and version
- Application Server type and version
- Database type and version
- JasperReports Server type and version
- Unique, anonymous identifier value

You can manually enable or disable the heartbeat by modifying the following bean in `jasperserver/WEB-INF/applicationContext-logging.xml` file:

```
bean id = "heartBean"
```

To disable the heartbeat, set the `enabled` property of the `heartBean` to `false`:

```
<property name="enabled" value="false"/>
```

For additional information about enabling and disabling the heartbeat component, see the *JasperReports Server Administrator Guide*.

5.5 Installing the WAR File Manually

In some case, you may need to install the WAR file manually when you cannot use the `js-install` scripts.

The manual buildomatic steps described in this procedure execute the same Ant targets as the `js-install` scripts (`js-install-ce.sh/.bat`). The procedure shows which buildomatic targets to execute manually if, for some reason, you are unable to use the `js-install` scripts.

If you're using MySQL, place a MySQL JDBC driver in `<js-install>/buildomatic/conf_source/db/mysql/jdbc`.

To install the WAR file distribution using the manual buildomatic steps:

1. Start your database server.
2. Stop your application server.
3. Create and edit a `default_master.properties` file to add the settings in for your database and application server as described in [5.2, "Installing the WAR File Using js-install Scripts," on page 32](#).
4. Open a Command Prompt as Administrator on Windows or open a terminal window on Linux and Mac OSX, and run these commands:

Table 5-2 Buildomatic Targets to Execute to Install the WAR File

Commands	Description
<code>cd <js-install>/buildomatic</code>	Makes the buildomatic directory your current directory.
<code>js-ant create-js-db</code>	Creates the JasperReports Server repository database.
<code>js-ant create-sugarcrm-db</code> <code>js-ant create-foodmart-db</code>	(Optional) Creates the sample databases.
<code>js-ant load-sugarcrm-db</code> <code>js-ant load-foodmart-db</code> <code>js-ant update-foodmart-db</code>	(Optional) Loads sample data into the sample databases.
<code>js-ant init-js-db-ce</code> <code>js-ant import-minimal-ce</code>	Initializes the <code>jasperserver</code> database, loads core application data. Running <code>js-ant import-minimal-ce</code> is mandatory. The server cannot function without this data.

Table 5-2 Buildomatic Targets to Execute to Install the WAR File

Commands	Description
<code>js-ant import-sample-data-ce</code>	(Optional) Loads the demos that use the sample data.
<code>js-ant deploy-webapp-ce</code>	Configures and deploys the WAR file to Tomcat, JBoss, or Glassfish.



On non-Linux Unix platforms, the `js-ant` commands may not be compatible with all shells. If you have errors, use the `bash` shell explicitly. For more information, see section [A.4, “Bash Shell for Solaris, IBM AIX, HP UX and FreeBSD,”](#) on page 93.

If you encounter an error when running `create-sugarcrm-db`, `create-foodmart-db`, or `create-js-db`, you can create the JasperReports Server database manually using the database administration tool for your particular database type. To create the JasperReports Server database manually for PostgreSQL or MySQL, see section [6.1, “Manually Creating the JasperReports Server Database,”](#) on page 41.

If you have previously installed the databases, you can drop the old versions and then recreate the databases. To do this, run the following drop commands before running the commands in [Table 5-2](#):

Table 5-3 Buildomatic Targets to Execute to Delete Sample Databases

Commands	Description
<code>js-ant drop-js-db</code>	(Optional) Deletes the JasperReports Server repository database.
<code>js-ant drop-sugarcrm-db</code> <code>js-ant drop-foodmart-db</code>	(Optional) Deletes the sample databases.

- Set Java JVM Options, as described in section [6.2, “Setting JVM Options for Application Servers,”](#) on page 43. This step is required.

5.6 Troubleshooting Your JasperReports Server Configuration

This section describes the most common installation problems.

5.6.1 JasperReports Server Startup Problems

If you encounter a problem trying to run a new JasperReports Server, an incorrect database configuration is the likely culprit. Another common cause is a mistake in the application server configuration files. For information about resolving these types of errors, see [Appendix A, “Troubleshooting,”](#) on page 91.

5.6.2 Error Running a Report

If you have trouble running reports in your new JasperReports Server instance, see section [A.7.8, “Error Running a Report”](#) in [Appendix A, “Troubleshooting,”](#) on page 91.

5.6.3 Error Running js-install Scripts (js-install-ce.bat/sh)

The `js-install` script creates an output log that captures standard output and error output. If you encounter problems during the execution of the script, or if you want to remember which options you chose, open the output log file.

To troubleshoot problems running js-install scripts:

- Open the output log file located in:

```
<js-install>/buildomatic/logs/js-install-<date>-<number>.log
```

2. Try to find the first error encountered by the js-install steps.
 - a. Go to the end of the output log.
 - b. Scroll back through lines of error messages until you find the first error logged. Typically, this error causes more errors later in the log.

Finding the original error is critical to understanding the problem and often difficult because the output log tends to be long. The js-install scripts execute a Java-based import operation that puts required minimal and optional sample data in place. Unfortunately, Java stack traces can be very long. Additionally, JasperReports Server uses the Spring framework for flexibility in tying application components together. An error that involves a Spring initialization XML file can also cause a long stack trace.

Incorrect settings in the `default_master.properties` file cause most problems, which you correct can correct by recreating your `default_master.properties` settings. Common errors are:

- Typos in the path for the application server
- Misspelling the hostname or password for the database

To recreate your `default_master.properties` settings:

1. Open the file in `<js-install>/buildomatic`, make corrections, and save it.
2. Re-run the js-install script.
The js-install script uses the current values in the `default_master.properties` file.

To help isolate errors, run the js-install scripts in `test` mode.

5.6.3.1 js-install Script Test Mode

You can run the js-install and js-upgrade scripts in `test` mode using the `test` option. In `test` mode, the js-install scripts check your `default_master.properties` settings and validate the application server location and connection to the specified database. Using `test` mode can help debug issues, such as an incorrect database password. Your system isn't altered when executing the script in `test` mode.

To run the js-install script in test mode on Windows:

1. Navigate to the buildomatic directory:
`cd <js-install>/buildomatic`
2. Enter the following command to run the js-install script in `test` mode:
`js-install-ce.bat test`

To run the js-install script in test mode on Linux or Mac OSX:

1. Navigate to the buildomatic directory:
`cd <js-install>/buildomatic`
2. Enter the following command to run the js-install script in `test` mode:
`./js-install-ce.sh test`

5.6.4 Problem Connecting to a Cloud Database Instance

A cloud database instance (such as Amazon EC2) typically disables unused IP ports. When the js-install script runs, it validates the database hostname using the built-in ant operation `<isreachable>`. This operation is similar to a network ping and may cause a hang issue if the port is unavailable. In this case, the `validateHost` step can be commented out in the `buildomatic/validation.xml` file. See the comment in the `do-pre-install-test` target.

5.7 Running the Import and Export Utilities

The buildomatic scripts automatically configure the database information needed by the buildomatic import and export functionality. This functionality is invoked through ant targets used by buildomatic. The scripts for importing and exporting repository content are located in the following directory:

```
cd <js-install>/buildomatic
```

This section describes the Ant targets and parameter settings you need to specify to send the options to the import and export commands. For complete information about the import-export options, see the *JasperReports Server Administrator Guide*.

5.7.1 Running Export from Buildomatic

The export target for ant has the following syntax:

```
Windows:  js-ant export -DexportFile=<filename> -DexportArgs="<export-options>"
```

Linux and

```
Mac OSX:  ./js-ant export -DexportFile=<filename> -DexportArgs=\"<export-options>\"
```

The export file format is a ZIP file or a set of files under a new directory name. If you specify the .zip extension for your output filename, a ZIP archive is created automatically. Otherwise, a directory with files and sub-directories is created as a uncompressed set of files.

The exportArgs argument requires double quotation marks (") and can contain more than one export option, as shown in these Windows examples:

```
js-ant export-help-ce

js-ant export-ce -DexportFile=my-reports.zip
-DexportArgs="--uris /reports"

js-ant export-ce -DexportFile=my-reports-and-users.zip
-DexportArgs="--uris /reports
--users jasperadmin,joeuser"

js-ant export-ce -DexportFile=my-datasources
-DexportArgs="--uris /datasources --roles ROLE_USER"

js-ant export-ce -DexportFile=js-everything.zip -DexportArgs="--everything"
```

On Linux, all double quotation marks (") and other characters, such as the vertical bar (|) must be escaped with a backslash (\). In addition, when listing user names, enclose the list in single quotation marks ('), as shown in this Linux example:

```
./js-ant export-help-ce
./js-ant export-ce -DexportFile=my-reports-and-users.zip
-DexportArgs=\"--uris /reports
--users 'jasperadmin,joeuser'\"
```

5.7.2 Running Import from Buildomatic

The import target for ant has the following syntax:

```
Windows:  js-ant import-ce -DimportFile=<filename> [-DimportArgs="<import-options>"]
```

Linux and

```
Mac OSX:  ./js-ant import-ce -DimportFile=<filename> [-DimportArgs=\"<import-options>\"]
```

The imported file is handled as a ZIP archive if its name ends in .zip, otherwise it is handled as a directory. The `importArgs` argument is optional and can contain more than one import option. On Linux, all double quotation marks (") must be escaped with a backslash (\).

The following examples are typical import commands on Windows:

```
js-ant import-help-ce

js-ant import -DimportFile=my-reports.zip

js-ant import -DimportFile=my-datasources -DimportArgs="--update"
```

The following examples are typical import commands on Linux:

```
./js-ant import-help-ce

./js-ant import -DimportFile=my-reports.zip

./js-ant import -DimportFile=my-datasources.zip -DimportArgs=\"--update\"
```

5.7.3 Running the Import-Export Shell Scripts

The import-export shell scripts are distinct from the buildomatic Ant import-export functionality. If you installed JasperReports Server using the WAR file distribution, you typically need to configure the import-export shell scripts, as described in [Chapter 12, “Configuring the Import-Export Utilities,” on page 87](#). For instance, you probably need to specify a JDBC driver for your database that is different from the one specified in the script.



The JDBC drivers are not in place for the MySQL database. You need to obtain and add it to the installation as described in [5.2, “Installing the WAR File Using js-install Scripts,” on page 32](#).

If you installed JasperReports Server using the binary installer, the scripts are already properly configured.

These import-export shell scripts are available for Windows and Linux/Mac OSX:

- Windows:
 - <js-install>/buildomatic/js-export.bat
 - <js-install>/buildomatic/js-import.bat
- Linux and Mac OSX:
 - <js-install>/buildomatic/js-export.sh
 - <js-install>/buildomatic/js-import.sh

These scripts used to be in the <js-install>/scripts directory.

Run the import-export shell scripts with the help option to list the other options that you can use. For example:

```
js-export.bat --help
js-import.sh --help
```


CHAPTER 6 MANUAL DATABASE CREATION AND OTHER INFORMATION

This chapter contains the following sections:

- [Manually Creating the JasperReports Server Database](#)
- [Setting JVM Options for Application Servers](#)
- [Locating and Changing Buildomatic Configuration Files](#)
- [Configuring Report Scheduling](#)
- [Updating XML/A Connection Definitions](#)

6.1 Manually Creating the JasperReports Server Database

If you can't use the js-install scripts to create the JasperReports Server database and the sample databases, create them manually.

To manually create the JasperReports Server database:

1. Follow these instructions to create the repository database and optional sample databases:
 - [6.1.1, “PostgreSQL 8.x and 9.x,” on page 42](#)
 - [6.1.2, “MySQL,” on page 42](#)

The commands in these sections have been tested at Jaspersoft, but the commands you need to use on your database instance may be different.

2. If you did not install the optional sample databases, complete the installation by executing these commands:

```
js-ant import-minimal-ce
js-ant deploy-webapp-ce
```

If you installed the optional sample databases, complete the installation by executing this command:

```
js-ant deploy-webapp-ce
```

For more information about executing the Ant scripts, see section [5.5, “Installing the WAR File Manually,” on page 35](#).

3. Set Java JVM Options, as described in section [6.2, “Setting JVM Options for Application Servers,” on page 43](#). This step is required.

6.1.1 PostgreSQL 8.x and 9.x

To manually create the JasperReports Server database in PostgreSQL:

1. On the Windows, Linux, or Mac OSX command line, enter these commands to create and initialize the JasperReports Server database:

```
cd <js-install>/builddomatic/install_resources/sql/postgresql

psql -U postgres -W
postgres=#create database jasperserver encoding='utf8';
postgres=#\c jasperserver;
postgres=#\i js-create.ddl
postgres=#\i quartz.ddl
postgres=#\q
```

2. (Optional) Run the following commands if you want to install sample databases:

```
cd <js-install>/builddomatic/install_resources/sql/postgresql

psql -U postgres -W
postgres=#create database sugarcrm encoding='utf8';
postgres=#create database foodmart encoding='utf8';
postgres=#\c sugarcrm;
postgres=#\i sugarcrm.sql; (first make sure the file is unzipped)
postgres=#\c foodmart;
postgres=#\i foodmart-postgresql.sql; (first make sure the file is unzipped)
postgres=#\i supermart-update.sql;
postgres=#\q
```

3. Continue from [step 2 in section 6.1, "Manually Creating the JasperReports Server Database," on page 41](#).

6.1.2 MySQL

To manually create the JasperReports Server database in MySQL:

The MySQL client software, `mysql.exe` or `mysql`, can be used to interact with the MySQL database.



If you are going to access MySQL on a remote server you should run a command similar to the following grant statement:

```
mysql>grant all on *.* to jasperdb@'%';
```

1. On the Windows, Linux, or Mac OSX command line, enter these commands to create and initialize the JasperReports Server database:

```
cd <js-install>/builddomatic/install_resources/sql/mysql

mysql -u root -p
mysql>create database jasperserver character set utf8;
mysql>grant all on *.* to jasperdb@'% ' identified by 'password';
mysql>flush privileges; (reload privilege tables)
mysql>use jasperserver;
mysql>source js-create.ddl
mysql>source quartz.ddl
mysql>exit
```

2. (Optional) Run these commands to install sample databases:

```
cd <js-install>/buildomatic/install_resources/sql/mysql
mysql -u root -p
mysql>create database sugarcrm;
mysql>create database foodmart;
mysql>use sugarcrm;
mysql>source sugarcrm.sql; (first make sure the file is unzipped)
mysql>use foodmart;
mysql>source foodmart-mysql.sql; (first make sure the file is unzipped)
mysql>source supermart-update.sql;
mysql>exit
```

3. Continue from [step 2 in section 6.1, "Manually Creating the JasperReports Server Database," on page 41](#).

6.2 Setting JVM Options for Application Servers

JasperReports Server is supported on Java 1.6 and 1.7. Java Virtual Machine (JVM) runtime parameters need to be correctly set to avoid conflicts with JasperReports Server's AXIS-based web service classes. These conflicts could cause web services and the resources that rely on them, such as XML/A connections, to fail.

The options you need and how you set them depends on your version of Java, your application server, and how it is deployed.

The settings in this section apply specifically to the Oracle/Sun JVM. Other JVMs may or may not have equivalent settings.

6.2.1 Tomcat and JBoss JVM Options

The following tables present some typical settings of JVM options that affect JasperReports Server. For information about changing a JVM option setting for your particular environment, see your application server documentation.



These example settings are for 64-bit systems. For 32-bit systems, see ["Checking your Java JVM Options" on page 21](#).

JVM Options on Windows	
Options for Java 1.6 and 1.7	set JAVA_OPTS=%JAVA_OPTS% -Xms1024m -Xmx2048m -XX:PermSize=32m -XX:MaxPermSize=512m -Xss2m -XX:+UseConcMarkSweepGC -XX:+CMSCClassUnloadingEnabled
Additional options for Java 1.6-1.7 and JBoss	set JAVA_OPTS=%JAVA_OPTS% -Djavax.xml.soap.MessageFactory=org.apache.axis.soap.MessageFactoryImpl -Djavax.xml.soap.SOAPConnectionFactory=org.apache.axis.soap.SOAPConnectionFactoryImpl -Djavax.xml.soap.SOAPFactory=org.apache.axis.soap.SOAPFactoryImpl
Additional option for JBoss 5.1	-Djavax.xml.transform.TransformerFactory=org.apache.xalan.processor.TransformerFactoryImpl

JasperReports Server doesn't provide a virtual X frame buffer on Linux. If your Linux applications are graphical, set the `-Djava.awt.headless=true` to prevent Java from trying to connect to an X-Server for image processing.

JVM Options on Linux and Mac OSX	
Options for Java 1.6 and 1.7	<code>export JAVA_OPTS="\$JAVA_OPTS -Xms1024m -Xmx2048m -XX:PermSize=32m -XX:MaxPermSize=512m -Xss2m -XX:+UseConcMarkSweepGC -XX:+CMSClassUnloadingEnabled"</code>
Additional options for Java 1.6-1.7 and JBoss	<code>set JAVA_OPTS=%JAVA_OPTS% -Djavax.xml.soap.MessageFactory=org.apache.axis.soap.MessageFactoryImpl -Djavax.xml.soap.SOAPConnectionFactory=org.apache.axis.soap.SOAPConnectionFactoryImpl -Djavax.xml.soap.SOAPFactory=org.apache.axis.soap.SOAPFactoryImpl</code>
Additional option for JBoss 5.1	<code>-Djavax.xml.transform.TransformerFactory=org.apache.xalan.processor.TransformerFactoryImpl</code>

There are a number of ways to set JVM options. Sections 6.2.2 - 6.2.5 present step-by-step instructions for performing this task. Alternatively, you can add your `JAVA_OPTS` settings to any one of these files:

File	Add JVM Options After This Line on Windows
<code><tomcat>/bin/setclasspath.bat</code>	<code>set JAVA_ENDORSED_DIRS=%BASEDIR%\common\endorsed</code>
<code><tomcat>/bin/setenv.bat</code>	JAVA_OPTS setting can go anywhere in this file.
<code><jboss>/bin/run.bat (JBoss 5.1)</code>	<code>set JAVA_OPTS=%JAVA_OPTS% -Dprogram.name=%PROGNAME%</code>
<code><jboss>/bin/standalone.bat (JBoss 7)</code>	<code>rem Setup JBoss specific properties</code>

File	Add JVM Options After This Line on Linux
<code><tomcat>/bin/setclasspath.sh</code>	<code>JAVA_ENDORSED_DIRS="\$BASEDIR"/common/endorsed</code>
<code><tomcat>/bin/setenv.sh</code>	JAVA_OPTS setting can go anywhere in this file.
<code><jboss>/bin/run.sh (JBoss 5.1)</code>	<code>export JAVA_OPTS="\$JAVA_OPTS -Dprogram.name=\$PROGNAME"</code>
<code><jboss>/bin/standalone.sh (JBoss 7)</code>	Add JAVA_OPTS setting before <code>#Display</code> our environment

6.2.2 Changing JVM Options for Bundled Tomcat as a Windows Service

As of release 4.0, the Windows installer binary installs the bundled Tomcat application as a Windows Service by default. The steps to change JVM options are:

1. Navigate to this directory:
`cd <js-install>/apache-tomcat/bin`
2. Open `service.bat` for editing.
3. Look for the following line to change the JVM heap size, for example:

```
"%EXECUTABLE%" //US//%SERVICE_NAME% --Startup auto --JvmOptions "-Xms1024M;-Xmx1536M;-Xss2M;-Dcatalina.base=%CATALINA_BASE%;-Dcatalina.home=%CATALINA_HOME%;-Djava.endorsed.dirs=%CATALINA_HOME%\endorsed" --StartMode jvm --StopMode jvm
```

4. Update this line to increase the maximum heap size from 1536M to 2048M, for example:
`-Xmx2048M`

- Because Tomcat is installed as a service, you need to re-install the service. From a Windows Command Prompt, enter these commands:

```
cd <js-install>\apache-tomcat\scripts
serviceinstall.bat REMOVE
serviceinstall.bat INSTALL
```

After running these commands in Windows XP, for example, the cmd shell closes after the commands are executed. The Tomcat service is removed and then installed. After execution of the commands, the service is running.

- Stop and restart PostgreSQL and Tomcat after completing this work as described in [Chapter 3, “Starting and Stopping JasperReports Server,”](#) on page 23.

6.2.3 Changing JVM Options for Existing Tomcat as a Windows Service

If you installed JasperReports Server to use an existing Tomcat (not the bundled version) that is running as a Windows service, you can set Java options on the Java Tab of the Tomcat Properties dialog:

- Launch the Tomcat configuration application from the Windows Start menu:
 - Start > Programs > Apache Tomcat > Configure Tomcat**
- In the Apache Tomcat Properties dialog, click the **Java** tab.
- In the Java Options field, add your `JAVA_OPTS` values according to the table above.
 - Enter only the options preceded by `-X` or `-D`, not `set JAVA_OPTS=%JAVA_OPTS%`.
 - Enter only one Java option setting per line.
- For instance, add options as follows:



These example settings are for 64-bit systems. For 32-bit systems, see [“Checking your Java JVM Options”](#) on page 21.

```
-Xms1024m
-Xmx2048m
-XX:PermSize=32m
-XX:MaxPermSize=512m
-Xss2m
```

- Click Apply, then click OK.
- Stop and restart PostgreSQL and Tomcat after completing this work as described in [Chapter 3, “Starting and Stopping JasperReports Server,”](#) on page 23.

6.2.4 Changing JVM Options for Bundled Tomcat on Linux

If you installed JasperReports Server to use the bundled Tomcat, you can set Java options by editing the appropriate Tomcat configuration script. The steps to change JVM options are:

- Navigate to the scripts directory:


```
cd <js-install>/apache-tomcat/scripts
```
- Open `ctl.sh` for editing.
- Look for the `start_tomcat()` function and locate the `JAVA_OPTS` variable inside it.

4. Modify the `JAVA_OPTS` values according to the example above:

```
start_tomcat() {
    is_tomcat_running
    ...
    export JAVA_OPTS="-Xms1024m -Xmx2048m -XX:PermSize=32m -XX:MaxPermSize=512m -
Xss2m -XX:+UseConcMarkSweepGC -XX:+CMSClassUnloadingEnabled"
    ...
}
```



There may be more than one occurrence of the `JAVA_OPTS` variable in the `ctl.sh` file. Make sure to edit the instance inside the `start_tomcat()` function.

5. Save and close the `ctl.sh` file.
6. Stop and restart PostgreSQL and Tomcat as described in [Chapter 3, “Starting and Stopping JasperReports Server,” on page 23](#).

6.2.5 Changing GlassFish JVM Options

The following sections describe how to set the JVM options for GlassFish for Java 1.6 and 1.7 using the command line or a configuration file.

6.2.5.1 Setting GlassFish JVM Options with `asadmin` Command

1. First make sure your GlassFish instance is up and running, then enter the following command as a single line:

```
asadmin create-jvm-options -Xms1024m:-Xmx2048m:-XX\:PermSize=32m:
-XX\:MaxPermSize=512m:-Xss2m:-XX\:+UseConcMarkSweepGC:
-XX\:+CMSClassUnloadingEnabled:
-Djavax.xml.soap.MessageFactory=org.apache.axis.soap.MessageFactoryImpl:
-Djavax.xml.soap.SOAPConnectionFactory=org.apache.axis.soap.SOAPConnectionFactoryImpl:
-Djavax.xml.soap.SOAPFactory=org.apache.axis.soap.SOAPFactoryImpl:
```

2. Restart the application server using the following commands:

```
asadmin stop-domain domain1
asadmin start-domain domain1
```

When running the `asadmin create-jvm-options` command, error messages like this might appear:

```
[exec] CLI167 Could not create the following jvm options. Options exist:
[exec] -Xmx512m
[exec] CLI137 Command create-jvm-options failed.
```

This message indicates that one of the options specified was already set in the JVM. The command will succeed for all other JVM options on the command line. No further action is necessary.

6.2.5.2 Setting GlassFish JVM Options by Editing `domain.xml`

1. Open the `<glassfish>/domains/domain1/config/domain.xml` configuration file for editing.

2. Add the following lines to the section entitled `java-config`:

```
<jvm-options>-Xms1024m -Xmx2048m -XX:PermSize=32m -XX:MaxPermSize=512m -Xss2
-XX:+UseConcMarkSweepGC -XX:+CMSClassUnloadingEnabled
-Djavax.xml.soap.MessageFactory=org.apache.axis.soap.MessageFactoryImpl
-Djavax.xml.soap.SOAPConnectionFactory=org.apache.axis.soap.SOAPConnectionFactoryImpl
-Djavax.xml.soap.SOAPFactory=org.apache.axis.soap.SOAPFactoryImpl
</jvm-options>
```

3. If you are modifying the settings for a running instance of GlassFish, restart the application server using the following commands:

```
asadmin stop-domain domain1
asadmin start-domain domain1
```

6.3 Locating and Changing Buildomatic Configuration Files

The Ant-based buildomatic scripts contain support files for setting up and configuring a number of databases and application servers. This section describes the locations and content of some of these files and how to change the content.

6.3.1 Regenerating Buildomatic Settings

Whenever you change your `default_master.properties` file and re-run the `js-install` scripts (or any other buildomatic target), your generated configuration settings are automatically updated. The generated settings are in this location:

```
<js-install>/buildomatic/build_conf/default
```

The settings are automatically regenerated based on the new timestamp found on the properties file.

If you want to explicitly cause your generated configuration to be regenerated, you can run the following buildomatic targets:

```
cd <js-install>/buildomatic
js-ant clean-config
js-ant gen-config
```

The first target clears the configuration template files in `buildomatic/build_conf/default` directory. The second rebuilds the configuration settings.



These commands exist as a convenience. Whenever `default_master.properties` is edited, the resulting configuration templates are regenerated automatically based on the updated time-stamp associated with the edited file.

6.3.2 Locating Buildomatic-Generated Property Files

After you set your database and application server property values, you initiate buildomatic which automatically generates the database and application server configuration files needed to prepare for a JasperReports Server installation.

The generated property files are in this location:

```
<js-install>/buildomatic/build_conf/default
```

Some of the key configuration files are:

```
js.jdbc.properties
js.quartz.properties
js-glassfish-ds.xml
js-jboss-ds.xml
maven_settings.xml - (used for source code build)
```

More generated property files are:

```
<js-install>/buildomatic/build_conf/default/webapp
```

Included in the /webapp directory are configuration files, such as:

```
META-INF/context.xml
```

```
WEB-INF/hibernate.properties
```

```
WEB-INF/js.quartz.properties
```

These autogenerated files are removed if you run the buildomatic target: `clean-config`. You can then regenerate the files by running the target: `gen-config`. (Also, after running `clean-config`, any subsequent target will regenerate the configuration files.)

6.3.3 Changing the JDBC Driver Deployed by Buildomatic

When you run the buildomatic target `deploy-webapp-ce` the JDBC driver for your specified database is copied to your application server.

If there is a different or more up-to-date JDBC driver that you prefer to use, change the driver used by updating your `default_master.properties` file:

```
<js-install>/buildomatic/default_master.properties
```

In `default_master.properties`, set the `maven.jdbc.artifactId` and the `maven.jdbc.version` to point to the name of the driver you would like to use:

```
maven.jdbc.artifactId=<first-part-of-filename>
```

```
maven.jdbc.version=<version-part-of-filename>
```

Make sure that the property settings and the JDBC driver name exactly match, as shown in examples in the following sections.

Put new JDBC drivers in the same location as the existing JDBC drivers for your DB type. The buildomatic scripts will look in the default `jdbc` folder location described in [6.3.4, “Buildomatic Location for JDBC Drivers,” on page 49](#) that is associated with your DB type:

```
<js-install>/buildomatic/conf_source/db/<dbType>/jdbc
```

If you plan to run the `js-import.bat/sh`, `js-export.bat/sh` shell scripts, manually update the JDBC driver version found in the `<js-install>/buildomatic/conf_source/ieCe/lib` folder.

6.3.3.1 PostgreSQL Driver Example

You can use the same logic above to deploy other JDBC drivers. For example, you want to use the JDBC4 PostgreSQL driver and automatically deploy the driver to your application server when you run the `deploy-webapp-ce` target. However, the default driver setting, shown in the following file, is JDBC3:

```
<js-install>/buildomatic/conf_source/db/postgresql/db.properties
```

```
maven.jdbc.artifactId=postgresql
```

```
maven.jdbc.version=9.0-801.jdbc3
```

```
So the driver used is: postgresql-9.0-801.jdbc3.jar
```

To change the default driver setting, edit your `default_master.properties` file, add the following lines:

```
maven.jdbc.artifactId=postgresql
```

```
maven.jdbc.version=9.0-801.jdbc4
```

Now, when you run `deploy-webapp-ce`, the driver used is: `postgresql-9.0-801.jdbc4.jar`.

6.3.4 Buildomatic Location for JDBC Drivers

The installer provides the JDBC drivers for PostgreSQL. If you use MySQL, obtain the JDBC driver (mysql-connector-java-5.1.17-bin.jar or later is recommended) and put it in the location shown in this table:

Database	Buildomatic JDBC Driver Location and Sample JARs
PostgreSQL	<js-install>/buildomatic/conf_source/db/postgresql/jdbc/postgresql-9.0-801.jdbc3.jar <js-install>/buildomatic/conf_source/db/postgresql/jdbc/postgresql-9.0-801.jdbc4.jar
MySQL	<js-install>/buildomatic/conf_source/db/mysql/jdbc/mysql-connector-java-5.1.17-bin.jar (not provided)

6.3.5 Buildomatic Location for JasperReports Server WAR File

Buildomatic takes the JasperReports Server WAR file from the root of the <js-install> directory:

```
<js-install>/jasperserver.war
```

When you run the `deploy-webapp-ce` target, buildomatic takes the war archive and unpacks it into your application server. Next, the database configuration files needed by the application server are copied to the appropriate locations. For instance, in the case of Tomcat:

- <js-install>/jasperserver.war
Unpacked and copied to <tomcat>/webapps/jasperserver/*
- <js-install>/buildomatic/build_conf/default/webapp/META-INF/context.xml
Copied to <tomcat>/webapps/jasperserver/META-INF/context.xml
- <js-install>/buildomatic/build_conf/default/webapp/WEB-INF/hibernate.properties
Copied to <tomcat>/webapps/jasperserver/WEB-INF/hibernate.properties
- <js-install>/buildomatic/build_conf/default/webapp/WEB-INF/js.quartz.properties
Copied to <tomcat>/webapps/jasperserver/WEB-INF/js.quartz.properties
- <js-install>/buildomatic/build_conf/db/postgres/jdbc/postgresql-9.0-801.jdbc4.jar
Copied to <tomcat>/lib

6.3.6 Buildomatic Location for SQL Scripts

Buildomatic comes with SQL scripts and other utilities that support a number of databases. These files are in:

```
<js-install>/buildomatic/install_resources/sql/
```

For example, some key files are (same pattern for additional databases):

```
<js-install>/buildomatic/install_resources/sql/postgresql/js-create.ddl
<js-install>/buildomatic/install_resources/sql/postgresql/quartz.ddl
<js-install>/buildomatic/install_resources/sql/postgresql/upgrade-postgresql-3.7.0-4.0.0-ce.sql
<js-install>/buildomatic/install_resources/sql/postgresql/js-drop.ddl
<js-install>/buildomatic/install_resources/sql/postgresql/drop-quartz.ddl
```



You can run these scripts manually by copying them to a location where your database client software is located.

6.3.7 Buildomatic Location for Database Creation Scripts

For most databases the buildomatic scripts are able to create the metadata repository database used by JasperReports Server. This is the database where the data defining users, roles, data sources, reports, OLAP views, domains, and other data are stored. This database is normally named `jasperserver`.

Buildomatic attempts to create the `jasperserver` database via JDBC when the `create-js-db` target is executed.

The scripts and property files used to create the jasperserver database are here:

```
<js-install>/buildomatic/conf_source/db/  
    postgresql/scripts.properties  
    mysql/scripts.properties
```

6.3.8 Buildomatic Location for Sample Data Catalog ZIP Files

Buildomatic includes export files which hold the JasperReports Server sample data (that have examples of new features). This sample data is loaded when you run the buildomatic target `import-sample-data-ce`, for instance. These export files along with other important export files are located here:

```
<js-install>/buildomatic/install_resources/export/
```

Here are some key files (same pattern for additional databases):

```
js-catalog-postgresql-minimal-ce.zip  
js-catalog-postgresql-ce.zip  
js-catalog-mysql-minimal-ce.zip  
js-catalog-mysql-ce.zip
```

6.3.9 Hibernate Properties Settings

Your hibernate.properties settings are in the following directory after buildomatic has been run to automatically generate your configuration files:

```
<js-install>/buildomatic/build_conf/default/webapp/WEB-INF/hibernate.properties
```

Within the jasperserver WAR file the hibernate.properties file is found at the following location:

```
<appserver-path>/jasperserver/WEB-INF/hibernate.properties
```

The buildomatic scripts automatically create this configuration file. When you run the buildomatic target `deploy-webapp-ce` this file is copied to JasperReports Server in your application server.

Hibernate property values are:

```
PostgreSQL metadata.hibernate.dialect=com.jaspersoft.hibernate.dialect.PostgresqlNoBlobDialect  
MySQL 5.1 metadata.hibernate.dialect=org.hibernate.dialect.MySQLInnoDBDialect  
MySQL 5.5 metadata.hibernate.dialect=org.hibernate.dialect.MySQL5InnoDBDialect
```

6.3.10 Database Connection Configuration Files

6.3.10.1 Tomcat

After setting up the buildomatic configuration for your database, the Tomcat context.xml will be automatically created with the appropriate settings for JasperReports Server.

When the buildomatic target `deploy-webapp-ce` is run, the context.xml will be automatically copied into the jasperserver WAR set of files.

You can view the automatically generated context.xml at the following location:

```
<js-install>/buildomatic/build_conf/default/webapp/META-INF/context.xml
```

The final location of the context.xml is:

```
<tomcat>/webapps/jasperserver/META-INF/context.xml
```

Tomcat will often create a copy of the context.xml file with a changed name that will be read instead of the one found in the jasperserver war file. This is often a source of confusion for Tomcat users who attempt change their database settings. If you change your settings, delete the file in this location:

```
<tomcat>/conf/Catalina/localhost
```

6.3.10.2 JBoss

After setting up the buildomatic configuration for your database, the JBoss data source definition file will be automatically created with the appropriate settings for JasperReports Server.

When the buildomatic target `deploy-webapp-ce` is run, the `js-jboss-ds.xml` will be automatically copied into the JBoss instance.

You can view the automatically generated `js-jboss-ds.xml` at the following location:

`<js-install>/buildomatic/build_conf/default/js-jboss-ds.xml` (JBoss 5.1)

`<js-install>/buildomatic/build_conf/default/js-jboss7-ds.xml` (JBoss 7.1)

The final location of the `js-jboss-ds.xml` is:

`<jboss>/server/default/deploy/js-jboss-ds.xml` (JBoss 5.1)

`<jboss>/server/default/deploy/js-jboss7-ds.xml` (JBoss 7.1)

When JasperReports Server is running under JBoss, there are a couple of INFO log messages and an XML/A connection error that might occur depending on the version of JBoss you are running with.

For more information, refer to troubleshooting section [A.8.7, “JBoss Modifications,” on page 100](#).

6.3.10.3 Glassfish

After setting up the buildomatic configuration for your database, the Glassfish data source definition file `js-glassfish-ds.xml` will be automatically created with the appropriate settings. When the buildomatic target `deploy-webapp-ce` is run, the file is automatically deployed to the Glassfish instance.

You can view the automatically generated `js-glassfish-ds.xml` at the following location:

`<js-install>/buildomatic/build_conf/default/js-glassfish-ds.xml`

To deploy the datasource definition manually, you can run a command similar to the following:

```
asadmin add-resources "<js-install>/buildomatic/build_conf/default/js-glassfish-ds.xml"
```

6.4 Configuring Report Scheduling

The JasperReports Server report scheduling feature is powered by the Quartz scheduler tool. The configuration settings for Quartz-based report scheduling is automatically handled by buildomatic.

In a deployed JasperReports Server instance, you will find the `js.quartz.properties` file in the following location:

`<app-server-path>/jasperserver/WEB-INF/js.quartz.properties`

For mail server configuration, there is an additional property setting for authentication in the following file:

`<app-server-path>/webapps/jasperserver/WEB-INF/applicationContext-report-scheduling.xml`

There are four main configurations to be discussed in this section:

- Mail Server Configuration
- Quartz Driver Delegate Class
- Report Scheduler Web URI
- Quartz Table Prefix

6.4.1 Mail Server Configuration Settings

If you schedule reports or run them in the background, you can specify email addresses to notify when the report completes. To use this feature, configure JasperReports Server to contact an email server:

Configuration File		
<app-server>/<deployment>/WEB-INF/js.quartz.properties		
Property	Description	
report.scheduler.mail.sender.host	The name of the computer hosting the mail server	
report.scheduler.mail.sender.username	The name of the user in the mail server that JasperReports Server can use	
report.scheduler.mail.sender.password	The password of the mail server user	
report.scheduler.mail.sender.from	The address that appears in the From field on email notifications	
report.scheduler.mail.sender.protocol	The protocol that the mail server uses. JasperReports Server only supports SMTP. Note: Your entry must be lower case. For example: smtp	
report.scheduler.mail.sender.port	The port number that the mail server uses. For SMTP, the default is typically 25 (values other than 25 may not work in earlier JasperServer versions).	
Configuration File		
<app-server>/<deployment>/WEB-INF/applicationContext-report-scheduling.xml		
Property	Bean	Description
javaMailProperties key="mail.smtp.auth"	reportScheduler MailSender	If your mail server requires authentication, change this property from false to true.

6.4.2 Database Settings for the Quartz Driver Delegate Class

Quartz uses the Quartz driver delegate class to interact with the JDBC driver.



If you used buildomatic to install JasperReports Server, the correct value of the Quartz driver delegate class is set automatically for your database.

If you didn't use buildomatic to install JasperReports Server, use this table to edit the js.quartz.properties file and set the value of the Quartz driver delegate class to the correct value for your database:

Configuration File		
<app-server>/<deployment>/WEB-INF/js.quartz.properties		
Property	Database	Value
quartz.delegateClass	PostgreSQL	org.quartz.impl.jdbcjobstore.PostgreSQLDelegate
	MySQL	org.quartz.impl.jdbcjobstore.StdJDBCDelegate

6.4.3 Settings for the Report Scheduler Web URI

JasperReports Server uses the Report Scheduler Web URI to construct the link it sends in the output from a scheduled job. This link must be correct for the user to access the report on the server.

The port on which you run JasperReports Server and context root of the deployed JasperReports Server web application determine the report scheduler Web URI. The default context root is `jasperserver`.

To set this value manually, edit the following file: `<app-server>/<deployment>/WEB-INF/js.quartz.properties`. Change the properties for your application server as shown in the following table.

Property	App Server	Example Value
<code>report.scheduler.web.deployment.uri</code>	Apache Tomcat	<code>http://localhost:8080/jasperserver</code>
	JBoss	<code>http://localhost:8080/jasperserver</code>
	GlassFish	<code>http://localhost:8080/jasperserver</code>

6.4.4 Settings for the Quartz Table Prefix

For databases that support schemas, such as PostgreSQL, you can set the Quartz table prefix so that it includes the schema if you use one.



If you installed JasperReports Server using buildomatic the Quartz table prefix is set automatically.

To set this value, edit the file `<app-server>/<deployment>/WEB-INF/js.quartz.properties`. Change the following property:

Property	Description
<code>quartz.tablePrefix</code>	The prefix for the quartz table, including any schema name.

6.4.5 Settings for Import-Export

If you are manually configuring the import-export shell scripts instead of using the buildomatic, make sure your settings for the Quartz driver delegate class property are correct for the database you are using.



If you installed using buildomatic these settings are handled automatically (in buildomatic import-export).

To configure the import-export scripts manually, edit the following file:

`<js-install>/buildomatic/conf_source/ieCe/js.quartz.properties`

Change the following properties:

Property	Description
<code>quartz.delegateClass</code>	Set to the same value as described in section 6.4.2, “ Database Settings for the Quartz Driver Delegate Class ,” on page 52.
<code>quartz.tablePrefix</code>	Set to the same value as described in section 6.4.4, “ Settings for the Quartz Table Prefix ,” on page 53

6.5 Updating XML/A Connection Definitions

Sample XML/A connections are included with the JasperReports Server sample data. If you plan to use XML/A Web Services in your environment, then you may want to check and possibly update the hard coded values in the sample connections.

If you have Jaspersoft OLAP enabled, JasperReports Server is able to make XML/A connections over the Web Services interface. These HTTP-based connections use a user account for authentication. You may have different usernames and passwords than the defaults that get loaded from the sample data load in the sections above. Additionally, your application server hostnames and port values might be different than the default values.

There are two sample OLAP views that use this connection:

- Foodmart Sample XMLA OLAP View
- SugarCRM Sample XMLA OLAP View

If you would like to validate and update these resources, do the following:

1. Log into JasperReports Server as an administrator (such as `jasperadmin`).
2. Navigate to the Repository Management page by selecting the **View > Repository** menu item.
3. Click to expand the Analysis Components folder, then the Analysis Connections folder. Click to highlight the **Foodmart XML/A Connection** resource, then click **Edit**.
4. Edit the following information on this screen:
 - URI (hostname and port)
 - Login Username
 - Login Password
5. Click **Next**, then **Save**.
6. Make the same updates for the **SugarCRM XML/A Connection** resource.

CHAPTER 7 PLANNING YOUR 4.7.0 UPGRADE

Some of the new and enhanced features in JasperReports Server 4.7.0 can significantly affect your deployment and you should plan your upgrade accordingly. Prior to upgrading to 4.7.0, you should make sure to back up your current JasperReports Server installation and repository.

This chapter describes the following changes:

- **Themes:** An upgraded user interface eliminates unnecessary white space and lets users see more on a single screen. There have been changes to the resources and CSS used by themes, which make it easier to build and maintain new themes. However, existing themes will have to be redesigned in order to work in 4.7. See [“Upgrading Themes in 4.7.0” on page 55](#) for details on how to upgrade themes.
- **Data Snapshots:** Reports stored in JasperReports Server can now be configured to store snapshots of the report data in the repository resource. Data snapshots load quickly without querying the data source, thus increasing performance and reducing database load. If data snapshots are turned on, size requirements for your repository may increase dramatically. Data snapshots are disabled globally by default. See [“Data Snapshots” on page 63](#) for more information on data snapshots.

For an overview of new features, improvements, and bug fixes see the release notes in the root directory of the distribution. For information on how to use the new features, see the *JasperReports Server User Guide* or the *JasperReports Server Administrator Guide*.

This chapter contains the following sections:

- [Upgrading Themes in 4.7.0](#)
- [Data Snapshots](#)

7.1 Upgrading Themes in 4.7.0

The look and feel of the JasperReports Server web interface has been redesigned to increase usable space and be more compatible across browsers. In addition, navigation has been improved with simplified menus and a new Library page that provides quick access to all reports, dashboards, and views a user may access. To accomplish this, images, markup, and styles have been modified and many elements and images are smaller in size. As a result of these modifications, custom themes developed for the previous interface will need to be updated for the new interface.

This section details the changes made to the user interface and describes some of the steps necessary to update custom themes. For information on developing new themes, see the *JasperReports Server Administrator Guide*.

7.1.1 Banner and Toolbar Modifications

Banner settings have changed in 4.7.0, which means you will need to modify your theme to work with the new banner.

Table 7-1 shows the elements you need to modify and their default locations. The default values for these elements are in the default.css file.

Table 7-1 Banner and Toolbar Settings

Element	Classname and Modification	File	Notes
Banner	.banner Give custom value to height.	containers.css	Default value: height:26px
Banner Logo	#logo Give custom values to height and width that match the dimensions of your logo.	theme.css	Default values: height: 20px width: 115px
Banner Main Navigation	.menu.primaryNav .wrap Set height and line-height to the same measurement as .banner .	containers.css	Default values: height: 26px line-height: 26px
Banner Main Navigation	.menu.primaryNav .wrap.over .menu.primaryNav .wrap.pressed If you already have values for these defined in your alternate theme, then you need to change height and line-height to match the height of .banner.	containers.css	Not explicitly defined, but a value of 26px for both elements is cascaded from .menu.primaryNav .wrap If the over and pressed effects fill the banner height after applying the modifications to .menu.primaryNav .wrap, then this step is not necessary.
Banner Main Navigation Home icon	.menu.primaryNav #main_home .wrap > .icon Set height to be 2px shorter than the height of .banner. Set values for width and background-position to fit your image.	containers.css	Default value: background-position: left -166px. Height should be two pixels shorter than banner height. Height is not explicitly defined, but a value of 24px is cascaded from .button .icon in buttons.css Width is not explicitly defined, but a value of 14px is cascaded from .menu.primaryNav #main_home .wrap > .icon in the containers.css file.
Banner Main Navigation Item arrow icon	.menu.primaryNav .node > .wrap > .icon Set height to your desired value, with the maximum value being the same height measurement as the .banner element. Set background-position to a value that properly displays the default or your custom image.	containers.css	Default values: height: 30px; background-position: 0 -78px background-position: 0 -78px (IE8-9) background-position: 0 -79px (Ch+Saf) If you are using Chrome or Safari, the classname that you apply background position to must be preceded by "body:nth-of-type(1)".
Banner Main Navigation Item arrow icon	.menu.primaryNav .wrap.over .menu.primaryNav .wrap.pressed Set background-position to a value that properly displays the default or your custom image.	containers.css	background-position is not explicitly defined. The value is cascaded from .menu.primaryNav .node > .wrap > .icon This only needs to be adjusted if you want a different color disclosure indicator for the pressed and over states of the main menu links.

Table 7-1 Banner and Toolbar Settings

Element	Classname and Modification	File	Notes
Banner Metadata	#metalinks li Set line-height to the desired value that will vertically center it within the banner.	theme.css	Default value: line-height: 9px
Banner Search container	#globalSearch.searchLockup Set margin-top to desired value that will vertically center it within the banner.	controls.css	Default value: margin-top:3px
Body	#frame Set a custom top value that positions the body of the application below the banner.	containers.css	Default value: top: 28px This value needs to be greater than the value you apply to the height of .banner.
Toolbar	.toolbar Set custom height value that will fit your capsule buttons.	containers.css	Default value: height: 28px This is only necessary if you have customized capsule buttons that differ in height from the default images.

7.1.2 Changes to IE Overrides

All style rules have been moved out of the IE-specific CSS files and into the base CSS files. IE-specific notations have been applied to these rules so they can be condensed into the class declarations with the other style rules for a given element. IE-specific styles are commented as shown below:

- The following comment indicates a general IE style rule:
`* IE *`
- The following comment indicates a style rule specific to IE7:
`* IE7 *`
- The following comment indicates a style rule specific to IE8 and IE9:
`* IE8 and IE9 *`

Table 7-2 shows an example of an IE7 style rule that has been combined into the base style rules in 4.7.

Table 7-2 Changes to Implementation for IE Style Rules

IE overrides (4.5.1 and earlier)	lists.css (4.5.1 and earlier)	lists.css (4.7)
<code>.stepIndicator .icon { *top: 2px; }</code>	<code>.stepIndicator .icon { margin: 0; top: 1px; margin-right: 3px; }</code>	<code>.stepIndicator .icon { position: relative; margin: 0px; top: 1px; *top: 2px; * IE7 \<\ margin-right: 3px; }</code>

7.1.3 Images in JasperReports Server 4.7.0

Images for navigation, buttons, and backgrounds in the default theme have changed in version 4.7.0. Some images have been deleted or moved into a sprite file with a different name, other images have been modified and their properties (such as height

and width) may have changed. You need to ensure your custom themes refer to these images. If you have used custom images, you may need to adjust them to work with the new default scheme. This section lists the image files and indicates the changes.

Table 7-3 lists the images that have been added in the 4.7 default theme.

Table 7-3 New Images in 4.7.0 Default Theme

Image	Usage	Notes
banner_bkgd.png	Background gradient image for application banner	
column_header_bkgd.png	Background gradient image for <code>.header</code> elements in columns and dialogs	Replaces <code>panel_dialog_header_sprite.png</code> image used for dialog box headers in 4.5.1.
inner_pagination_sprite.png	Pagination icons (next, previous, etc) for dashboard report widgets	
spacer.gif	Icon that appears in the column header of spacer columns in Ad Hoc Editor	
viewer_toolbar_buttons_sprite.png	Button icons for the Ad Hoc Editor toolbar	

Table 7-4 lists the images that have been modified for the 4.7 default theme.

Table 7-4 Modified Images in 4.7.0 Default Theme

Image	Usage	Notes
adhoc_toolbar_buttons_sprite.png	Button icons for the Ad Hoc Editor toolbar	Some icons are new. All icons have new shadow effect.
button_action_primary_sprite.png	Background gradient images for the three states of <code>.action.primary</code> buttons	Removed rounded corners. Colors were adjusted.
button_action_sprite.png	Background gradient image for the up, over, and pressed states of action buttons	Removed rounded corners. Colors were adjusted.
button_action_square_icons_sprite.png	Button icons for the <code>.action.square</code> buttons	All icons have new shadow effect.
button_capsule_sprite.png	Background images with rounded corners for capsule buttons	Rounded corners have a smaller radius. Colors adjusted. Buttons are shorter in height.
button_options_sprite.png	Background images for the up, over, and pressed states of options buttons	Removed rounded corners. Added a shadow-only image for the pressed state.
disclosure_indicators_sprite.png	Small icons that disclose functionality in various parts of the application (within buttons, menus, lists, and forms)	Shadow effect added to icons for <code>.button.capsule.indicator</code> .
home_bkgd.png	Large background image on home page	Image refactored.
home_icons_sprite.png	Large icons for <code>.action.jumbo</code> button on home and admin home pages	Icons for View Reports and Create Ad Hoc View refactored.
input_bkgd.png	Top inner shadow background image for text inputs	Shadow lightened. Image height changed from 2px tall to 3px tall.

Table 7-4 Modified Images in 4.7.0 Default Theme

Image	Usage	Notes
login_welcome_bkgd.jpg	Main image on login page of pro edition	Image refactored.
login_welcome_ce_bkgd.jpg	Main image on login page of community edition	Image refactored.
logo.png	Logo used in the banner of the application	Image is smaller.
menu_primaryNav_sprite.png	Sprite containing background gradients for mouse over effects and the home icon used in the main menu	Home icons reduced in size and shadow effect added.
panel_inset_inset_bkgd.png		
search_sprite.png	Icons used in search controls (search and clear search)	Images refactored. Input background removed.
tabs_horizontal_sprite.png	Background gradient images for the over and pressed states of horizontal tabs	Removed rounded corners. Colors adjusted.
tabs_vertical_sprite.png	Background gradient images for the up, over and pressed states of vertical tabs	Removed rounded corners. Colors adjusted.

Table 7-5 lists the images that have not been affected by the refactoring of themes in 4.7.0.

Table 7-5 Images With No Modifications in 4.7.0

Image	Usage
adhoc_datatree_icons_sprite.png	Node and leaf images for fields, measures, and dimensions in the ad hoc data tree
floatingMenu_sprite.png	Popup menu icons for dashboard widgets
grid_20x20_bkgd.png	Grid background for dashboard designer canvas
list_node_animation.gif	Miniature "please wait" animation used in list nodes
list_pressed_bkgd.png	Background image for the top inner shadow that appears for the pressed state of a <code>.list.filter</code> list item
lists_sprite.png	Small icons used in various lists throughout the application. Also used for tokens, menus and OLAP cell sorting
loadinfo.gif	"Please wait" animation used for iPad only
message_icons_sprite.png	Sprite containing a single icon used for alert message
panel_sizer_sprite.png	Sprite containing icons for resizing panels and dialogs
report_load_animation.gif	Animation used in the toolbar of the interactive viewer when a report is loading
sort_indicators_sprite.png	Sprite containing icons used for the up, over, and pressed states of sortable lists (ascending and descending)
tabs_horizontal_buttons_bkgd.png	Image used as the top drop shadow for <code>.tabSet.horizontal.buttons</code>
wait_animation_large.gif	"Please wait" animation used in dialog boxes on page loads and in dashboard iframes when widgets are loading

Table 7-6 shows images used in the earlier themes that have been removed in 4.7.0, as well as images that have been refactored by placing them in the file `inner_pagination_sprite.png`.

Table 7-6 Images from Previous Default Theme Removed in 4.7.0

Image	Usage	Notes
<code>ajax-loader.gif</code>	Animation used in the "please wait" dialog for printing dashboards that contain charts	replaced with <code>wait_animation_large.gif</code>
<code>body_bkgnd.png</code>	Gradient background image used as the body background for all pages	
<code>button_action_jumbo_sprite.png</code>	Background images with rounded corners used for up, over, and pressed states of jumbo buttons	
<code>column_bkgd_corners.png</code>	Background image used for rounded corners with drop shadow on column decorated elements	
<code>column_bkgd_edges_rl.png</code>	Background image used for right and left borders with drop shadow on column decorated elements	
<code>column_bkgd_edges_tb.png</code>	Background image used for top and bottom borders with drop shadow on column decorated elements	
<code>first-d.gif</code>	Disabled state of "go to first" icon used for pagination controls inside report dashboard widgets	Image placed in sprite file <code>inner_pagination_sprite.png</code>
<code>first.gif</code>	"Go to first" icon used for pagination controls inside report dashboard widgets	Image placed in sprite file <code>inner_pagination_sprite.png</code>
<code>frame_background_sprite.png</code>	Gradient background image for the <code>#frame</code> element	
<code>frame_bkgd_corners.png</code>	Background image used for rounded corners with drop shadow on the <code>#frame</code> element	
<code>frame_bkgd_edges_rl.png</code>	Drop shadow image used for right and left edges of the <code>#frame</code> element	
<code>frame_bkgd_edges_tb.png</code>	Drop shadow image used for top and bottom edges of the <code>#frame</code> element	
<code>frame_header_sprite.png</code>	Gradient background image with rounded corners used for the header of the <code>#frame</code> element.	<code>#frame</code> header element holds the main navigation.
<code>last-d.gif</code>	Disabled state of "go to last" icon used for pagination controls inside report dashboard widgets	Image moved to sprite file <code>inner_pagination_sprite.png</code> .
<code>last.gif</code>	"Go to last" icon used for pagination controls inside report dashboard widgets	Image placed in sprite file <code>inner_pagination_sprite.png</code> .
<code>login_welcome_bkgd_ipad.jpg</code>	Main image on login page for iPad	
<code>logo_small.png</code>	Logo used in the frame header/main navigation bar for iPad	

Table 7-6 Images from Previous Default Theme Removed in 4.7.0

Image	Usage	Notes
menu_context_bkgd_corners.png	Background image used for rounded corners with drop shadow on <code>.menu.context</code> elements	
menu_context_bkgd_edges_rl.png	Background image used for right and left borders with drop shadow on <code>.menu.context</code> elements	
menu_context_bkgd_edges_tb.png	Background image used for top and bottom borders with drop shadow on <code>.menu.context</code> elements	
menu_dropDown_bkgd_corners.png	Background image used for rounded corners with drop shadow on <code>.menu.dropDown</code> elements	
menu_dropDown_bkgd_edges_rl.png	Background image used for right and left borders with drop shadow on <code>.menu.dropDown</code> elements	
menu_dropDown_bkgd_edges_tb.png	Background image used for top and bottom borders with drop shadow on <code>.menu.dropDown</code> elements	
menu_vertical_bkgd.png	Background for <code>.menu.vertical</code>	
next-d.gif	Disabled state of next icon used for pagination controls inside report dashboard widgets	Image placed in sprite file <code>inner_pagination_sprite.png</code> .
next.gif	Next icon used for pagination controls inside report dashboard widgets	Image placed in sprite file <code>inner_pagination_sprite.png</code> .
panel_bkgd_corners.png	Background image of rounded corners with drop shadow used for the <code>.dialog.overlay</code> element	
panel_bkgd_edges_rl.png	Background image of right and left borders with drop shadow used for the <code>.dialog.overlay</code> element	
panel_bkgd_edges_tb.png	Background image of top and bottom borders with drop shadow used for the <code>.dialog.overlay</code> element	
panel_dialog_header_sprite.png	Background for dialog headers	
panel_info_bkgd_corners.png	Background image of rounded corners with drop shadow used for the <code>.info</code> element	
panel_info_bkgd_edges_rl.png	Background image of right and left borders with drop shadow used for the <code>.info</code> element	
panel_info_bkgd_edges_tb.png	Background image of top and bottom borders with drop shadow used for the <code>.info</code> element	
panel_info_system_bkgd_corners.png	Background image of rounded corners used for the <code>.info.system</code> element	

Table 7-6 Images from Previous Default Theme Removed in 4.7.0

Image	Usage	Notes
panel_info_system_bkgd_edges_rl.png	Background image of right and left borders used for the <code>.info.system</code> element	
panel_info_system_bkgd_edges_tb.png	Background image of top and bottom borders used for the <code>.info.system</code> element	
panel_info_tooltip_bkgd_corners.png	Background image of rounded corners used for the <code>.panel.tooltip.info</code> element	
panel_info_tooltip_bkgd_edges_rl.png	Background image of right and left borders used for the <code>.panel.tooltip.info</code> element	
panel_info_tooltip_bkgd_edges_tb.png	Background image of top and bottom borders used for the <code>.panel.tooltip.info</code> element	
panel_inlay_bkgd_corners.png	Background image of rounded corners with drop shadow used for the <code>.dialog.inlay</code> element	
panel_inlay_bkgd_edges_rl.png	Background image of left and right borders with drop shadow used for the <code>.dialog.inlay</code> element	
panel_inlay_bkgd_edges_tb.png	Background image of top and bottom borders with drop shadow used for the <code>.dialog.inlay</code> element	
panel_inlay_gradient_left.png	Left background image with gradient and rounded corners used for the <code>.dialog.inlay .footer</code> element	
panel_inlay_gradient_right.png	Right background image with gradient and rounded corners used for the <code>.dialog.inlay .footer</code> element	
panel_inset_bkgd_corners.png	Background image of rounded corners with drop shadow used for the <code>.groupBox</code> element	
panel_inset_bkgd_edges_rl.png	Background image of left and right borders with drop shadow used for the <code>.groupBox</code> element	
panel_inset_bkgd_edges_tb.png	Background image of top and bottom borders with drop shadow used for the <code>.groupBox</code> element	
panel_widget_header_sprite.png	Background gradient with rounded corners used for the headers of the <code>.panel.widget</code> and <code>.dialog.overlay.widget</code> elements	
prev-d.gif	Disabled state of Previous icon used for pagination controls inside report dashboard widgets	Image placed in sprite file <code>inner_pagination_sprite.png</code>

Table 7-6 Images from Previous Default Theme Removed in 4.7.0

Image	Usage	Notes
prev.gif	Previous icon used for pagination controls inside report dashboard widgets	Image placed in sprite file inner_pagination_sprite.png
toolbar_bkgd.png	Background gradient image for the .toolbar element	

7.2 Data Snapshots

Reports in JasperReports Server can now store snapshots of the report data as part of the report unit in the repository. Data snapshots load quickly without querying the data source, thus increasing performance and reducing database load. Data snapshots can be refreshed on-demand, by scheduling, or by setting server-wide or report-specific policies. Existing installations should also resolve any upgrade issues before enabling data snapshots.

Data snapshots are stored in the JasperReports Server repository, which may significantly increase the size of the repository. Before enabling data snapshots, you may want to analyze the effect of snapshots on your repository size and upgrade your storage. For most installations, the benefits from reducing the overall number of queries should outweigh the cost of additional storage.

The actual effect on your repository size depends on the nature of your reports and how many snapshots you enable. It is difficult to give a standard estimate, because snapshot size depends on factors such as the number of columns and the complexity of the report. For a given report, the overall snapshot size is usually proportional to the number of rows. For example, if you have a report of 100,000 records, with 20 records per page, and the snapshot size for one page is 2KB, the overall size of the snapshot will be $(100,000 \div 20) \times 2$ KB, or 10 MB.

Data snapshots are turned off globally by default in 4.7.0. If you enable data snapshots, they can be turned on or off for each individual report.

See the *JasperReports Server Administrator Guide* for more information about enabling data snapshots in your JasperReports Server deployment. See the *JasperReports Server User Guide* for more information about using data snapshots with individual reports.

7.2.1 Using Data Snapshots

Enabling data snapshots has the following benefits:

- Queries are only run when a new data snapshot is enabled, when the snapshot is refreshed, and automatically when the server detects that the snapshot is out of date. This allows multiple users to look at the same report without running multiple queries.
- Sharing data snapshots means everyone sees identical data. A data snapshot is a static entity that reflects the state of the data at the time the snapshot was created. This eliminates differences sometimes experienced when data has changed in the repository, or queries are run in different time zones.
- Users can refresh a snapshot and view the latest data by clicking a **Refresh** button. Non-administrative users do not have permissions to overwrite the snapshot after refresh.
- You can turn off snapshots for reports that need to be dynamic, such as real-time reports in dashboards.

CHAPTER 8 UPGRADING FROM 3.7 - 4.5 TO 4.7.0

This chapter describes the procedure for upgrading from JasperReports Server 3.7.0, 3.7.1, 4.0.0, 4.1.0, 4.2.1, or 4.5.0 to JasperReports Server 4.7.0. The upgrade procedures described in this chapter use the JasperReports Server WAR File Distribution ZIP release package and the included buildomatic scripts.

The procedure in this chapter can be used with JasperReports Server 4.5.0. However, we recommend you use the procedure in [Chapter 9, “Upgrading from 4.5.0 to 4.7.0,” on page 73](#).

The examples in this chapter use JasperReports Server 4.2.1.

This chapter contains the following sections:

- ♦ [Standard Upgrade Steps](#)
- ♦ [Backing Up Your Previous JasperServer Instance](#)
- ♦ [Exporting Old Repository Data](#)
- ♦ [Preparing the JasperReports Server 4.7.0 WAR File Distribution](#)
- ♦ [Configuring Buildomatic for Your Database and Application Server](#)
- ♦ [Upgrading to JasperReports Server 4.7.0](#)
- ♦ [Starting and Logging into JasperReports Server 4.7.0](#)
- ♦ [Additional Tasks to Complete the Upgrade](#)
- ♦ [Manual Upgrade Steps](#)

8.1 Standard Upgrade Steps

This section lists the standard upgrade steps. In general, perform these steps when upgrading to any new JasperReports Server release.

1. Plan your upgrade and evaluate how changes in JasperServer 4.7.0 impact your existing deployment.
2. Back up your old JasperReports Server instance. For example, back up your JasperReports Server 4.2.1.
3. Export your existing repository data. For example, export your 4.2.1 data.
4. Download and set up the 4.7.0 JasperReports Server WAR file distribution zip.
5. Run the js-upgrade script as described in section [8.6, “Upgrading to JasperReports Server 4.7.0,” on page 68](#).

If your previous instance of JasperReports Server has any custom modifications or extensions, keep track of these and re-integrate them into your 4.7.0 instance after upgrading.

8.2 Backing Up Your Previous JasperServer Instance

First, back up your old JasperServer WAR file and jasperserver database. For example, back up your JasperReports Server 4.2.1 WAR file and 4.2.1 jasperserver database, so you can restore them in the event of an upgrade problem. Perform these steps from the command line in a Windows or Linux shell.

The following instructions are for the PostgreSQL and MySQL databases in Tomcat.

1. Back up the jasperserver directory in Tomcat to a backup directory. For example:

```
cd <tomcat>
```

Windows: `rename <tomcat>/webapps/jasperserver <tomcat>/js-4.2.1-war-backup`

Linux: `mv <tomcat>/webapps/jasperserver <tomcat>/js-4.2.1-war-backup`

2. Back up the jasperserver database. Go to the location where you originally unpacked WAR file distribution zip for the previous version or installed the previous version from the JasperReports Server installer:

- a. Navigate to the location of your previous, unpacked WAR file distribution.

```
cd <js-install-4.2.1>
```

- b. Back up the MySQL or PostgreSQL database using one of the following commands:

- PostgreSQL

```
cd <js-install-4.2.1>
```

```
pg_dump --username=postgres jasperserver > js-db-4.2.1-dump.sql
```

- MySQL

Windows: `mysqldump --user=root --password=<password> jasperserver > js-db-4.2.1-dump.sql`

Linux: `mysqldump --user=root --password=<password> --host=127.0.0.1 jasperserver > js-db-4.2.1-dump.sql`

If you installed the previous release from the installer, specify `--user=jasperdb` in this command.

If you receive a packet size error, see section [A.7.3, “Maximum Packet Size in MySQL,” on page 96](#).



Jaspersoft has tested the `mysqldump` utility for backing up and restoring MySQL databases, but there are other MySQL backup mechanisms, some of which may work better for your JasperReports Server installation.

8.3 Exporting Old Repository Data

You need to export your old repository data, for example your 4.2.1 repository data, using the JasperReports Server export utility. There are two ways to export:

- Use the buildomatic scripts (if you originally installed using buildomatic).
- Use the `js-export.bat/sh` script found in the `<js-install>/buildomatic` directory.

8.3.1 Using Buildomatic Scripts to Export Data

If you configured buildomatic and your `default_master.properties` file as described in section [12.3.1, “Creating a default_master.properties File,” on page 88](#), you can export your old repository data. For example, to export 4.2.1 repository data, use the following commands:

1. Navigate to the buildomatic directory:

```
cd <js-install-4.2.1>/buildomatic
```

2. Run buildomatic with the export target:

Windows: `js-ant.bat export-everything -DexportFile=js-4.2.1-export.zip`

Linux: `./js-ant export-everything -DexportFile=js-4.2.1-export.zip`

This operation uses the export option `--everything` which exports all your repository data.



Note the location of this export file so that you can use it during the 4.7.0 upgrade process.

8.3.2 Using the js-export Script to Export Data

To use the `js-export.bat/.sh` script, navigate to the `builddomatic` folder of your previous release, for example, the `<js-install-4.2.1>/builddomatic` folder. If you are using the PostgreSQL database then the `js-export` script should already be configured to run. If you are using a different database, or you have changed database passwords, you need to update the `js-export` configuration.

The `import-export` utility for JasperServer 3.7.0 needs additional configuration. See section [12.6, “Configuring the Import-Export Utility for JasperServer 3.7,”](#) on page 90 for more information.

Run the following commands:

1. Navigate to the `builddomatic` directory:

```
cd <js-install-4.2.1>/builddomatic
```

2. Run the `js-export` script:

```
Windows: js-export-ce.bat --everything --output-zip js-4.2.1-export.zip
```

```
Linux: js-export-ce.sh --everything --output-zip js-4.2.1-export.zip
```

This operation uses the export option `--everything` which exports all your repository data.



Note the location of this export file so that you can use it during the 4.7.0 upgrade process.

8.4 Preparing the JasperReports Server 4.7.0 WAR File Distribution

Use the `builddomatic` `js-upgrade` scripts included in the 4.7.0 WAR file distribution ZIP release package to carry out the upgrade. Follow these steps to obtain and unpack the WAR file distribution ZIP file:

1. The WAR file distribution comes in a file named `jasperreports-server-cp-4.7.0-bin.zip` in the compressed ZIP format. Download the WAR file distribution from [Jaspersoft technical support](http://support.jaspersoft.com) (<http://support.jaspersoft.com>) or contact your sales representative.
2. Extract all files from `jasperreports-server-cp-4.7.0-bin.zip`. Choose a destination, such as a `C:\Jaspersoft` folder on Windows, `/home/<user>` on Linux, or `/Applications` on Mac OSX.

This document refers to the location where you unpacked the 4.7.0 WAR File Distribution ZIP as:

```
<js-install-4.7.0>
```

8.5 Configuring Builddomatic for Your Database and Application Server

The upgrade procedure in this chapter uses the `js-upgrade-newdb-ce` shell script.



For Unix, the bash shell is required for the `js-upgrade` scripts. If you are installing to a non-Linux Unix platform such as HP-UX, IBM AIX, FreeBSD or Solaris, you need to download and install the bash shell. See [“Bash Shell for Solaris, IBM AIX, HP UX and FreeBSD”](#) on page 93 for more information.

8.5.1 Example Buildomatic Configuration

The upgrade configuration is handled by the `default_master.properties` file. Jaspersoft provides a sample configuration file for each database, for example, `postgresql_master.properties`. You must specify your database credentials and your application server location, and rename the file to `default_master.properties`.

8.5.1.1 PostgreSQL Example

This section shows how to configure `default_master.properties` for PostgreSQL.

1. Locate the `postgresql_master.properties` sample configuration file:

Database	Master Properties File
PostgreSQL	<js-install-4.7.0>/buildomatic/sample_conf/postgresql_master.properties

2. Copy the file to <js-install-4.7.0>/buildomatic.
3. Rename the file `default_master.properties`.
4. Edit `default_master.properties` for your database and application server:

Database	Sample Property Values
PostgreSQL	<pre>appServerType=tomcat6 [tomcat7, tomcat5, jboss, glassfish2, glassfish3] appServerDir=c:\Apache Software Foundation\Tomcat 6 dbUsername=postgres dbPassword=postgres dbHost=localhost</pre>

8.5.1.2 MySQL Example

This section shows how to configure `default_master.properties` for MySQL.

1. Locate the `mysql_master.properties` sample configuration file:

Database	Master Properties File
MySQL	<js-install-4.7.0>/buildomatic/sample_conf/mysql_master.properties

2. Copy the file to <js-install-4.7.0>/buildomatic.
3. Rename the file `default_master.properties`.
4. Edit `default_master.properties` for your database and application server:

Database	Sample Property Values
MySQL	<pre>appServerType=tomcat6 [tomcat7, tomcat5, jboss, glassfish2, glassfish3] appServerDir=C:\Apache Software Foundation\Tomcat 6 dbUsername=root dbPassword=password dbHost=localhost</pre>

8.6 Upgrading to JasperReports Server 4.7.0

Once your buildomatic scripts have been configured, you can complete the upgrade.



- Make sure you have backed up your `jasperserver` database before proceeding.
- Make sure you have backed up your old JasperReports Server WAR file before proceeding.

1. Stop your application server.
2. Start your database server.
3. Run the following commands:

Commands	Description
<code>cd <js-install-4.7.0>/buildomatic</code>	Change to buildomatic directory
<code>js-upgrade-newdb-ce.bat <path>/js-4.2.1-export.zip</code>	(Windows) Upgrade jasperserver war file, drop and recreate the database, import data file from previous version.
<code>./js-upgrade-newdb-ce.sh <path>/js-4.2.1-export.zip</code>	(Linux) Upgrade jasperserver war file, drop and recreate the database, import data file from previous version.



On MySQL, if you receive an error about packet size, see section [A.7.3, “Maximum Packet Size in MySQL,”](#) on page 96.



If you have auditing enabled, see section [A.10.1, “Include Audit Events on Upgrade,”](#) on page 103 for information on including auditing events.

8.6.1 js-upgrade Test Mode

You can run the js-upgrade script in test mode using the `test` option. For example, on Window, enter:

```
cd <js-install-4.7.0>/buildomatic
js-upgrade-newdb-ce.bat test <path>/js-4.2.1-export.zip
```

In test mode, the js-upgrade scripts check your `default_master.properties` settings. The application server location and the capability to connect to the specified database are validated. Using `test` mode can help debug issues such as an incorrect database password. Your system will not be altered when executing the script in `test` mode.

8.6.2 Output Log Location

The js-upgrade script creates an output log that captures standard output and error output. If there are any problems during the execution of the script, or if you want to remember which options you chose, you can open the output log file.

The output log file is located here:

```
<js-install>/buildomatic/logs/js-upgrade-ce-<date>-<number>.log
```

8.6.3 Errors

If you encounter errors during the js-upgrade script execution, first look at the output log to see if you can spot any errors. Also, refer to the Troubleshooting section [Appendix A, “Troubleshooting,”](#) on page 91. The information in this section applies to js-upgrade scripts as well as js-install scripts.

If you need to modify values in your `default_master.properties` file, you can simply edit the file. When you run the js-upgrade script again, the new values are used.

8.7 Starting and Logging into JasperReports Server 4.7.0

Start your Tomcat, JBoss, or GlassFish application server. Your database should already be running.

8.7.1 Clearing Your Browser Cache

Before you log in, make sure you and your end users clear the browser cache. JavaScript files, which enable UI elements of JasperReports Server, are typically cached by the browser. Clear the cache to ensure that the newer files are used.

8.7.2 Logging into JasperReports Server

Log in to your updated JasperReports Server instance using the following URL, user IDs, and passwords:

URL: `http://localhost:8080/jasperserver`

User ID	Password	Description
jasperadmin	<your-password>	Administrator



Your jasperadmin password might be reset to the default setting by the upgrade operation. For example, the jasperadmin password might be reset to `jasperadmin`. For security reasons, you should change your jasperadmin password to a non-default value.

Your JasperReports Server instance has now been upgraded to 4.7.0. In the event of startup or login problems, refer to [Appendix A, “Troubleshooting,” on page 91](#).

8.8 Additional Tasks to Complete the Upgrade

8.8.1 Handling JasperReports Server Customizations

If you made modifications or customizations to the old JasperReports Server application, JasperReports Server 4.2.1 for example, these configurations are typically found in the `WEB-INF/applicationContext-*.xml` set of files.

Configuration modifications, such as client-specific security classes or LDAP server configurations, need to be hand-copied from your previous environment and re-integrated into the new 4.7.0 environment.

8.8.2 Clearing the Application Server Work Folder

Application servers have work folders where JasperReports Server files are compiled and cached and other objects are stored. When you update the WAR file, the `buildomatic deploy-webapp-ce` target should automatically clear the application server's work directory, but it's a good practice to double-check. A permission problem, or some other problem, could prevent the clearing of the work folder.

To clear the work folder in Tomcat:

1. Change directory to `<tomcat>/work`.
2. Delete all the files and folders in this directory.

8.8.3 Clearing the Application Server Temp Folder

JasperReports Server uses caching to speed operations within the application. Caching files are created and stored in the application server to support this functionality. Typically, these cached files are stored in a `temp` folder. Clear this `temp` folder to avoid any conflicts after the upgrade is complete. Typically, the `temp` folder used by an application server corresponds to the path pointed at by the `java.io.tmpdir` Java system property. For Apache Tomcat the `temp` folder is `<tomcat>/temp`.

To clear the temp folder in Apache Tomcat:

1. Change directory to `<tomcat>/temp`
2. Delete all the files and folders in this directory

8.8.4 Clearing the Repository Cache Database Table

In the `jasperserver` database, compiled JasperReports Library resources are cached in the `JIREpositoryCache` table for increased efficiency at runtime. In some cases, you may encounter errors running reports after an upgrade. Because the JasperReports Library JAR is typically updated with each new JasperReports Server release, old cached items can get out of date and thus cause errors at runtime. If you encounter errors that mention a JasperReports Library “local class incompatible,” check your repository cache table. You can clear your `jasperserver` database cache table whether there are errors or not as part of this upgrade process.

To manually clear the repository cache database table, run a SQL command similar to one shown in this example:

```
update JIREpositoryCache set item_reference = null;
delete from JIREpositoryCache;
```

8.8.5 Updating the XML/A Connections (Optional)

When you upgrade your sample data to 4.7.0, your XML/A connection sample data is updated. XML/A connections use JasperReports Server login accounts for authentication. Because of this, and because you would normally modify your default `jasperadmin` password as a standard security procedure, your sample XML/A connection may fail due to a mismatched password.

To update your sample XML/A connections, refer to section [6.5, “Updating XML/A Connection Definitions,”](#) on page 54.

8.8.6 Upgrading the Liferay Portal

JasperReports Server can be configured to run with the Liferay Portal. If your JasperReports Server is set up to run with Liferay, you must do the following as part of the upgrade process.

1. Delete the `webapps/Jaspersoft` folder of the application server hosting Liferay. This deletes libraries used in older versions that conflict with libraries in the latest version.
2. Deploy the new portlet WAR.

For more information, refer to the *JasperReports Server Administrator Guide*.

8.9 Manual Upgrade Steps

This section has the older, manual upgrade steps that were in place before the `js-upgrade` shell scripts were implemented in JasperReports Server release 4.0.0. These are provided in the following table as a reference, mainly for internal use. The `js-upgrade` shell scripts execute these buildomatic targets “behind the scenes.” We recommend using the `js-upgrade` scripts described in the beginning of this upgrade chapter instead of these manual steps.

Older buildomatic upgrade steps for this chapter are the following (using a 4.2.1 to 4.7.0 upgrade as an example):

Commands	Description
<code>cd <js-install-4.7.0>/buildomatic</code>	
<code>js-ant drop-js-db</code> <code>js-ant create-js-db</code> <code>js-ant init-js-db-ce</code>	This will delete and recreate your jasperserver db. Make sure your original database is backed up.
<code>js-ant import-minimal-ce</code>	
<code>js-ant import-upgrade</code> <code>-DimportFile="<path-and-filename>"</code>	The <code>-DimportFile</code> should point to the <code><path></code> and <code><filename></code> of the <code>js-4.2.1-export.zip</code> file you created earlier. On Windows, you must use double quotation marks (") if your path or filename contains spaces. On Linux, you must use double quotation marks, escaped with a backslash (\") in this case.

Commands	Description
<code>js-ant import-sample-data-upgrade-ce</code>	(Optional) This step is optional; it loads the 4.7.0 sample data. The old sample data is overwritten, so you may need to redo certain changes such as configuring the sample data sources for your database.
<code>js-ant deploy-webapp-ce</code>	Delete the existing older war file, deploy the 4.7.0 war file.



On MySQL, if you receive an error about packet size, see section [A.7.3, “Maximum Packet Size in MySQL,”](#) on page 96.

CHAPTER 9 UPGRADING FROM 4.5.0 TO 4.7.0

This chapter describes how to upgrade from JasperReports Server 4.5.0 to JasperReports Server 4.7.0.

The examples describe how to use the `js-upgrade` shell scripts to carry out the upgrade from JasperReports Server 4.5.0 to 4.7.0. The steps are similar to the `js-upgrade` steps used in the previous chapter except the `jasperserver` database is not dropped and recreated. Instead, a SQL script upgrades the database to a 4.7.0 database.

This chapter contains the following sections:

- [Upgrade Steps Overview](#)
- [Backing Up Your Old JasperReports Server Instance](#)
- [Preparing the JasperReports Server 4.7 WAR File Distribution](#)
- [Configuring Buildomatic for Your Database and Application Server](#)
- [Upgrading to JasperReports Server 4.7.0](#)
- [Starting and Logging into JasperReports Server 4.7.0](#)
- [Additional Tasks to Complete the Upgrade](#)
- [Manual Upgrade Steps](#)

9.1 Upgrade Steps Overview

These are the general steps used in this section.

1. Plan your upgrade and evaluate how changes in JasperServer 4.7.0 impact your existing deployment.
2. Back up your old `jasperserver` database instance.
3. Download and set up the new 4.7.0 JasperReports Server WAR file distribution zip.
4. Run the `js-upgrade` script as described in [9.5, “Upgrading to JasperReports Server 4.7.0,” on page 75](#).

If your old instance of JasperServer has any custom modifications or extensions, keep track of these and re-integrate them into your 4.7.0 instance after the upgrade is complete.

9.2 Backing Up Your Old JasperReports Server Instance

First you must backup your JasperReports Server WAR file and `jasperserver` database so that they can be restored in case there is a problem with the upgrade. These steps are performed from the command line in a Windows or Linux shell.

9.2.1 Backing Up Your JasperReports Server WAR File

Back up the `jasperserver` directory in Tomcat to a backup directory.

1. Go to the `<tomcat>` directory.
2. Make a new backup directory, named `js-4.5.0-war-backup`, for example.
3. Copy `<tomcat>/webapps/jasperserver` to `<tomcat>/js-4.5.0-war-backup`.
4. Delete the `<tomcat>/webapps/jasperserver` directory.

9.2.2 Backing Up Your JasperServer Database

Go to the location where you originally unpacked the WAR file distribution zip of the old release that you're upgrading, `<js-install-4.5.0>`, for example.

1. Go to `<js-install-4.5.0>` (the location of your original unpacked 4.5 WAR file distribution).
2. Run the following commands for either PostgreSQL or MySQL:

- PostgreSQL

```
cd <js-install-4.5.0>
pg_dump --username=postgres jasperserver > js-db-4.5.0-dump.sql
```

- MySQL

```
Windows: mysqldump --user=root --password=<password> jasperserver > js-db-4.5.0-dump.sql
```

```
Linux: mysqldump --user=root --password=<password> --host=127.0.0.1 jasperserver >
js-db-4.5.0-dump.sql
```



If you receive an error about packet size, see section [A.7.3, “Maximum Packet Size in MySQL,”](#) on page 96.

9.3 Preparing the JasperReports Server 4.7 WAR File Distribution

Use the buildomatic `js-upgrade` scripts included in the 4.7.0 WAR file distribution ZIP release package to carry out the upgrade. Follow these steps to obtain and unpack the WAR file distribution ZIP file:

1. The WAR file distribution comes in a file named `jasperreports-server-cp-4.7.0-bin.zip` in the compressed ZIP format. Download the WAR file distribution from [Jaspersoft technical support](http://support.jaspersoft.com) (<http://support.jaspersoft.com>) or contact your sales representative.
2. Extract all files from `jasperreports-server-cp-4.7.0-bin.zip`. Choose a destination, such as the `C:\Jaspersoft` folder on Windows, `/home/<user>` on Linux, or `/Applications` on Mac OSX.

After you unpack the WAR File Distribution Zip, the resulting location will be known as:

```
<js-install-4.7.0>
```

9.4 Configuring Buildomatic for Your Database and Application Server

This upgrade procedure uses the `js-upgrade-samedb` shell scripts.



For Unix, the bash shell is required for the `js-upgrade` scripts. If you are installing to a non-Linux Unix platform such as HP-UX, IBM AIX, FreeBSD or Solaris, you need to download and install the bash shell. See “[Bash Shell for Solaris, IBM AIX, HP UX and FreeBSD](#)” on page 93 for more information.

The following section contains example configurations using the PostgreSQL and MySQL databases.

9.4.1 Example Buildomatic Configuration

The upgrade configuration is handled by the `default_master.properties` file. Jaspersoft provides a sample configuration file for each database. You must specify your database credentials and your application server location, and rename the file to `default_master.properties`.

9.4.1.1 PostgreSQL Example

This example uses PostgreSQL.

1. Locate the `postgresql_master.properties` sample configuration file:

Database	Master Properties File
PostgreSQL	<js-install-4.7.0>/buildomatic/sample_conf/postgresql_master.properties

2. Copy the file to <js-install-4.7.0>/buildomatic.
3. Rename the file to `default_master.properties`.
4. Edit `default_master.properties` for your database and application server:

Database	Sample Property Values
PostgreSQL	<pre>appServerType=tomcat6 [tomcat7, tomcat6, jboss, glassfish2, glassfish3] appServerDir=c:\\apache-tomcat-6.0.26 (for example) dbUsername=postgres dbPassword=postgres dbHost=localhost</pre>

9.4.1.2 MySQL Example

This example uses MySQL.

1. Locate the `mysql_master.properties` sample configuration file:

Database	Master Properties File
MySQL	<js-install-4.7.0>/buildomatic/sample_conf/mysql_master.properties

2. Copy the file to <js-install-4.7.0>/buildomatic.
3. Rename the file to `default_master.properties`.
4. Edit `default_master.properties` for your database and application server:

Database	Sample Property Values
MySQL	<pre>appServerType=tomcat6 [tomcat7, tomcat6, jboss, glassfish2, glassfish3] appServerDir=c:\\Apache Software Foundation\\Tomcat-6 (for example) dbUsername=root dbPassword=password dbHost=localhost</pre>

9.5 Upgrading to JasperReports Server 4.7.0

Now that your buildomatic scripts have been configured, you can complete the upgrade.



- Make sure you have backed up your `jasperserver` database before proceeding.
- Make sure you have backed up your old JasperReports Server WAR file before proceeding.

1. Stop your application server.
2. Start your database server.
3. Run these commands to upgrade JasperReports Server 4.5.0 to JasperReports Server 4.7.0:

Commands	Description
<code>cd <js-install-4.7.0>/buildomatic</code>	
<code>js-upgrade-samedb-ce.bat</code>	(Windows) Upgrade jasperserver war file, upgrade jasperserver database to 4.7.0, add 4.7.0 repository resources into the database
<code>./js-upgrade-samedb-ce.sh</code>	(Linux) Upgrade jasperserver war file, upgrade jasperserver database to 4.7.0, add 4.7.0 repository resources into the database

9.5.1 js-upgrade Test

You can run the js-upgrade scripts in test mode using the `test` option. For example, in Windows enter:

```
cd <js-install-4.7.0>/buildomatic
js-upgrade-samedb-ce.bat test
```

In `test` mode, the js-upgrade scripts check your `default_master.properties` settings. The application server location is validated and the capability to connect to the specified database is validated. Using `test` mode can help debug issues, such as an incorrect database password. Your system is not altered when executing the script in `test` mode.

9.5.2 Output Log Location

The js-upgrade script creates an output log that captures standard output and error output. If there are any problems during the execution of the script or if you want to remember which options you chose, you can open the output log file.

The output log file is located here:

```
<js-install>/buildomatic/logs/js-upgrade-<date>-<number>.log
```

9.5.3 Errors

If you encounter errors during the js-upgrade script execution, look at the output log to see if you can spot any errors. Additionally, you should refer to [Appendix A, “Troubleshooting,” on page 91](#). The information in this section applies to both js-upgrade scripts and the js-install scripts.

If you need to modify values in your `default_master.properties` file, you can simply edit the file. When the js-upgrade script is run again, the new values will be used.

9.6 Starting and Logging into JasperReports Server 4.7.0

Start your Tomcat, JBoss, or GlassFish application server. Your database should already be running.

9.6.1 Clearing Your Browser Cache

Before you log in, make sure you and your end-users clear the browser cache. JavaScript files, which enable UI elements of JasperReports Server, are typically cached by the browser. Clear the cache to ensure that the newer files are used.

9.6.2 Logging into JasperReports Server

Log in using the following URL, user IDs, and passwords:

URL: `http://localhost:8080/jasperserver`

User ID	Password	Description
jasperadmin	<your-password>	Administrator

Your JasperReports Server instance has now been upgraded to 4.7.0. In the event of startup or login problems, refer to [Appendix A, “Troubleshooting,” on page 91](#)

9.7 Additional Tasks to Complete the Upgrade

For additional information about optional steps after you complete the main upgrade steps are complete, see section [8.8, “Additional Tasks to Complete the Upgrade,” on page 70](#).

9.7.1 Clearing the Application Server Temp Folder

JasperReports Server uses caching to speed operations within the application. In the application server, caching files are created and stored for this caching functionality. Typically, these cached files are stored in a `temp` folder. Clear this `temp` folder to avoid any conflicts after the upgrade is complete. For Apache Tomcat the `temp` folder is `<tomcat>/temp`. (In general, the `temp` folder used by an Application Server corresponds to the path pointed at by the `java.io.tmpdir` Java system property.)

To clear the temp folder in Apache Tomcat:

1. Change directory to `<tomcat>/temp`

Delete all the files and folders in this directory

9.7.2 Clearing the Repository Cache Database Table

In the `jasperserver` database, compiled JasperReports Library resources are cached in the `JIREpositoryCache` table for increased efficiency at runtime. In some cases, you may encounter errors running reports after an upgrade. Because the JasperReports Library JAR is typically updated with each new JasperReports Server release, old cached items can get out of date and thus cause errors at runtime. If you encounter errors that mention a JasperReports Library “local class incompatible,” check your repository cache table. In summary, you can clear your `jasperserver` database cache table whether there are errors or not as part of this upgrade process.

To manually clear the repository cache database table, run a SQL command similar to one shown in this example:

```
update JIREpositoryCache set item_reference = null;
delete from JIREpositoryCache;
```

9.8 Manual Upgrade Steps

This section has the older, manual upgrade steps that were in place before the `js-upgrade` shell scripts were implemented in JasperReports Server release 4.0.0. These are provided in the following table as a reference, mainly for internal use. The `js-upgrade` shell scripts execute these buildomatic targets “behind the scenes.” We recommend using the `js-upgrade` scripts described in the beginning of this upgrade chapter instead of these manual steps.

Older buildomatic upgrade steps for this chapter are the following:

Commands	Description
<code>cd <js-install-4.7.0>/buildomatic</code>	
<code>js-ant upgrade-4.5-4.7-ce</code>	Execute SQL script for database upgrade to 4.7.0. Executes script <code>buildomatic/install_resources/sql/<dbType>/upgrade-<dbType>-4.5.0-4.7.0-ce.sql</code>
<code>js-ant import-minimal-for-upgrade-ce</code>	Loads themes and other core resources for 4.7.0.
<code>js-ant import-sample-data-upgrade-ce</code>	(Optional) This step is optional. Loads the 4.7.0 sample data.
<code>js-ant deploy-webapp-ce</code>	Delete old 4.5.0 war file, deploy the 4.7.0 war file.

CHAPTER 10 UPGRADING JASPERSERVER 3.5.0

If you are running JasperServer version 3.5.0, you must upgrade to 3.7.0 before you can upgrade to 4.7.0. Upgrading directly from JasperServer 3.5.0 to 4.7.0 is not a certified (fully Jaspersoft-tested) procedure. To upgrade to 3.7.0, download the WAR File Distribution zip package for 3.7.0 (jasperserver-ce-3.7.0-bin.zip).

The steps to perform this upgrade are documented in the *JasperServer Installation Guide* for the 3.7.0 release. Download the JasperServer 3.7.0 WAR file distribution zip package to get the relevant files and documentation. The Installation Guide is in the docs folder.

Download the JasperServer 3.7.0 WAR file distribution zip package from [Jaspersoft technical support](http://support.jaspersoft.com) (http://support.jaspersoft.com) or contact your sales representative.

If you are running a JasperServer version earlier than 3.5.0, first upgrade to 3.7.0, then to 4.7.0.

CHAPTER 11 CHANGING PASSWORD ENCRYPTION IN JASPERREPORTS SERVER

By default, password encryption is enabled in JasperReports Server and passwords are stored as cipher text in the database. System administrators can change the encryption algorithm and specify the salt key used to initiate the encryption algorithm.

This chapter describes the procedure to enable password encryption if you have a JasperReports Server instance without encryption turned on. For more information about encryption options, see the *JasperReports Server Administrator Guide*.

This chapter contains the following sections:

- ♦ [Backing Up Your JasperReports Server Database](#)
- ♦ [Stopping Your Application Server](#)
- ♦ [Running the Repository Export Utility](#)
- ♦ [Specifying Encryption Settings in the JasperReports Server WAR](#)
- ♦ [Specifying Encryption Settings for the Import Utility](#)
- ♦ [Recreating the JasperReports Server Database](#)
- ♦ [Importing Your Repository Data](#)
- ♦ [Starting the Application Server](#)
- ♦ [Logging into JasperReports Server](#)

11.1 Backing Up Your JasperReports Server Database

As a precaution, you must back up your `jasperserver` database in the event of a problem while enabling encryption.

To back up the default PostgreSQL database, go to the `<js-install>` directory and run the following command:

```
pg_dump -U postgres jasperserver > js-backup.sql
```

To back up a MySQL database, refer to your database product documentation.

11.2 Stopping Your Application Server

You can now stop your application server. You should leave your database running.

11.3 Running the Repository Export Utility

The repository export utility writes the JasperReports Server repository objects to a set of XML and binary files. The output of the export operation is known as an export catalog.

To create the export catalog, go to the `<js-install>/buildomatic` directory and run the following commands. Note that there are two dashes (`--`) in front of the command options:

Windows: `js-export-ce.bat --everything --output-dir js-backup-catalog`

Linux: `js-export-ce.sh --everything --output-dir js-backup-catalog`

For information about running the export utility, refer to [Chapter 12, “Configuring the Import-Export Utilities,” on page 87](#).

11.4 Specifying Encryption Settings in the JasperReports Server WAR

JasperReports Server uses the Spring configuration and security to enable and configure encryption. Use these options to set up strong encryption. This section is focused on the minimal configuration for enabling encryption.

1. Open the following file for editing:

```
<tomcat/webapps/jasperserver/WEB-INF/ApplicationContext-security.xml
```

2. In the definition of the `daoAuthenticationProvider` bean, there is a commented-out reference to the `passwordEncoder` bean. Look for the section of the XML file that starts with:

```
<bean id="daoAuthenticationProvider"
```

In this bean definition, uncomment the reference to `passwordEncoder`. This causes the `passwordEncoder` logic to be used. After removing the commenting characters the line should look like this:

```
<property name="passwordEncoder"><ref local="passwordEncoder"/></property>
```

3. Enable encryption in the `passwordEncoder` bean by modifying the `allowEncoding` property. Change the value from `false` to `true` so that it looks like this:

```
<property name="allowEncoding"><value>true</value></property>
```

4. If the default DESede algorithm is used, the `secretKey` represents the salt key and must be 24 characters. By default, the `keyInPlainText` property is `true`, meaning the key can be in plain text to make it easier to enter, for example:

```
<property name="keyInPlainText"><value>true</value></property>
```

```
<property name="secretKey"><value>jaspersoftInSanFrancisco</value></property>
```



The text `jaspersoftInSanFrancisco` is 24 characters long, therefore the two properties above work with their default values. However, for better security, we recommend that they be changed.

5. The last two properties may be left unchanged. They are set to DESede by default. The default values are the following:

```
<property name="secretKeyAlgorithm"><value>DESede</value></property>
```

```
<property name="cipherTransformation"><value>DESede/CBC/PKCS5Padding</value></property>
```



The `secretKey`, `secretKeyAlgorithm`, and `cipherTransformation` property settings must be consistent with each other. For example, different algorithms expect different key lengths.

6. Save and close the file. Encryption is now enabled for the JasperReports Server application upon the next restart.

11.4.1 Specifying Encryption Settings - Reference Table

The information in the [Table 11-1](#) is a summary of the available password encryption configuration options described in the previous section.

Table 11-1 Password Encryption Configuration Options

Configuration File		
...\WEB-INF\applicationContext-security.xml		
Property	Bean	Description
passwordEncoder	daoAuthenticationProvider	Comment this property out to disable the encryption of passwords.
allowEncoding	passwordEncoder	Determines whether JasperReports Server should encrypt the passwords it writes to the database. Set it to TRUE to use encrypted passwords.
secretKey	passwordEncoder	The salt key to use as a pass phrase when encrypting passwords. This string is used to encrypt passwords. This value can be a simple string or a numeric representation that can be parsed by Integer.decode(). For example: String: This is my secret key Numeric representation: 0xC8,0x43,0x29,0x49,0xAE,0x25,0x2F,0xA1,0xC1
keyInPlainText	passwordEncoder	Determines whether the secret key is a simple string or a numeric representation. Set this parameter to TRUE if the secretKey is a string; set it to FALSE if the key is a numeric representation.
secretKeyAlgorithm	passwordEncoder	The name of the algorithm to use, such as DESede.
cipherTransformation	passwordEncoder	The name of the transformation, such as DES/CBC/PKCS5Padding.

The secretKey, secretKeyAlgorithm, and cipherTransformation must be consistent with each other. For example, if the secretKeyAlgorithm is DESede, the secretKey must be 24 bytes long. For more information about secretKey, secretKeyAlgorithm, and cipherTransformation, see Sun's javax.crypto documentation.

11.5 Specifying Encryption Settings for the Import Utility

Before starting JasperReports Server, you must convert the plain text passwords that are currently stored in the repository export catalog that you created in section [11.3, “Running the Repository Export Utility,” on page 82](#). These plain-text passwords need to be converted to cipher text and reloaded into the database in order to successfully login after the server restarts. To do this, you must add the same encryption settings to the configuration file that is used by the import and export utilities.

1. Open the following configuration file for editing:

```
<js-install>/buildomatic/conf_source/ieCe/applicationContext-security.xml
```

2. This file contains the passwordEncoder bean definition, the same as in the JasperReports Server WAR, only by itself. Modify this file to specify the encryption settings as described in section [11.4, “Specifying Encryption Settings in the JasperReports Server WAR,” on page 82](#).

11.6 Recreating the JasperReports Server Database

Next, drop your existing jasperserver database and recreate an empty jasperserver database.

11.6.1 Dropping and Recreating the Database in PostgreSQL

1. Change directory to <js-install>/buildomatic/install_resources/sql/postgresql.
2. Start psql using an administrator account such as postgres:
`psql -U postgres`
3. Drop the jasperserver database, create a new one, and load the jasperserver schema:

```
drop database jasperserver;  
create database jasperserver encoding='utf8';  
\c jasperserver  
\i js-create.ddl  
\i quartz.ddl
```

11.6.2 Dropping and Recreating the Database in MySQL

1. Change directory to <js-install>/buildomatic/install_resources/sql/mysql.
2. Log into your MySQL client:
`mysql -u root -p`
3. Drop the jasperserver database, create a new one, and load the jasperserver schema:

```
mysql>drop database jasperserver;  
mysql>create database jasperserver character set utf8;  
mysql>use jasperserver;  
mysql>source js-create.ddl;  
mysql>source quartz.ddl;
```

11.7 Importing Your Repository Data

The import utility reloads all of your repository data. As the data is being saved to the repository, the password fields that were plain text are encrypted using the encryption settings you made in the previous sections.

To import your backup catalog to the repository:

1. Change directory to <js-install>/buildomatic.
2. Run the import utility with the command for your platform. Note that there are two dashes (--) in front of the command options:

Windows: `js-import-ce.bat --input-dir js-backup-catalog`

Linux: `js-import-ce.sh --input-dir js-backup-catalog`

For information about running the import utility, see [Chapter 12, “Configuring the Import-Export Utilities,”](#) on page 87.

11.8 Starting the Application Server

You can now start your application server. Your database should already be running.

11.9 Logging into JasperReports Server

You can now log into JasperReports Server.

Enter your user ID and password in the same manner as you did before encryption was turned on. You can check the contents of the `JUser` table in the `jasperserver` database and examine the password column to see that the password is no longer stored in plain text.

CHAPTER 12 CONFIGURING THE IMPORT-EXPORT UTILITIES

The import and export utilities add resources to or extract resources from the JasperReports Server repository. Typically, users export data from their previous instance and import it into their new installation when upgrading JasperReports Server. The import utility is also used at installation time to load the sample data into the repository.

This chapter covers the basic importing and exporting commands. For more information about command options, see the *JasperReports Server Administrator Guide*.

This chapter contains the following sections:

- [Introduction](#)
- [Import-Export Configuration Files](#)
- [Changing Your Configuration Settings](#)
- [Deploying a JDBC Database Driver](#)
- [Running Import or Export](#)
- [Configuring the Import-Export Utility for JasperServer 3.7](#)

12.1 Introduction

You can import and export resources by running either of these types of scripts:

Type	Example Command	Description
Buildomatic	<code>js-ant export-everything-ce -DexportFile=js-catalog-exp.zip</code>	Auto-configured buildomatic Ant script
Shell Script	<code>js-export-ce.sh --everything --output-file=js-catalog-exp.zip</code>	Import-export shell scripts

Both types of scripts are located in:

```
<js-install>/buildomatic
```

You use the same commands, regardless of operating system, to run the buildomatic Ant scripts. The shell scripts, on the other hand, have different names:

Windows Commands	Linux and Mac OSX Commands
<code>js-import-ce.bat</code>	<code>js-import-ce.sh</code>
<code>js-export-ce.bat</code>	<code>js-export-ce.sh</code>

The redundant sets of commands are being merged, and eventually, all database configuration will be done automatically from the `buildomatic/default_master.properties`. However, as of JasperReports Server version 4.7.0, this merging work is not complete.

If you are running the `js-import-ce.sh/bat` or `js-export-ce.sh/bat` shell scripts, you might have to manually configure some databases as described in the following section.

12.2 Import-Export Configuration Files

The import-export utility files are in the `buildomatic` folder. Use or modify these files to make configuration changes.

File or Location	Purpose
<code><js-install>/buildomatic/js-import-ce.bat</code> and <code>.sh</code>	Import scripts for Windows and Linux, respectively
<code><js-install>/buildomatic/js-export-ce.bat</code> and <code>.sh</code>	Export scripts for Windows and Linux, respectively
<code><js-install>/buildomatic/default_master.properties</code>	File that you must edit (already in place if you installed from the binary installer)
<code><js-install>/buildomatic/build_conf/default/js.jdbc.properties</code>	Database and hibernate dialect settings file (put in place after you run <code>js-ant gen-config</code>)
<code><js-install>/buildomatic/conf_source/ieCe/log4j.properties</code>	<code>log4j.properties</code> file controls output logging levels
<code><js-install>/buildomatic/conf_source/ieCe/applicationContext-*.xml</code>	Spring configuration files
<code><js-install>/buildomatic/conf_source/ieCe/lib</code>	All of the JasperReports Server jar files and the JDBC driver location

12.3 Changing Your Configuration Settings

If you are running the `js-import-ce.bat/.sh` or `js-export-ce.bat/.sh` shell scripts, then this section applies.

When you install JasperReports Server from the installer binary, the import and export shell scripts are automatically configured. However, if you are installing the WAR file distribution you must modify the `default_master.properties` configuration file to include your database settings and perform other configuration tasks.

12.3.1 Creating a default_master.properties File

If you don't have a `<js-install>/buildomatic/default_master.properties` file in place, create one. For instance, copy and rename `buildomatic/sample_conf/postgresql_master.properties` to `buildomatic/default_master.properties`.

Perform the following steps:

1. Edit: `<js-install>/buildomatic/default_master.properties`.
Perform [step 2](#) to [step 4](#) of the procedure **“To install the WAR file using js-install scripts:” on page 32**.
2. Run: `js-ant gen-config`.

12.3.2 Location of Properties Files and JDBC Driver

Paths and file names for properties files are:

- `<js-install>/buildomatic/build_conf/default/js.jdbc.properties`
- `<js-install>/buildomatic/build_conf/default/js.quartz.properties` (only for PostgreSQL)

The JDBC driver file must be in this directory:

- `<js-install>builddomatic/conf_source/ieCe/lib`
Copy your DB type JDBC driver to this folder.

The JDBC drivers are not in place for the MySQL database.

12.3.3 Checking the `js.jdbc.properties` File

Table 12-1 and **Table 12-2** contain sample settings for each database.

The `js.jdbc.properties` file will be automatically configured by `builddomatic`. Check the file to make sure the settings are correct for your environment. You may specify an encrypted password instead of the clear-text password by default by modifying `js.jdbc.properties`. For instructions on how to encrypt passwords, see the System Configuration chapter of the *JasperReports Server Administration Guide*.

Table 12-1 JDBC Settings in the `js.jdbc.properties` File

Database	Sample Property Values
PostgreSQL	<pre> metadata.hibernate.dialect= com.jaspersoft.hibernate.dialect.PostgresqlNoBlobDialect metadata.jdbc.driverClassName=org.postgresql.Driver metadata.jdbc.url=jdbc:postgresql://localhost:5432/jasperserver metadata.jdbc.username=postgres metadata.jdbc.password=postgres or metadata.jdbc.encryptedPassword=encrypted-postgres </pre>
MySQL	<pre> metadata.hibernate.dialect=org.hibernate.dialect.MySQLInnoDBDialect (MySQL 5.1) metadata.hibernate.dialect=org.hibernate.dialect.MySQL5InnoDBDialect (MySQL 5.5) metadata.jdbc.driverClassName=com.mysql.jdbc.Driver metadata.jdbc.url=jdbc:mysql://localhost:3306/ jasperserver?useUnicode=true&characterEncoding=UTF-8 metadata.jdbc.username=root metadata.jdbc.password=password or metadata.jdbc.encryptedPassword=encrypted-password </pre>

12.3.4 Checking the `js.quartz.properties` File

If your database is PostgreSQL, check your `js.quartz.properties` file. This file will be automatically configured by `builddomatic` and is located in:

`<js-install>/builddomatic/build_conf/default`

Table 12-2 Quartz Settings in the `js.quartz.properties` File

Database	Sample Property Values
PostgreSQL	<pre> quartz.delegateClass=org.quartz.impl.jdbcjobstore.PostgreSQLDelegate quartz.tablePrefix=QRTZ_ </pre>

12.4 Deploying a JDBC Database Driver

To run the import-export shell scripts, you need the proper JDBC driver. This allows a connection to be made to the JasperReports Server repository database.

Put the appropriate JDBC driver JAR into the following directory:

```
<js-install>/buildomatic/conf_source/ieCe/lib
```

All Jaspersoft distributed JDBC drivers can be found at this location:

```
<js-install>/buildomatic/conf_source/db/<db-type>/jdbc
```

12.5 Running Import or Export

To see that the import and export shell scripts are properly configured, you can run the scripts using the `--help` option (with two dashes `--`) that displays the command options.

On Windows and Linux, run these commands:

```
Windows: js-import-ce.bat --help
         js-export-ce.bat --help
Linux:   js-import-ce.sh --help
         js-export-ce.sh --help
```

If your repository contains international characters, you may need to perform additional configuration for the import and export utilities. See section [A.9.1, “Exporting a Repository That Contains UTF-8,” on page 102](#).

For complete information on the standard import-export options refer to the *JasperReports Server Administrator Guide*.

12.5.1 Import-Export Access Events Option

As of JasperReports Server 4.0.0, there is a new option for import and export. This option is `--include-access-events`. Specifying this option will allow the import or export of access event records stored in the JasperServer repository database.

12.6 Configuring the Import-Export Utility for JasperServer 3.7

You may need to configure your 3.7.0 import-export utility as part of the upgrade to 4.7.0 process.

In 3.7.0, the import-export shell scripts and configurations are located in the `<js-install-3.7.0>/scripts` folder.

There are two sub-folders that hold the configuration property files and the required jar files:

```
<js-install-3.7.0>/scripts/config
<js-install-3.7.0>/scripts/lib
```

To configure import-export for your database type and to handle database password changes you have made in your system, check and correct settings in this file:

```
<js-install-3.7.0>/scripts/config/js.jdbc.properties
```

Additionally, you will need to deploy the appropriate JDBC driver to the following folder:

```
<js-install-3.7.0>/scripts/lib
```

For more information about checking and correction the `js.jdbc.properties` file, see section [12.3.3, “Checking the `js.jdbc.properties` File,” on page 89](#).

APPENDIX A TROUBLESHOOTING

This appendix contains the following sections:

- **Binary Installer Freezes**
- **Error Running Buildomatic Scripts**
- **Unable to Edit Files on Windows 7**
- **Bash Shell for Solaris, IBM AIX, HP UX and FreeBSD**
- **Linux Installer Issue with Unknown Host Error**
- **Problem Starting JasperReports Server on the Mac**
- **Database-related Problems**
- **Application Server-related Problems**
- **Problems Importing and Exporting Data from the Repository**
- **Problems Importing and Exporting Data from the Repository**

A.1 Binary Installer Freezes

If you run the JasperReports Server installer on any platform and the installer freezes, the following resources can help you find the source of the error.

A.1.1 Installer Log Files

If you run the JasperReports Server installer on any platform and the installer freezes, it is helpful to look at the log file created by the installer. This log file records the status and completion of installer operations. If your installer has had an explicit error, there may be a specific error message in the log. At a minimum, the log file should help narrow where the error has occurred even if there is not a specific error message.

You can find the installer log in the following locations:

```
Windows: <js-install>/installation.log
Linux:   <js-install>/installation.log
Mac     <js-install>/installation.log
```

If you have tried multiple installs, make sure you view the most recent install log file.

A.1.2 Installer DebugTrace Mode

If you run the JasperReports Server installer on any platform and have a problem, you can run the installer a second time using the `--debugtrace` option. The `--debugtrace` option creates a binary output file that gives precise details about the execution of the installer and any problems encountered; this file can be analyzed by Jaspersoft Technical Support.

To use the `--debugtrace` option, you must run the installer from the command line and specify an output filename. The precise command depends on your platform (Linux, Windows, or Mac OSX). For example, you can execute the installer with a command similar to the following:

```
jasperreports-server-<ver>-linux-x64-installer.run --debugtrace install-trace-out.bin
```

When you run the installer in `--debugtrace` mode, the installer will take extra time to write the binary output file. The final size of the output file is approximately 10 mg. Contact Jaspersoft Technical Support for help reading the contents of this file.

A.2 Error Running Buildomatic Scripts

The buildomatic scripts depend on both Java and Apache Ant. There are two common configuration errors when attempting to do an installation using these scripts (if you are not using the included, bundled Apache Ant).

A.2.1 Missing Java JDK

If you have the Java JRE (Java Runtime Environment) instead of the JDK, you will not have the additional utilities that are required. In particular, an error referring to the `tools.jar` might occur, as in the following message:

```
[exec] [ERROR] BUILD FAILURE
[exec] [INFO] -----
[exec] [INFO] Compilation failure
[exec] Unable to locate the Javac Compiler in:
[exec]   c:\Program Files\Java\jdk1.6.0_10\jre\..\lib\tools.jar
[exec] Please ensure you are using JDK 1.6 or above and
[exec] not a JRE (the com.sun.tools.javac.Main class is required).
[exec] In most cases you can change the location of your Java
[exec] installation by setting the JAVA_HOME environment variable.
```

The solution is to download and install the Sun Java JDK, labeled as the Java SE Development Kit on the Sun web site.

If you are upgrading JasperReports Server, you can also use the Java 1.6 JDK bundled in the previously installed version, as described in section [8.8.1, “Handling JasperReports Server Customizations,” on page 70](#).

A.2.2 Forgot to Copy the File `ant-contrib.jar`

If you are using your own version of Ant and your Ant instance does not have the `ant-contrib.jar` in the `lib` directory, you will get an error similar to the following:

```
BUILD FAILED
c:\js-builds\jasperserver\buildomatic\install.xml:6:
```

Ant failed to create a task or type. To correct the error, copy `<js-install>/buildomatic/extra-jars/ant-contrib.jar` to your `<apache-ant>/lib` directory.

A.2.3 Older Apache Ant Version

As of the release of JasperReports Server 4.0.0, Apache Ant version 1.8.1 or higher is required. There are improvements to error handling routines in the buildomatic js-install scripts which required the higher level of Ant. If you are using an older version of Ant, you will get an error similar to the following:

```
BUILD FAILED
c:\js-builds\jasperserver\buildomatic\install.xml:37:
Problem: failed to create task or type componentdef
```

To check your version of Ant and verify that it is at a high enough level, enter:

```
ant -version
```

If you have a lower version of Ant, check to see if it is set in your class path by entering:

```
echo %CLASSPATH
```

To use the JasperReports Server version of Ant, update your CLASSPATH variable to point at the <js-install>/apache-ant/bin directory.

A.3 Unable to Edit Files on Windows 7

In some cases, you may want to manually edit files under your C:/Program Files directory during or after installation. For security reasons, Windows 7 doesn't allow normal processes to change files in the Program Files folder. When you attempt to edit these files, you may see an error such as the following:

You don't have permission to save in this location. Contact the administrator to obtain permission.

You can edit these files by running as administrator. For example, to edit these files with Notepad on Windows 7:

Click **Start > All Programs > Accessories**, right-click **Notepad**, and click **Run as administrator**.

A.4 Bash Shell for Solaris, IBM AIX, HP UX and FreeBSD

To execute the js-install shell scripts described in Section 5 of this guide, the bash shell is required. The js-install and js-upgrade scripts which are found in the buildomatic folder are the following:

```
js-install-ce.sh
js-upgrade-newdb-ce.sh
js-upgrade-samedb-ce.sh
```

The bash shell is not included by default in all Unix platforms. In the case where the bash shell is not available, it is necessary to download and install the bash shell. Bash shells are available for the following platforms:

```
Solaris
IBM AIX
HP UX
FreeBSD
```

Alternatively, you can manually run the same "buildomatic" Ant targets that are run by the js-install script. These Ant targets are listed in Section 5.4 "Installing WAR File Manually".

Also, check that you have updated your local Ant to include the ant-contrib.jar which supports conditional logic in Ant. The ant-contrib.jar is found in the location below and it should be copied to your <ant_home>/lib folder:

```
buildomatic/extra-jars/ant-contrib.jar
```

For updating your local Ant instance with the ant-contrib.jar see Section A.2.2 above ("Forgot to Copy the File ant-contrib.jar").

If you try and use the bundled ant that is included with the JasperServer WAR file Distribution ZIP package, you may get the same non-bash syntax error. You may get the error below, for example:

```
js-ant help-install
ANT_HOME=../apache-ant: is not an identifier
```

If you have the bash shell installed, you can try executing the js-ant command by calling bash explicitly, for example:

```
bash js-ant help-install
```

A.5 Linux Installer Issue with Unknown Host Error

If a Linux server does not have proper hostname entries in the `/etc/hosts` file, it is possible to get installer errors.

The installer carries out an import operation in order to load the core, minimal data into the repository database. This import operation can fail if the host is not configured.

If the import operation fails during installation, the installation will also fail. However, there should be an `installation.log` in the root of the installation folder to help debug the problem. The `installation.log` is located here:

```
<js-install>/installation.log
```

An improperly configured hosts file typically causes the log, or error messages displayed on the console, to contain error messages such as these:

```
Caused by: java.net.NoRouteToHostException: No route to host
com.mysql.jdbc.exceptions.jdbc4.CommunicationsException: Communications link failure
ERROR Cache:145 - Unable to set localhost. This prevents creation of a GUID
java.net.UnknownHostException
org.quartz.SchedulerException: Couldn't get host name!
```

To fix the hosts file:

1. Include entries that look similar to these:
127.0.0.1 localhost.localdomain localhost
172.17.5.0 myserver.mydomain.com myserver
2. After fixing the hosts file, reinstall JasperReports Server.

A.6 Problem Starting JasperReports Server on the Mac

Jaspersoft has seen some issues caused by the improper shutdown of the Tomcat included with the JasperReports Server. This could be related to the machine being shutdown while Tomcat is running.

When the Tomcat scripts start Tomcat, they write a pid (Process ID) file to the Tomcat folder. Tomcat uses this to determine whether the Tomcat instance is already running. When Tomcat is shutdown, this pid file is removed. However, if the pid file is not removed on shutdown, Tomcat will fail to start up.

You may see this when you double-click the `jasperServerStart.app` startup. It will seem like JasperReports Server is starting up but it never actually starts up.

In order to recover from this issue, you will need to manually delete the pid file.

Delete catalina.pid using Finder:

1. Navigate to the `<js-install>/tomcat/temp` folder.
For instance: `/Applications/jasperreports-server-<ver>/tomcat/temp`
2. Delete `catalina.pid`.

Delete the catalina.pid file using Terminal shell:

1. Open a Terminal shell (Finder > Go > Utilities > Terminal Icon)
2. Navigate to the <js-install>/tomcat/temp folder.
For instance: /Applications/jasperreports-server-<ver>/tomcat/temp.
3. Enter the following command:
`rm catalina.pid`

To start and stop the PostgreSQL and Tomcat components separately from the command line shell:

1. Open a Terminal shell (Finder > Go > Utilities > Terminal Icon).
2. Navigate to the <js-install> folder.
For instance: /Applications/jasperreports-server-<ver>.
3. To Start:
`./ctlscript start postgresql`
`./ctlscript start tomcat`
4. To shutdown:
`./ctlscript stop`
or
`./ctlscript stop tomcat`
`./ctlscript stop postgresql`

A.7 Database-related Problems

A.7.1 Installation Error Using PostgreSQL 8.1

During installation using the installer, if you select the option to use an existing database and it's PostgreSQL 8.1, the installer displays error messages about granting privileges, and then finishes installing all the files.

The problem is caused by this difference in the grant command syntax in PostgreSQL 8.1 and other PostgreSQL 8.x versions:

- PostgreSQL 8.1
`GRANT ALL PRIVILEGES ON hibernate_sequence TO jasperdb;`
- Other PostgreSQL 8.x
`GRANT ALL PRIVILEGES ON SEQUENCE hibernate_sequence TO jasperdb;`



If you install the server from the WAR file distribution, using either js-install or buildomatic scripts, there is no problem.

To work around this problem:

1. On the PostgreSQL 8.1 command line, enter the correct command to grant privileges to the single sequence object in the server repository:
`GRANT ALL PRIVILEGES ON hibernate_sequence TO jasperdb;`
2. To use the optional sugarcrm sample database and reports, enter similar grant commands for the sugarcrm sequences.
The sample foodmart database contains no sequences.

A.7.2 Database Connectivity Errors

The most common problems encountered with a new JasperReports Server instance are database configuration problems. If the connection fails, perhaps the application server cannot find the driver for the data source. For example, in a default installation of JasperReports Server, Tomcat looks for data source drivers in <js-install>/apache-tomcat/lib. If the driver is not there, put a copy of the driver in this directory and restart Tomcat.

This section contains information that may help resolve other connectivity issues.

A.7.2.1 Testing the Database Connection

The simplest database configuration problem is an incorrect user name or password. If you encounter database problems upon startup or login, check the user name and password by logging directly into your RDBMS as described in the following sections.

You can connect to your database using the database configuration settings that are found in JasperReports Server. This validates the database hostname, port, username, and password that are being used.

If you are having trouble logging into JasperReports Server on the login page, you can check the users and passwords that exist by viewing the contents of the `jasperserver.JIUser` table.

A.7.2.2 Logging into MySQL

Start MySQL from the command line and try to log in directly using the `jasperdb` user, for example:

```
<mysql>/bin/mysql -u jasperdb -p
or
<mysql>/bin/mysql -u root -p
```

You are prompted for a password for the user you specified on the command line. Enter the appropriate password to login. The default password used in the sample configuration scripts is `password` (`jasperadmin` in 2.1 and earlier).

A.7.3 Maximum Packet Size in MySQL

If you are upgrading or importing into a MySQL database and your repository contains large objects such as images, you may see an error such as:

```
ERROR 1153 (08S01): Got a packet bigger than 'max_allowed_packet' bytes
```

The default `max_allowed_packet` on the MySQL server is 1M (one Megabyte = 1,048,576 bytes). The most effective fix is to change this value in the server configuration to accommodate the largest resource stored in your repository. The server configuration file is typically named `my.cnf` (or `my.ini`) and is located in the MySQL root directory, although this may vary. Change the configuration setting to a larger value, for example:

```
max_allowed_packet = 16M
```

For more information, see <http://dev.mysql.com/doc/refman/5.0/en/packet-too-large.html>.

After changing this value, restart the MySQL server. Then perform the upgrade or import step again.

A.7.4 Database Error after Changing Database Port Number

The default ports for supported databases are:

Database Setting	MySQL	PostgreSQL
Port	3306	5432

If you entered a different port when you installed your database, the JasperReports Server installer prompts you for the port number, and configures the server to communicate properly. If the port number has changed, or if you encounter a problem, check the database configuration files to verify your port number.

If it is incorrect, change it to the correct port number, save the file, and restart the application server. For more information, see section [A.8.3, “Configuration File Locations,” on page 98](#).

A.7.5 Case Sensitivity for Table and Column Names

Some databases are case-sensitive with respect to table names and will consider “customer” and “Customer” to be two different tables. If JasperReports Server is using a case-sensitive database, it’s important that the table names specified in

query strings in the JRXML file of a saved report match the actual table names found in the database. A mismatch may occur if you are transferring data from one database to another, which may cause the capitalization of table names to change.

Under Windows MySQL, table and column names are *not* case-sensitive.

Under Linux MySQL, table and column names are case-sensitive. Linux MySQL can be configured to be non-case-sensitive by setting the configuration parameter `lower_case_table_names` to 1 in the `my.ini` or `my.cnf` file. For more information search the MySQL documentation for a section about identifier case sensitivity.

Table and column names in PostgreSQL are case-sensitive.

A.7.6 PostgreSQL: Job Scheduling Error

If the Quartz settings under the PostgreSQL database have not been updated to specify the driver delegate class specific to PostgreSQL you will get errors when you try and run a scheduled report. The errors would look similar to the following:

```
Error while fetching Quartz runtime information
org.quartz.JobPersistenceException: Couldn't obtain triggers: Bad value for type int
org.postgresql.util.PSQLException: Bad value for type int
```

If you see this error you will need to check your Quartz properties file found at the following location:

```
<tomcat>/webapps/jasperserver/WEB-INF/js.quartz.properties
```

Make sure that the following property does not have the standard driver delegate, but instead has the PostgreSQL specific driver delegate. It should look like the following for PostgreSQL:

```
quartz.delegateClass=org.quartz.impl.jdbcjobstore.PostgreSQLDelegate
```

A.7.7 Error Running Scheduled Report

If you setup a scheduled report, chose to run it, and chose to save it as HTML or RTF, the report size can potentially get quite large. If you are running MySQL and you get the following error:

```
JDBC exception on Hibernate data access
org.hibernate.exception.GenericJDBCException: could not insert
```

the problem may be the default size of the MySQL blob datatype. You can increase the size of this datatype by updating your `my.ini` or `my.cnf` MySQL configuration file with the following setting:

```
max_allowed_packet=32M
```

A.7.8 Error Running a Report

If you can log into JasperReports Server but encounter an error when running a report within it, you can browse the repository to identify and resolve the problem.

One common problem with an individual report is the data source being used. To validate a data source connection:

1. Log into JasperReports Server as a user with administrative permissions and locate the report unit that returns errors.
2. Select the report and click the **Edit** button in the toolbar to identify the data source the report uses. The data source name is found on the fourth edit page.
3. Select this data source in the repository and click the **Edit** button in the toolbar.
4. Review the information specified for this data source.
5. Click the **Test Connection** button in order to validate the connection.

If the connection fails, perhaps the application server cannot find the driver for the data source. For example, in a default installation of JasperReports Server, Tomcat looks for data source drivers in `<js-install>/apache-tomcat/lib`.

6. Click **Save** or **Cancel** when you are done.
7. Test your report. If it still returns errors, edit the data source again and try checking other values, such as the port used by the database.

A.8 Application Server-related Problems

A.8.1 Memory Issues Running Under Tomcat

If you experience problems related to the release of memory or to container tag pooling, the following steps might solve the problem:

1. Set the following parameter in the global \$CATALINA_BASE/conf/web.xml:
enablepooling = false
2. Restart Tomcat.

A.8.2 Java Out of Memory Error

If you encounter a Java out of memory error, try increasing your Java heap size setting. See section 6.2, “[Setting JVM Options for Application Servers](#),” on page 43. As a minimum, add `-Xms1024m -Xmx1280m` to your `JAVA_OPTS` setting, but you may increase that to `-Xms1024m -Xmx2048m`, assuming your server can support these settings.

This Java option is set within the application server, so you must set it then restart your application server.

A.8.3 Configuration File Locations

JasperReports Server configuration properties are found in the following files, according to your application server.

The following list shows the location of the properties for supported application servers:

Tomcat: <tomcat>/webapps/jasperserver/META-INF/context.xml
 <tomcat>/webapps/jasperserver/WEB-INF/hibernate.properties
 <tomcat>/apache-tomcat/webapps/jasperserver/WEB-INF/web.xml (JNDI config)
 <tomcat>/apache-tomcat/config/Catalina/localhost/jasperserver.xml (delete: see below)

JBoss: <jboss>/server/default/deploy/js-postgresql-ds.xml or js-<mysql>-ds.xml
 <jboss>/server/default/deploy/jasperserver.war/WEB-INF/hibernate.properties
 <jboss>/server/default/deploy/jasperserver.war/WEB-INF/web.xml
 <jboss>/server/default/deploy/jasperserver.war/WEB-INF/jboss-web.xml

GlassFish: <glassfish>/domains/domain1/autodeploy/jasperserver.war/WEB-INF/hibernate.properties
 <glassfish>/domains/domain1/autodeploy/jasperserver.war/WEB-INF/js.quartz.properties
 <glassfish>/domains/domain1/config/domain.xml

A.8.4 Context.xml under Tomcat: Special Case

If you deploy multiple instances of JasperServer to Tomcat, the context.xml (database connection configuration) can be superseded by a file in this location: `<tomcat>/conf/Catalina/localhost/jasperserver.xml` file.

When JasperServer is deployed, the context.xml will be copied to `<tomcat>/conf/Catalina/localhost/jasperserver.xml` (Tomcat does this by default).

Now, if you make changes to your `<tomcat>/webapps/jasperserver/META-INF/context.xml`, Tomcat will not “see” them. Instead, the `jasperserver.xml` will be used. This is confusing, but is the way that Tomcat operates.

If you edit your context.xml to fix a database problem:

```
<tomcat>/webapps/jasperserver/META-INF/context.xml
```

Remember to delete the `jasperserver.xml` file:

```
<tomcat>/conf/Catalina/localhost/jasperserver.xml     (delete this file)
```

A.8.5 Tomcat 6 Installed Using apt-get

If you are installing JasperReports Server to an instance of Tomcat that has been installed using a package managers such as apt-get, yum, or rpm then you can use the CATALINA_HOME and CATALINA_BASE properties found in your default_master.properties file.

Go to the section of the default_master.properties that looks like this:

```
# Tomcat app server root dir
appServerDir = C:\\Program Files\\Apache Software Foundation\\Tomcat 7.0
# appServerDir = /home/devuser/apache-tomcat-7.0.26
# if linux package managed tomcat instance, set two properties below
# CATALINA_HOME = /usr/share/tomcat6
# CATALINA_BASE = /var/lib/tomcat6
```

and change it to the following:

```
# Tomcat app server root dir
# appServerDir = C:\\Program Files\\Apache Software Foundation\\Tomcat 7.0
# appServerDir = /home/devuser/apache-tomcat-7.0.26
# if linux package managed tomcat instance, set two properties below
CATALINA_HOME = /usr/share/tomcat6
CATALINA_BASE = /var/lib/tomcat6
```

Note that you must set both CATALINA_HOME and CATALINA_BASE.

A.8.6 GlassFish Modifications

A.8.6.1 Ad Hoc for OLAP Problems Under GlassFish 2

Conflicting JAR files can cause problems with Ad Hoc for OLAP when the application is hosted by Glassfish version 2. The workaround is:

1. Remove <glassfish>/lib/jackson-asl-0.9.4.jar
2. Copy the jackson-core-lgpl-1.7.4.jar and jackson-mapper-lgpl-1.7.4.jar files from:

```
<glassfish>/domains/domain1/autodeploy/jasperserver.war/WEB-INF/lib
```

to:

```
<glassfish>/lib
```

A.8.6.2 Using a Custom Domain

If your application server is Glassfish and you're using a custom domain, set up the following authentication information in the <js-install>/buildomatic/build_conf/default/app.srv.properties:

- Custom domain name
- Port number
- User name
- Password

Do this before running the js-install script in section 5.2, “Installing the WAR File Using js-install Scripts,” on page 32. You should also recreate your buildomatic configuration settings.

To recreate your buildomatic configuration settings:

```
cd <js-install>/buildomatic
js-ant clean-config
js-ant gen-config
```

A.8.6.3 Using Glassfish 3.1.0

There is a known issue with Glassfish 3.1.0 where Java JVM options are not properly set. This issue is fixed in Glassfish 3.1.1 and later.

To set the JVM options in Glassfish 3.1.0:

1. Open this buildomatic property file:
`<js-install>/buildomatic/build_conf/default/app.srv.properties`
2. Uncomment the `glassfishPort` property as follows:
 from
`#glassfishPort=4848`
 to
`glassfishPort=4848`

A.8.7 JBoss Modifications

A.8.7.1 JBoss 7 Startup Error

JBoss7 has a default startup time period. If your JBoss 7 takes longer than 60 seconds to startup or deploy, you may receive the following error:

```
"(DeploymentScanner-threads - 1) Did not receive a response to the deployment operation within the allowed timeout period [60 seconds]. Check the server configuration file and the server logs to find more about the status of the deployment".
```

To fix this, you need to increase your `deployment-timeout` setting as follows:

1. Change to the JBoss standalone configuration directory.
`cd <jboss>/standalone/configuration`
2. Open the `standalone.xml` file.
3. Look for the `<subsystem xmlns="urn:jboss:domain:deployment-scanner:1.1">` element, for example:
`<subsystem xmlns="urn:jboss:domain:deployment-scanner:1.1">`
 `<deployment-scanner path="deployments" relative-to="jboss.server.base.dir" scan-`
 `interval="5000"/>`
`</subsystem>`
4. Edit this to add or set the attribute `deployment-timeout` to the desired amount of time in seconds, for example:
`<subsystem xmlns="urn:jboss:domain:deployment-scanner:1.1">`
 `<deployment-scanner path="deployments" relative-to="jboss.server.base.dir" scan-`
 `interval="5000" deployment-timeout="600">`
`</subsystem>`
5. Save the file.
 On server restart, your system will have the specified amount of time to start up.

A.8.7.2 JBoss Large INFO Log Message on Drill-through

JBoss has an internal mechanism to track and log information on unclosed JDBC connections. Jaspersoft OLAP Views leaves a connection open for performance reasons when doing a drill-through. In this case, JBoss puts a large INFO level message into the `server.log`.

To silence this INFO message, perform these steps:

1. Open the JBoss `log4j` configuration file for editing:
`<jboss>/server/default/conf/jboss-log4j.xml`
2. Set the logging level for the `CachedConnectionManager` class to the following value:

```
<category name="org.jboss.resource.connectionmanager.CachedConnectionManager">
<priority value="WARN"/>
</category>
```

A.8.7.3 JBoss 5.0.1 and 5.1.x Error

With JBoss 5.0.1 and 5.1.x, you may see the following error:

```
org.jboss.xb.binding.JBossXBRuntimeException: Failed to create a new SAX parser
Caused by: java.lang.ClassCastException
```

This is a class conflict with the xercesImpl-2.7.1.jar in JasperReports Server. To correct it, delete the following file:

```
<jboss>/server/default/deploy/jasperserver.war/WEB-INF/lib/xercesImpl-*.jar
```



When running the buildomatic scripts to deploy to JBoss, the xercesImpl-3.7.jar file is automatically deleted in order to fix this problem.

A.8.7.4 Using a Non-default JBoss Profile

If your application server is JBoss and you're using a profile other than the default, you need to set the `jboss.profile` property before running the `js-install` script in section 5.2, [“Installing the WAR File Using js-install Scripts,” on page 32](#):

1. Open this buildomatic property file:

```
<js-install>/buildomatic/build_conf/default/app.srv.properties
```

2. Uncomment the `jboss.profile` property and change the profile name as follows:

from

```
# jboss.profile = default
```

to

```
jboss.profile = <your_profile>
```

A.8.7.5 JBoss 4.2 XML/A Connection Fix

JBoss 4.2 includes the JBossWS service as a standard, default feature. JasperReports Server has web services support for XML/A connections.

The web services classes in JasperReports Server and JBoss can conflict and cause the following error when attempting to utilize a JasperReports Server XML/A connection:

```
javax.xml.soap.SOAPException: Unable to create message factory for
SOAP: org.jboss.ws.core.soap.MessageFactoryImpl
```

To prevent the web services class conflict, set the special Java JVM options for JBoss 4.2, as described in section 6.2.1, [“Tomcat and JBoss JVM Options,” on page 43](#).

A.8.8 Disabling User Session Persistence in Application Servers

JasperReports Server stores non-serializable data in its user sessions, which can cause errors after restarting your application server:

```
Exception loading sessions from persistent storage
Cause: java.io.NotSerializableException ...
```

The errors appear in the JasperReports Server log when users log in after the application server has been restarted. The errors do not appear to users, and they have no impact on JasperReports Server operations.

Because JasperReports Server user sessions are not persistent, you can configure your application server to disable persistence and avoid the error. For example, in Apache-Tomcat 5.5, 6, and 7 edit the file `<tomcat>/conf/context.xml` and locate the following lines:

```
<!-- Uncomment this to disable session persistence across Tomcat restarts -->
<!--
<Manager pathname="" />
-->
```

Remove the comment markers from lines 2 and 4 above, then restart Apache-Tomcat for the change to take effect. For other application servers, refer to the product documentation.

A.8.9 Session Error Using JasperReports Server and Tomcat 7

On some versions of Tomcat 7, a session error might occur while running reports, with the log error “A request has been denied as a potential CSRF attack.” This is due to a known conflict between security settings in Direct Web Remote library (DWR) 2.x and some versions of Tomcat 7.0.x:

- Tomcat 7 sets httpOnly on session ID cookies to safeguard against cross-site scripting (XSS) attacks.
- DWR 2.x uses session ID cookies to safeguard against cross-site request forgery (CSRF).

To work around this problem, you must modify these safeguards by doing one of the following:

- Disabling httpOnly for cookies in Tomcat

OR

- Allowing requests from other domains in DWR

For more information on the security impact and relative risks of these two choices, see, for example, the Cross-site Scripting and Cross-site Request Forgery pages at the [Open Web Application Security Project \(OWASP\)](#).

A.8.9.1 Disabling httpOnly for Cookies in Tomcat

The application server that hosts JasperReports Server handles the session cookie. To prevent malicious scripts on a client from accessing the session cookie, and thus the user connection, Tomcat 7 is set to use httpOnly cookies. This tells the browser that only the server may access the cookie, not scripts running on the client. When enabled, this setting safeguards against XSS attacks.

You can disable this by disabling httpOnly in the file <tomcat>/conf/context.xml:

```
<Context useHttpOnly="false">
...
</Context>
```

A.8.9.2 Allowing Requests from Other Domains in DWR

DWR is a server-side component used for Input Controls. By default, DWR uses session ID cookies to prevent against cross-site request forgery. You can disable the protection in DWR by setting the `crossDomainSessionSecurity` parameter for the `dwr` servlet in the file <tomcat>\webapps\jasperserver\WEB-INF\web.xml:

```
<servlet>
  <servlet-name>dwr</servlet-name>
<servlet-class>org.directwebremoting.spring.DwrSpringServlet</servlet-class>
  ...
  <init-param>
    <param-name>crossDomainSessionSecurity</param-name>
    <param-value>>false</param-value>
  </init-param>
</servlet>
```

A.9 Problems Importing and Exporting Data from the Repository

A.9.1 Exporting a Repository That Contains UTF-8

The following errors may happen when you have international characters in repository objects, for example, in user IDs.

A.9.1.1 Error During Export

Upgrading typically requires doing an export operation on your database. If you are using MySQL and getting this null pointer exception, it may be due to an incorrect character in the file `js.jdbc.properties`:

```
java.lang.NullPointerException
ResourceExporter.exportResource(ResourceExporter.java:258)
```

Check the URL in this file in `<js-install>buildomatic/build_conf/default/`; it should look like the following:

```
jdbc:mysql://localhost:3306/jasperserver?useUnicode=true&characterEncoding=UTF-8
```

Note the ampersand `&` character. It is incorrect if it appears as `&`. The `&` is only correct in an HTML or XML context. It is incorrect in a properties file.

A.9.2 Importing Scheduled Jobs with Update Option

You can experience a JasperReports Server bug if you import a set of resources that contain Report Jobs. These jobs are not loaded into the JasperReports Server repository if you are using the `--update` option of the import tool. To work around this problem, do not specify the `--update` option on your `js-import.bat/.sh` command line.

A.10 Problems with Upgrade

A.10.1 Include Audit Events on Upgrade

If you have auditing enabled and you run upgrade using `js-upgrade-newdb-ce.bat/sh`, audit events are not imported by default. To import audit events, you need to run an additional command after the `js-upgrade-newdb-ce` script completes. To do this, change to the `buildomatic` directory:

```
cd <js-install>/buildomatic
```

Then run one of the following commands (for example):

```
js-import-ce.bat --input-zip=js-my-export-all.zip --include-audit-events (Windows)
```

```
js-import-ce.sh --input-zip=js-my-export-all.zip --include-audit-events (Linux)
```

or

```
ant import -DimportFile=js-my-export-all.zip -DimportArgs="--include-audit-events"
```

These commands reimport all resources from the specified export file. Existing resources will not be overwritten and the audit event will be added.

