



JASPERREPORTS SERVER COMMUNITY PROJECT INSTALLATION GUIDE

RELEASE 4.2

<http://www.jasperforge.org>

Copyright © 2011 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 1011-JSO42-26 of the *JasperReports Server Community Project Installation Guide*.

TABLE OF CONTENTS

Chapter 1	Introduction	9
1.1	Conventions	10
1.2	Supported Java Version	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	14
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	17
2.6.4	Enabling Connections to a Remote Linux Host	18
2.7	Specifying PostgreSQL Parameters	18
2.8	Installing Sample Data	19
2.9	Installing iReport Designer	19
2.10	Completing the Installation	19
2.11	Post-Installation Steps	20
2.11.1	Updates Made by the Installer During Installation	20
2.11.2	Installer Output Log File Location	20
2.11.3	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
3.3	Start/Stop Apps — Mac OSX	24
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	26
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
5.2	Installing the WAR File Using Auto-install Scripts	31
5.3	Starting JasperReports Server	33
5.4	Logging into JasperReports Server	34
5.4.1	JasperReports Server Heartbeat	34
5.5	Troubleshooting Your JasperReports Server Configuration	35
5.5.1	JasperReports Server Startup Problems	35
5.5.2	Error Running a Report	35
5.5.3	Error Running Auto-install Scripts (js-install-ce.bat/sh)	35
5.5.4	Problem Connecting to a Cloud Database Instance	36
5.6	Running the Import and Export Utilities	36
5.6.1	Running Export from Buildomatic	36
5.6.2	Running Import from Buildomatic	37
5.6.3	Running the Import-Export Shell Scripts	37
5.7	Manual Buildomatic Install Steps	38
Chapter 6	Manual Database Creation and Other Information	39
6.1	Manually Creating the JasperReports Server Database	39
6.1.1	PostgreSQL 8.x and 9.x	40
6.1.2	MySQL	40
6.2	Setting JVM Options for Application Servers	41
6.2.1	Tomcat and JBoss JVM Options	41
6.2.2	Bundled Tomcat as a Windows Service JVM Options	42
6.2.3	Existing Tomcat as a Windows Service JVM Options	43
6.2.4	GlassFish JVM Options	43

6.3	Additional Buildomatic Configuration Information	44
6.3.1	Additional Commands for Buildomatic Settings	44
6.3.2	Buildomatic: Generated Property Files	44
6.3.3	Buildomatic: SQL Scripts Location	45
6.3.4	Buildomatic: Database Creation Statements Location	45
6.3.5	Buildomatic: JDBC Driver Locations	45
6.3.6	Buildomatic: Change your Deployed JDBC Driver	46
6.3.7	Buildomatic: JasperReports Server WAR File Location	46
6.3.8	Buildomatic: Sample Data Catalog ZIP Files	47
6.4	Notes on the Hibernate Properties File	47
6.5	Notes on Database Connections for Tomcat	47
6.6	Notes on Data Source Definitions for JBoss	48
6.6.1	Notes on Extra JBoss Configuration Steps	48
6.7	Notes on Database Connections for Glassfish	48
6.8	Report Scheduling Configuration	48
6.8.1	Mail Server Configuration Settings	49
6.8.2	Database settings for the Quartz Driver Delegate Class	49
6.8.3	Settings for the Report Scheduler Web URI	50
6.8.4	Settings for Import-Export	50
6.9	Notes on Updating XML/A Connection Definitions	50
Chapter 7	Upgrading from 3.7.0 to 4.2	53
7.1	Standard Upgrade Steps	54
7.2	Backing Up the JasperServer 3.7 Instance	54
7.3	Exporting 3.7.0 Repository Data	54
7.3.1	Export Using Buildomatic Scripts	55
7.3.2	Export Using the js-export Script	55
7.4	Preparing the JasperReports Server 4.2 WAR File Distribution	55
7.5	Configuring Buildomatic for Your Database and Application Server	56
7.5.1	Example Buildomatic Configuration	56
7.6	Upgrading to JasperReports Server 4.2	57
7.6.1	Auto-Upgrade Test Mode	57
7.6.2	Auto-Upgrade Script Help	58
7.6.3	Output Log Location	58
7.6.4	Errors	58
7.7	Starting JasperReports Server 4.2	58
7.8	Logging into JasperReports Server 4.2	58
7.8.1	Clearing Your Browser Cache	58
7.8.2	Logging into JasperReports Server	58
7.9	Additional Notes on JasperReports Server Upgrade	59
7.9.1	Handling JasperReports Server Customizations	59
7.9.2	Clearing the Application Server Work Directory	59
7.9.3	Clearing the Application Server Temp Directory	59
7.9.4	Clearing the Repository Cache Table	59
7.9.5	Updating the XML/A Connections (Optional)	59

7.9.6	Upgrading the Liferay Portal	60
7.10	Older Manual Upgrade Steps	60
Chapter 8	Upgrading from 4.0.0 or 4.1.0 to 4.2	61
8.1	Upgrade Steps	61
8.2	Backing Up Your JasperReports Server 4.0.0 or 4.1.0 Instance	62
8.2.1	Backing Up Your JasperReports Server WAR File	62
8.2.2	Backing Up Your JasperServer Database	62
8.3	Preparing the JasperReports Server 4.2 WAR File Distribution	62
8.4	Configuring Buildomatic for Your Database and Application Server	62
8.4.1	Example Buildomatic Configuration	63
8.5	Upgrading to JasperReports Server 4.2	64
8.5.1	Auto-upgrade Test	64
8.5.2	Output Log Location	64
8.5.3	Errors	64
8.6	Starting JasperReports Server 4.2	65
8.7	Logging into JasperReports Server 4.2	65
8.7.1	Clearing Your Browser Cache	65
8.7.2	Logging into JasperReports Server	65
8.8	Additional Information on Post-Upgrade Steps	65
8.8.1	Clearing the Application Server Work Directory	65
8.8.2	Clearing the Application Server Temp Directory	65
8.8.3	Clearing the Repository Cache Table	66
8.9	Running Buildomatic DB Upgrade Steps Manually	66
Chapter 9	Upgrade Notes for JasperServer 3.7.0	67
9.1	Upgrade from JasperServer 3.5.0	67
Chapter 10	Changing Password Encryption in JasperReports Server	69
10.1	Backing Up Your JasperReports Server Database	69
10.2	Stopping Your Application Server	69
10.3	Running the Repository Export Utility	70
10.4	Specifying Encryption Settings in the JasperReports Server WAR	70
10.4.1	Specifying Encryption Settings - Reference Table	70
10.5	Specifying Encryption Settings for the Import Utility	71
10.6	Recreating the JasperReports Server Database	71
10.6.1	Dropping and Recreating in PostgreSQL	71
10.6.2	Dropping and Recreating in MySQL	72
10.7	Importing Your Repository Data	72
10.8	Starting the Application Server	72
10.9	Logging into JasperReports Server	72
Chapter 11	Configuring the Import-Export Utilities	73
11.1	Introduction	73
11.2	Import-Export Configuration Files	74

11.3	Changing Your Configuration Settings	74
11.3.1	Creating a default_master.properties File	74
11.3.2	Location of Properties Files and JDBC Driver	74
11.3.3	Checking the js.jdbc.properties File	75
11.3.4	Checking the js.quartz.properties File	75
11.4	Deploying a JDBC Database Driver	75
11.5	Running Import or Export	76
11.5.1	Import-Export Access Events Option	76
11.6	Configuring the Import-Export Utility for JasperServer 3.7.0	76
Appendix A	Troubleshooting	77
A.1	Installation Error Using PostgreSQL 8.1	78
A.2	Installer Freezes	78
A.3	Database Connectivity Errors	78
A.3.1	Testing the Database Connection	78
A.3.2	Configuration File Locations	79
A.3.3	Context.xml under Tomcat: Special Case	79
A.3.4	Maximum Packet Size in MySQL	79
A.4	Error Running a Report	80
A.5	Database Error after Changing Database Port Number	80
A.6	Case Sensitivity for Table and Column Names	80
A.7	Java Out of Memory Error	81
A.8	Error Running Scheduled Report	81
A.9	Exporting a Repository That Contains UTF-8	81
A.9.1	Error During JasperServer 1.2 Export	81
A.10	Importing Scheduled Jobs with Update Option	81
A.11	Glassfish Modifications	82
A.11.1	Using a Custom Domain	82
A.11.2	Using Glassfish 3.1.0	82
A.12	JBoss Modifications	82
A.12.1	JBoss 4.2 XML/A Connection Fix	82
A.12.2	JBoss Large INFO Log Message on Drill-through	82
A.12.3	JBoss 5.0.1 and 5.1.x Error	83
A.13	PostgreSQL: Job Scheduling Error	83
A.14	Error Running Buildomatic Scripts	83
A.14.1	Missing Java JDK	83
A.14.2	Forgot to Copy the File ant-contrib.jar	84
A.14.3	Older Apache Ant Version	84
A.15	Troubleshooting on Solaris	84
A.16	Disabling User Session Persistence in Application Servers	84
A.17	Linux Installer Issue with Unknown Host Error	85
A.18	Problem Starting JasperReports Server on the Mac	85
A.19	Session Error Using JasperReports Server and Tomcat 7	86

CHAPTER 1 INTRODUCTION

The 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.

In a nutshell, JasperReports Server provides the ability to:

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

For business intelligence users, we offer Jaspersoft OLAP, which runs on the server. This component is described in its own user guide.

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](#).
- JasperForge, our [community website](#), hosts open source projects, associated source code, tools for bug tracking, version control, and forums for community discussion. You'll find development and implementation advice, a secure development environment for community-driven projects, and community feedback.
- 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**
- **Supported Java Version**
- **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 is 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 Supported Java Version

JasperReports Server supports Java 1.5 and 1.6. Versions earlier than Java 1.5 are not supported.

JasperReports Server is tested and certified using 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 Binary Distribution Zip	Used for manual installation on Windows, Linux, Mac, and other platforms.

The installers have the capability of installing JasperReports Server, automatically configuring the JasperReports Server database, and optionally installing sample data that highlight the server features.

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:

Platform	Versions supported
Linux	Red Hat Enterprise Linux SUSE Ubuntu And additional Linux distributions

Platform	Versions supported
Mac OSX	10.5 (Leopard) 10.6 (Snow Leopard)
Windows	XP Vista Windows 7

1.3.1.1 Installer Naming for 32 bit and 64 bit

Native 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-4.2.1-windows-x86-installer.exe jasperreports-server-cp-4.2.1-linux-x86-installer.run jasperreports-server-cp-4.2.1-osx-x86-installer.app.zip
64 bit installer	jasperreports-server-cp-4.2.1-windows-x64-installer.exe jasperreports-server-cp-4.2.1-linux-x64-installer.run jasperreports-server-cp-4.2.1-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. If using an existing, remote PostgreSQL on Linux, see 2.6.4, “Enabling Connections to a Remote Linux Host,” on page 18 .

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 Linux 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. This makes it more convenient for users to manage JasperReports Server under the Windows operating system. JasperReports Server can still be started and stopped from the Windows Startup menu. But, additionally, it can be managed from the Windows Services panel.

Also, the bundled PostgreSQL and Tomcat applications are automatically restarted when the host Windows system is restarted. If you do not want to run these components to automatically restart, you can change the settings in the Windows Services panel.

You can do this in the following location:

Control Panel > System and Security > Administrative Tools > Services

Then, change the Startup Type from automatic to manual.

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

- ♦ jasperreportsPostgreSQL
- ♦ jasperreportsTomcat



The Windows installer installs PostgreSQL and Tomcat as Services that are automatically restarted on a Windows system restart. If you don't want them to automatically restart, set the Start Type of these applications to Manual in the Windows Services panel.

1.3.2 WAR File Binary Distribution Support

The WAR file binary distribution is the package you use to install the JasperReports Server application if you cannot use the installer. The WAR file supports more applications than the installer. You can use a database other than PostgreSQL and an application server other than Apache Tomcat.

Note: 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 [2.6.4, “Enabling Connections to a Remote Linux Host,” on page 18](#). The application server should reside on the local machine.

Auto-install scripts included in the WAR file distribution that handle the installation tasks by putting your local settings in a single properties file. These scripts are named:

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



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.

If you cannot use the auto-install scripts for some reason, such as encountering errors when trying to use the scripts, the WAR file distribution contains another set of scripts you can use: the buildomatic Ant scripts. Using these scripts instead of the auto-install scripts accomplishes the same goal (to configure and deploy JasperReports Server) but takes more steps. The buildomatic Ant scripts are found in the buildomatic directory. They rely on the Apache Ant build tool, included in the WAR

file binary distribution, and the Java JVM for execution. For more information about the buildomatic Ant scripts, see [6.3, “Additional Buildomatic Configuration Information,” on page 44](#).

The main contents of the WAR file binary distribution are:

Content Item	Description
JasperReports Server auto-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.15, “Troubleshooting on Solaris,” on page 84](#).

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, including 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 are using the bundled Tomcat and PostgreSQL software, UTF-8 is configured by default. If you are using any other existing software, refer to the *JasperReports Server Localization Guide* for instructions on how to configure 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**
- **Specifying PostgreSQL Parameters**
- **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-4.2.1-windows-x86-installer.exe    (32 bit)
jasperreports-server-cp-4.2.1-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-4.2.1-linux-x86-installer.run    (32 bit)
```

```
./jasperreports-server-cp-4.2.1-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-4.2.1-osx-x86-installer.app (32 bit)
jasperreports-server-cp-4.2.1-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-4.2.1
Linux:    <USER_HOME>/jasperreports-server-cp-4.2.1
Mac OSX   /Applications/jasperreports-server-cp-4.2.1
```



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

2.5 Selecting a Tomcat Configuration

JasperReports Server requires a web application server in order to run. The installer is pre-configured to run with the Apache Tomcat server. There are two options available for your Tomcat configuration.

The first option is to choose a bundled Tomcat. If you choose this option, the installer puts an instance of Tomcat 6 onto your system. Click **Next**. 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**.

The second option is to choose an existing Tomcat. If you already have an instance of Tomcat on your system, then you can choose this option. Choose the existing Tomcat option and click **Next**. You are prompted for its location. Enter the correct location for Tomcat or click **Browse** to locate and select another location. Click **Next**. You are prompted for Tomcat's server port and shutdown port. Accept the default values or enter alternate values, then click **Next**.

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:

- To install the PostgreSQL bundled with JasperReports Server
- To use an existing instance of PostgreSQL

2.6.1 Choosing the Bundled PostgreSQL

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

Values to be entered or set to defaults for the bundled PostgreSQL configuration:

Parameter	Default Value and Description
Port	5432 - User must choose an alternate port if 5432 is in use.
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.

The second option is to choose an existing PostgreSQL. You can choose this option if you already have an instance of PostgreSQL running locally or remotely.

2.6.2 Choosing an Existing PostgreSQL on a Local Host

If you choose the option to use an existing PostgreSQL and click **Next**, you are prompted for the location of PostgreSQL and the port to use. If you have an instance of PostgreSQL installed locally, then take the default, which is 127.0.0.1 (otherwise known as localhost).

You can check the version of the local PostgreSQL instance by entering this command:

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

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

2.6.3 Using an Existing to 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. It is best if the client tools are the same version as the remote Postgresql 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
```

If your remote PostgreSQL runs under Linux, you might also need to enable connections as described in the next section.

2.6.4 Enabling Connections to a Remote Linux Host

On some Linux platforms, the default PostgreSQL installation does not 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:
`/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 `/var/lib/pgsql/data/postgresql.conf` file on the remote machine:
 From: `listen_addresses = 'localhost'`
 To: `listen_addresses = '*'`
4. Restart PostgreSQL.
5. Once again, using your local PostgreSQL client tools, try to verify that you can connect to the target remote PostgreSQL from the local installation machine, as described in

2.7 Specifying PostgreSQL Parameters

Accept the default location for the PostgreSQL `\bin` directory, or click **Browse** to locate and select another location. Click **Next**. You are 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 installation. You enter values for the first set of parameters in the table. The second set of parameters are used by the installer when using an existing installation of PostgreSQL:

Parameter	Default Value and Description
Binary Directory	The directory where the <code>postgres</code> and <code>pgAdmin3</code> binaries are located.
Port	The port number that PostgreSQL uses (default is 5432).
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: <code>postgres</code> . The installer cannot handle special characters at the end of a password string. Incompatible characters include: <code>&</code> ; <code>\$</code>
Defaults Used	Hardcoded Default Values Used or Created
PostgreSQL Administrative User Name	<code>postgres</code> - The default administrative database user.

Parameter	Default Value and Description
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.

2.8 Installing Sample Data

JasperReports Server can be installed with sample data that can help you evaluate its features. Sample data and resources 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, 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. Click **Yes** to install the sample data and click **Next**.

2.9 Installing iReport Designer

iReport Designer is the leading GUI-based JasperReports Library creation tool. It communicates directly with JasperReports Server and can retrieve an existing JasperReports Library from a JasperReports Server instance for editing, uploading, or execution.

In the installer, iReport comes pre-configured with a plug-in that allows it to communicate with JasperReports Server via the web services interface.

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

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

2.10 Completing the Installation

After the files have been installed, you will see the final installation screen. There are several post-installation options that you can choose from, each with its own check box. Simply click to make your choices then click **Finish**.

- **View Release Notes** - If you choose to view the Release Notes, they are displayed in a new window. If you are running from the command line, you can page through the Release Notes by pressing the Enter key.
- **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. There is a 25 second or so pause to allow the server to start up. When this pause is complete, 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](#).

Also, if you chose to view the Release Notes, JasperReports Server will not startup until you close the Release Notes.

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

You should now be ready to log into the server.

2.11 Post-Installation Steps

2.11.1 Updates Made by the Installer During Installation

This sub-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 to be sure 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 41 .
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.

2.11.2 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 for any potential errors. You can find the installer log at `<js-install>/installation.log`.

2.11.3 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 then your Java JVM options for heap memory allocation will be the following by default:

Installer Type	Setting	File Location
32 bit (x86) Windows	-Xms128M -Xmx512M	<js-install>/apache-tomcat/bin/service.bat
32 bit (x86) Linux	-Xms128M -Xmx512M	<js-install>/apache-tomcat/scripts/ctl.sh
64 bit (x64) Windows	-Xms1024m -Xmx2048m	js-install>/apache-tomcat/bin/service.bat
64 bit (x64) Linux	-Xms1024m -Xmx2048m	<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

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 4.2.1 > Start or Stop Services > Start Service**.
- Click **Start > All Programs > JasperReports Server CP 4.2.1 > 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

If you used your own existing installation for one of either Tomcat or PostgreSQL you can still use the Windows start/stop scripts mentioned in the previous section. The scripts would only start the bundled application that you chose to have the installer install.

For example, if you have an existing Tomcat and installed the bundled PostgreSQL, the scripts and menus specified in the previous section would only start and stop the PostgreSQL application. For 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 your database server.
- Second start your application server.

3.2 Start/Stop Scripts — Linux

Starting and stopping JasperReports Server is typically done at the Linux command line. The following commands are meant to be run 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 postgresql start|stop
./ctlscript.sh tomcat start|stop
```

3.3 Start/Stop Apps — Mac OSX

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

/Applications/jasperreports-server-cp-4.2.1

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 postgresql start|stop
./ctlscript.sh tomcat start|stop
```

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 here.

The login page appears.

4. Login using the following credentials:

User ID	Password	Description
jasperadmin	jasperadmin	Administrator for the default organization

If you installed the sample data, an additional sample end-user is also created. The non-administrative user has 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 joeuser. Leaving the default passwords and end-user 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 4.2.1 > 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-4.2.1/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

If you chose to install iReport during the JasperReports Server installation, start iReport as described in this section.

Windows:

Click **Start > All Programs > JasperReports Server CP 4.2.1 > 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.

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 Vista and Windows 7:

Click **Start > All Programs > JasperReports Server CP 4.2.1 > Uninstall JasperReports Server**.

To uninstall JasperReports Server on Windows XP:

1. Click **Start > All Programs > JasperReports Server CP 4.2.1 > 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-4.2.1
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, the JasperReports Server application is distributed as a stand-alone WAR file distribution. Community members who do not want to use the installer, or who have target configurations other than those supported by the installer, should use the WAR file distribution. The WAR file distribution comes in a file named `jasperreports-server-cp-4.2.1-bin.zip` in the compressed ZIP format. Download the WAR file distribution from our [community website](#).

This chapter contains the following sections:

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

5.1 Applications Supported by the WAR File Distribution

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

Database	App Server	Instructions Located In
MySQL PostgreSQL	Apache Tomcat JBoss GlassFish	This chapter.

For information on the specific versions of third party applications that are supported by the WAR file distribution ZIP refer to the release notes for the distribution you are using. The release notes are found in the root of the unpacked distribution ZIP.

5.2 Installing the WAR File Using Auto-install Scripts

Follow the steps in this procedure to install JasperReports Server using the WAR file distribution. These installation steps use the auto-install shell scripts to carry out the installation. The auto-install scripts are supported for Windows, Linux, and Mac OSX.

To meet prerequisites for installing the WAR file:

1. Install the Sun Java JDK 1.5 or 1.6. 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 6
- 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.11, “Glassfish Modifications,” on page 82](#):

4. Locate or install PostgreSQL or MySQL.



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

5. If you are using MySQL:
 - a. Download the JDBC driver, `mysql-connector-java-5.1.17-bin.jar` or later, from this web site:
<http://dev.mysql.com/downloads/connector/j/>
 PostgreSQL users can skip this step. JDBC drivers for PostgreSQL are included in the WAR file distribution.
 - b. Place the MySQL driver in `<js-install>/buildomatic/conf_source/db/mysql/jdbc`.
6. For use in [step 4](#) of the next procedure, make a note of this information: the DB username, DB password, and DB hostname for your database.

If you would like to run a pre-install validation test, you can run a command such as: `js-install-ce.bat test`. For more information, see section [5.5.3.1, “Auto-Install Script Test Mode,” on page 35](#).

To install the WAR file using auto-install scripts:

The scripts are intended for the bash shell.



Installing the bash shell on HP-UX, IBM AIX, FreeBSD, and Solaris is required for using the auto-install scripts.

1. Extract all files from `jasperreports-server-cp-4.2.1-bin.zip`. Choose a destination, such as Program Files on Windows, `/home/<user>` on Linux, and `/Applications` on Mac OSX.
 The directory, `jasperreports-server-cp-4.2.1-bin`, appears in the file location you chose.
2. Copy the `.properties` file for your database:
 - From — `<js-install>/buildomatic/sample_conf/`
 - 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 that are specific to your database and your 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, glassfish2, glassfish3, skipAppServerCheck] appServerDir=c:\Program Files\Apache Software Foundation\Tomcat 6* dbUsername=postgres dbPassword=postgres dbHost=localhost</pre>

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

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

* 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
- ♦ glassfish2
- ♦ glassfish3

5. Run the auto-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. Make <js-install>/buildomatic your current directory:


```
cd <js-install>/buildomatic
```
 - e. Run the following command:
 - ♦ Windows: js-install-ce.bat
 - ♦ Linux and Mac OSX: ./js-install-ce.sh

JasperReports Server, sample data, and sample databases (foodmart and sugarcrm) are installed.

Or, run this command if you don't want to include sample data and sample databases in the installation:

- ♦ Windows: js-install-ce.bat minimal
- ♦ Linux and Mac OSX: ./js-install-ce.sh minimal

If you encounter errors during the auto-install script execution, see section 5.5.3, “Error Running Auto-install Scripts (js-install-ce.bat/sh),” on page 35.

6. Set Java JVM Options, as described in 6.2, “Setting JVM Options for Application Servers,” on page 41. 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:

1. First, start your application server with one of the following commands:

Tomcat: <tomcat>/bin/startup.bat (Windows) *or* <tomcat>/bin/startup.sh (Linux and Mac OSX)

JBoss: <jboss>/bin/run.bat (Windows) *or* <jboss>/bin/run.sh (Linux and Mac OSX)

GlassFish: asadmin start-domain domain1

2. 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 26](#).

5.4 Logging into JasperReports Server

If JasperReports Server started up cleanly you should be able to login.

Login by going to the following URL:

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

Example:

`http://localhost:8080/jasperserver`

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

The login page should appear after taking some time to compile the necessary JSP file.

Use the following credentials to log into JasperReports Server:

User ID	Password	Description
jasperadmin	jasperadmin	Administrator for the default organization

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 weakens the security of your installation.

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

5.4.1 JasperReports Server Heartbeat

Upon first login to a newly installed JasperReports Server, you will be asked whether to opt-in to the JasperReports Server Heartbeat or not.

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

The heartbeat helps Jaspersoft create better products by improving our understanding of customer installation environments. If you choose to enable the heartbeat, at server startup time information like the following will be sent to Jaspersoft via an HTTPS call:

- 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 also 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" />
```

5.5 Troubleshooting Your JasperReports Server Configuration

This section describes the most common installation-related problems.

5.5.1 JasperReports Server Startup Problems

If you encounter a problem when trying to run a new JasperReports Server, an incorrect database configuration is the likely culprit. Another possibility is a mistake in the application server configuration files. For information on resolving these types of errors, refer to troubleshooting section [A.3, “Database Connectivity Errors,” on page 78](#).

5.5.2 Error Running a Report

If you have trouble running reports in your new JasperReports Server instance, refer to troubleshooting section [A.4, “Error Running a Report,” on page 80](#).

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

If you encounter errors during the auto-install script execution, look at the output log to see if you can spot any errors. Common errors include:

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

The auto-install 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, open the output log file.

The output log file is located in:

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

It is usually necessary to scroll back into the error message lines and see if there was an original error reported (that later caused additional errors). During an auto-installation of JasperReports Server, a Java based import operation is executed in order to put minimal and/or sample data in place.

Unfortunately, Java stack traces can be very long. Additionally, JasperReports Server uses the Spring framework to flexibly tie application components together. An error that involves a Spring initialization XML file can also cause a long stack trace.

Again, it will be important to try and isolate the first error encountered by the auto-installation steps.

Recreate your default_master.properties settings

If you need to make a correction to your default_master.property files, you can edit the file and re-run the js-install script. The js-install script always uses the most current values found in the default_master.properties file.

To help isolate errors, run the auto-install scripts can be run in test mode.

5.5.3.1 Auto-Install Script Test Mode

The auto-install (and auto-upgrade) scripts can be run in a test mode using the `test` option. In test mode, the js-install scripts will check your default_master.properties settings. The application server location will be validated and the capability to connect to the specified database will be 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.

Auto-upgrade scripts can also be run in test mode.

To run the auto-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 auto-install script in test mode:

```
js-install-ce.bat test
```

To run the auto-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 auto-install script in test mode:

```
./js-install-ce.sh test
```

5.5.4 Problem Connecting to a Cloud Database Instance

It is typical for a cloud database instance (such as Amazon EC2) to have all non-used IP ports disabled. When the auto-install script runs, it makes a validation call to the database hostname. This validation call uses the built-in ant operation `<isreachable>`. This operation is typically carried out similar to a network ping and may cause a hang issue if the port is not available. 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.

5.6 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 via ant targets used by buildomatic and located in the following directory:

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

This section describes the Ant targets and parameter setting you need to specify in order to send the standard options to the import and export commands.

5.6.1 Running Export from Buildomatic

The export target for ant has the following syntax:

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

The export file format can be a ZIP file or it can be a set of files under a new directory name. If you specify the `.zip` extension for your output filename then a ZIP archive will automatically be created. Otherwise, a directory with files and sub-directories will be created as a non-compressed 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 (|), which separates login user and organization names must be escaped with a backslash (\). In addition, when giving a list of user names, it must be enclosed in single quotation marks ('), as shown in the Linux example below:

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

5.6.2 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, it can contain more than one import option. On Linux, all double quotation marks (") must be escaped with a backslash (\).

The following examples on Windows are typical import commands:

```
js-ant import-help-cd
js-ant import-ce -DimportFile=my-reports.zip
js-ant import-ce -DimportFile=my-datasources -DimportArgs="--update"
```

The following examples on Linux are typical import commands:

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

5.6.3 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 11, “Configuring the Import-Export Utilities,” on page 73](#). For instance, you typically need to specify the correct JDBC driver for your database. As of JasperReports Server 4.2, the JDBC drivers are not in place for the MySQL database.

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
```

5.7 Manual Buildomatic Install Steps

The manual buildomatic steps described in the following procedure are the same Ant targets that are executed by the js-install scripts (js-install-ce.sh/.bat). The procedure below shows the buildomatic targets to manually execute if you are not able to use the js-install-ce 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 [Table 5-1 on page 32](#) for your database and application server. For instance, copy a sample properties file for your database type from <js-install>/buildomatic/sample_conf to the following location and file name (i.e., rename the file to default_master.properties):
<js-install>/buildomatic/default_master.properties
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 Instead of Using the js-install Scripts

Commands	Description
cd <js-install>/buildomatic	Go to the buildomatic directory
js-ant create-js-db	Creates the JasperReports Server repository database.
js-ant create-sugarcrm-db js-ant create-foodmart-db	(Optional) Creates the sample databases.
js-ant load-sugarcrm-db js-ant load-foodmart-db js-ant update-foodmart-db	(Optional) Loads sample data into the sample databases.
js-ant init-js-db-ce js-ant import-minimal-ce	Initializes database, loads core application data. Running js-ant import-minimal-ce is mandatory. The server cannot function without this data.
js-ant import-sample-data-ce	(Optional) Loads the demos that use the sample data.
js-ant deploy-webapp-ce	Configures and deploys the WAR file to Tomcat, JBoss, or Glassfish.



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.15, “Troubleshooting on Solaris,” on page 84](#).

If you encounter an error when running one of the create db targets (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 39](#).

5. Set Java JVM Options, as described in [6.2, “Setting JVM Options for Application Servers,” on page 41](#). This step is required.

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](#)
- [Additional Buildomatic Configuration Information](#)
- [Notes on the Hibernate Properties File](#)
- [Notes on Database Connections for Tomcat](#)
- [Notes on Data Source Definitions for JBoss](#)
- [Notes on Database Connections for Glassfish](#)
- [Report Scheduling Configuration](#)
- [Notes on Updating XML/A Connection Definitions](#)

6.1 Manually Creating the JasperReports Server Database

If you can't use the auto-install scripts to create the database, then you can create the database manually.

To manually create the JasperReports Server Database:

1. Enter the set of SQL commands for your particular database from one of the following sections:
 - [6.1.1, “PostgreSQL 8.x and 9.x,” on page 40](#)
 - [6.1.2, “MySQL,” on page 40](#)
- a. Enter the first set of commands to create and initialize the JasperReports Server database.
- b. Complete the installation by going to section [5.7, “Manual Buildomatic Install Steps,” on page 38](#) and start at the `js-ant import-minimal-ce` step.

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

6.1.1 PostgreSQL 8.x and 9.x

To manually create the JasperReports Server database in PostgreSQL 9.x:

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

```
cd <js-install>/buildomatic/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. Run the following commands if you want to install sample databases:

```
cd <js-install>/buildomatic/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-postgresql.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. Complete the installation by going to section [5.7, “Manual Buildomatic Install Steps,” on page 38](#) and start at the `js-ant import-minimal-ce` step.

6.1.2 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:

```
postgres=#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>/buildomatic/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. Run the following commands if you want 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-mysql.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. Complete the installation by going to section [5.7, “Manual Buildomatic Install Steps,” on page 38](#) and start at the `js-ant import-minimal-ce` step.

6.2 Setting JVM Options for Application Servers

JasperReports Server runs better with certain Java options for the JVM in which its application server is running. 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 Sun JVM. Other JVMs may or may not have equivalent settings.

6.2.1 Tomcat and JBoss JVM Options

JasperReports Server is supported on Java 1.5 and 1.6. If you are using Java 1.6, there are some additional JVM settings to avoid conflicts with JasperReports Server's AXIS-based web service classes. These conflicts could cause web services and the resources which rely on them to fail (such as XML/A connections). Similarly, JBoss 4.2 includes a web service that conflicts with AXIS-based web service classes and requires the same additional settings.

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 Windows	
Tomcat file	<tomcat>/bin/setenv.bat (or <tomcat>/bin/setclasspath.bat)
JBoss file	<jboss>/bin/run.bat
Options for Java 1.5 and 1.6	set JAVA_OPTS=%JAVA_OPTS% -Xms1024m -Xmx2048m -XX:PermSize=32m -XX:MaxPermSize=128m -Xss2m -XX:+UseConcMarkSweepGC -XX:+CMSCClassUnloadingEnabled
Additional options for Java 1.6 and JBoss 4.2	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 -Djavax.xml.transform.TransformerFactory=org.apache.xalan.processor.TransformerFactoryImpl

JVM Options on Linux and Mac OSX	
Tomcat file	<tomcat>/bin/setclasspath.sh or <tomcat>/bin/setenv.sh
JBoss file	<jboss>/bin/run.sh

JVM Options on Linux and Mac OSX, continued	
Options for Java 1.5 and 1.6	export JAVA_OPTS="\$JAVA_OPTS -Xms1024m -Xmx2048m -XX:PermSize=32m -XX:MaxPermSize=128m -Xss2m -XX:+UseConcMarkSweepGC -XX:+CMSClassUnloadingEnabled"
Additional options for Java 1.6 and JBoss 4.2	export 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 -Djavax.xml.transform.TransformerFactory=org.apache.xalan.processor.TransformerFactoryImpl"

Add your JAVA_OPTS setting directly below the following lines:

File	Add JVM Options Here
<tomcat>/bin/setclasspath.bat	set JAVA_ENDORSED_DIRS=%BASEDIR%\common\endorsed
<tomcat>/bin/setclasspath.sh	JAVA_ENDORSED_DIRS="\$BASEDIR"/common/endorsed
<tomcat>/bin/setenv.bat or <tomcat>/bin/setenv.sh	JAVA_OPTS setting can go anywhere in this file.
<jboss>/bin/run.bat <jboss>/bin/run.sh	set JAVA_OPTS=%JAVA_OPTS% -Dprogram.name=%PROGNAME% or export JAVA_OPTS="\$JAVA_OPTS -Dprogram.name=\$PROGNAME"

6.2.2 Bundled Tomcat as a Windows Service JVM Options

As of release 4.0, the bundled Tomcat application that is included with the Windows installer binary is installed as a Windows Service by default. Therefore, the steps required to change JVM options are different than they were in earlier installer versions.

The location where you will make the JVM edits is different. And after your edits are complete you will need to re-install the Tomcat service.

Here are the steps, for instance, to increase the Java heap values:

```
cd <js-install>/apache-tomcat/bin
```

Edit service.bat

Look for the following line (first line of two that set JVM options):

```
"%EXECUTABLE%" //US//%SERVICE_NAME% --Startup auto --JvmOptions "-Xms128M;-Xmx512M;  
-Xss2M;-Dcatalina.base=%CATALINA_BASE%;-Dcatalina.home=%CATALINA_HOME%;  
-Djava.endorsed.dirs=%CATALINA_HOME%\endorsed" --StartMode jvm --StopMode jvm
```

Update the line above to increase the Java heap:

```
-Xms1024M;-Xmx2048M
```

Because Tomcat is installed as a service, you will need to re-install the service. From a Windows cmd shell:

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

Note: After running each of the commands above (in Windows XP testing), the cmd shell was closed after the commands were executed. Also, note that the Tomcat service is removed and then installed. It is left in a running state after the INSTALL command is executed. You can make these updates while the services are running or not. But, you should stop and restart both

PostgreSQL and Tomcat after completing this work. You can use the normal JasperReports Server menu items to stop and start the services.

6.2.3 Existing Tomcat as a Windows Service JVM Options

If you installed to an existing Tomcat that is running as a Windows service, then you would typically add the Java options for JasperReports Server to the Java Tab of the Tomcat Properties dialog:

1. Launch the Tomcat configuration application from the Windows Start menu:
Start > Programs > Apache Tomcat > Configure Tomcat
2. In the Apache Tomcat Properties dialog, click the **Java** tab.
3. 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.
4. For instance, add as shown here:

```
-Xms1024m
-Xmx2048m
-XX:PermSize=32m
-XX:MaxPermSize=128m
-Xss2m
```

5. Click Apply, then click OK.

6.2.4 GlassFish JVM Options

For GlassFish, the JVM settings are identical for Java 1.5 and Java 1.6. The following sections show how to set the JVM options for GlassFish either through the command line or in a configuration file.

6.2.4.1 Setting GlassFish JVM Options with asadmin Command

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

```
asadmin create-jvm-options -Xms1024m:-Xmx2048m:-XX\:PermSize=32m:
-XX\:MaxPermSize=128m:-Xss2m:-XX\:+UseConcMarkSweepGC:
-XX\:+CMSClassUnloadingEnabled:-XX\:+CMSPermGenSweepingEnabled:
-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:
```

Now, restart the application server with the following commands:

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

When running the `asadmin create-jvm-options` command above, you may see some error messages such as the following:

```
[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.

6.2.4.2 Setting GlassFish JVM Options by Editing domain.xml

Open the <glassfish>/domains/domain1/config/domain.xml configuration file for editing, and add the following lines to the section entitled java-config:

```
<jvm-options>-Xms1024m -Xmx2048m -XX:PermSize=32m -XX:MaxPermSize=128m -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>
```

If you are modifying the settings for a running instance of GlassFish, you must restart the application server with the following commands:

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

6.3 Additional Buildomatic Configuration Information

The Ant-based buildomatic scripts contain files for the setup and configuration of a number of databases and application servers. The section gives you some pointers to the locations and content of some of these files.

6.3.1 Additional Commands for Buildomatic Settings

Whenever you change your default_master.properties file and re-run the auto-install scripts (or any other buildomatic target), your generated configuration settings are automatically updated. The generated settings are found 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 will clear the configuration template files found in buildomatic/build_conf/default directory. The second will re-build the configuration settings.



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

6.3.2 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.

You will find the generated property files in the following location:

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

Here are some of the key configuration files:

```
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:

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

In this directory you will find config files such as:

META-INF/context.xml

WEB-INF/hibernate.properties

WEB-INF/js.quartz.properties

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

6.3.3 Buildomatic: SQL Scripts Location

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

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

In the /sql directory, a directory representing each database contains key files. For example, in /postgresql the key files are:

<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/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.

6.3.4 Buildomatic: Database Creation Statements Location

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, 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

6.3.5 Buildomatic: JDBC Driver Locations

The installer provides the JDBC drivers for PostgreSQL shown in this table. 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
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

6.3.6 Buildomatic: Change your Deployed JDBC Driver

When you run the buildomatic target `deploy-webapp-ce` the JDBC driver for your specified database will be 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 the following example.

6.3.6.1 PostgreSQL 9.0-801 Driver Example

To change the deployed JDBC driver example for PostgreSQL:

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`.

To deploy other JDBC JAR files:

You can use the logic shown in the previous example to deploy other JDBC drivers. Then, make sure that the property settings and the JDBC driver name exactly match.

6.3.6.2 Location of Drivers

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 `jdb` folder location that is associated with your DB type:

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

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.

6.3.7 Buildomatic: JasperReports Server WAR File Location

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`.

6.3.8 Buildomatic: 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
```

6.4 Notes on the Hibernate Properties File

Your hibernate.properties settings will be found 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.

For example, this is a hibernate property values:

```
MySQL: metadata.hibernate.dialect=org.hibernate.dialect.MySQLDialect
```

6.5 Notes on Database Connections for 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, you should delete the file in this location:

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

6.6 Notes on Data Source Definitions for 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
```

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

```
<jboss>/server/default/deploy/js-jboss-ds.xml
```

6.6.1 Notes on Extra JBoss Configuration Steps

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.12, “JBoss Modifications,” on page 82](#).

6.7 Notes on Database Connections for 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"
```

6.8 Report Scheduling Configuration

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

6.8.1 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. In order to use this feature, you must 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: <code>smtp</code>	
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 <code>false</code> to <code>true</code> .

6.8.2 Database settings for the Quartz Driver Delegate Class

The Quartz driver delegate class is a class which Quartz uses to interact with the JDBC driver. For the MySQL database, you need to change the value of the quartz.delegateClass.



If you used buildomatic to install JasperReports Server, the correct value for your database is set automatically.

To set this value manually, edit the following file:

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

6.8.3 Settings for the Report Scheduler Web URI

For the web URI setting, the exact settings depend on what port your application server is running on and the name of your deployed jasperserver web application (that is, if you do not use the default name `jasperserver`).



If you installed using buildomatic these settings are handled automatically.

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>

6.8.4 Settings for Import-Export

If you are manually configuring the import-export shell scripts (that is, not using the buildomatic scripts), then depending on the database you are using, you will need to make sure your settings are correct for the Quartz driver delegate class property.



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 6.8.2, “ Database settings for the Quartz Driver Delegate Class ,” on page 49.

6.9 Notes on 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.

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 XMLA 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 XMLA Connection** resource.

CHAPTER 7 UPGRADING FROM 3.7.0 TO 4.2

This chapter describes the recommended procedure for upgrading from 3.7.0 to 4.2. If you are upgrading from 4.0.0 or 4.1.0 to 4.2, follow the procedure in [Chapter 8, “Upgrading from 4.0.0 or 4.1.0 to 4.2,” on page 61](#).

It is typical for JasperReports Server to have database schema updates with new major or minor releases. However, there have been no database changes between 4.0.0, 4.1.0, and 4.2. Because there have been no database changes between 4.x releases, it’s possible to upgrade from release 3.7.0 to release 4.2 in one set of steps.



To upgrade from 4.0.0 or 4.1.0 to 4.2 do not follow steps in this chapter. Use the steps described in [Chapter 8, “Upgrading from 4.0.0 or 4.1.0 to 4.2,” on page 61](#).

The upgrade procedure described in this chapter uses the JasperReports Server WAR File Distribution ZIP release package and the included buildomatic scripts.

This chapter contains the following sections:

- [Standard Upgrade Steps](#)
- [Backing Up the JasperServer 3.7 Instance](#)
- [Exporting 3.7.0 Repository Data](#)
- [Preparing the JasperReports Server 4.2 WAR File Distribution](#)
- [Configuring Buildomatic for Your Database and Application Server](#)
- [Upgrading to JasperReports Server 4.2](#)
- [Starting JasperReports Server 4.2](#)
- [Logging into JasperReports Server 4.2](#)
- [Additional Notes on JasperReports Server Upgrade](#)
- [Older Manual Upgrade Steps](#)

7.1 Standard Upgrade Steps

This section lists the standard upgrade steps. In general, follow these steps to install each new JasperReports Server release.

1. Back up your 3.7.0 JasperServer instance.
2. Export your 3.7.0 repository data.
3. Download and setup the 4.2 JasperReports Server WAR file distribution zip.
4. Run the auto-upgrade script.

If your instance of JasperServer 3.7.0 has any custom modifications or extensions, you will need to keep track of these and re-integrate them into your 4.2 instance after the upgrade is complete.

7.2 Backing Up the JasperServer 3.7 Instance

First, backup your JasperServer WAR file and your 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.

The following instructions are for the MySQL and PostgreSQL databases. For other databases, consult your DB administration documentation for back up information.

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

```
cd <tomcat>
mkdir js-3.7-war-backup
copy <tomcat>/webapps/ jasperserver to <tomcat>/js-3.7-war-backup
delete the <tomcat>/webapps/jasperserver directory
```

2. Back up the jasperserver database. Go to the location where you originally unpacked the 3.7.0 WAR file distribution zip or installed from the JasperServer 3.7.0 installer:

- a. Navigate to the location of your original unpacked 3.7.0 WAR file distribution.

```
cd <js-install-3.7>
```

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

- ♦ MySQL

Windows: `mysqldump --user=root --password=<password> jasperserver > js-db-3.7-dump.sql`

Linux: `mysqldump --user=root --password=<password> --host=127.0.0.1 jasperserver > js-db-3.7-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.3.4, “Maximum Packet Size in MySQL,” on page 79](#).

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 JasperServer installation.

- ♦ PostgreSQL

```
cd <js-install-3.7>
```

```
pg_dump --username=postgres jasperserver > js-db-3.7-dump.sql
```

7.3 Exporting 3.7.0 Repository Data

You will need to export your 3.7.0 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>/scripts` folder.

7.3.1 Export Using Buildomatic Scripts

If you have buildomatic and your default_master.properties file configured, you can export your 3.7.0 repository data using the following commands:

1. Navigate to the buildomatic directory:

```
cd <js-install-3.7.0>/buildomatic
```

2. Run buildomatic with the export target:

```
Windows: js-ant.bat export-everything-ce -DexportFile=js-3.7-export.zip
```

```
Linux:    ./js-ant export-everything-ce -DexportFile=js-3.7-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 point to it for the 4.2 upgrade.

7.3.2 Export Using the js-export Script

To use the js-export.bat/.sh script, you will move to the `<js-install-3.7.0>/scripts` folder. If you are using the MySQL 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 will need to update the js-export configuration. For information on configuring the 3.7 import-export utility see section [11.6, “Configuring the Import-Export Utility for JasperServer 3.7.0,” on page 76](#).

Run the following commands:

1. Navigate to the scripts directory:

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

2. Run the export script:

```
Windows: js-export.bat --everything-ce --output-zip js-3.7-export.zip
```

```
Linux:    js-export.sh --everything-ce --output-zip js-3.7-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 point to it for the 4.2 upgrade.

7.4 Preparing the JasperReports Server 4.2 WAR File Distribution

We will use the buildomatic auto-upgrade scripts included in the 4.2 WAR file distribution ZIP release package in order 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-4.2.1-bin.zip` in the compressed ZIP format. To download the WAR file distribution, go to JasperForge.org.
2. Extract all files from `jasperreports-server-4.2.1-bin.zip`. Choose a destination, such as Program Files on Windows, `/home/<user>` on Linux, and `/Applications` on Mac OSX.

This Guide refers to the path where you unpack the WAR File Distribution Zip as:

```
<js-install-4.2>
```

7.5 Configuring Buildomatic for Your Database and Application Server



There are no database changes between 4.0.0, 4.1.0, and 4.1.0. If you are upgrading from 4.0.0 or 4.1.0 to 4.2, use the steps described in [Chapter 8, “Upgrading from 4.0.0 or 4.1.0 to 4.2,” on page 61](#).

This upgrade procedure is based on using the buildomatic scripts which are included with the WAR File Distribution ZIP release package. The buildomatic scripts are based on the `ant` utility and require the Java Development Kit (JDK) to run. If you don't have Java available in your environment, see section [7.9.1, “Handling JasperReports Server Customizations,” on page 59](#).

Follow the configuration steps that match your database and application server in section [5.7, “Manual Buildomatic Install Steps,” on page 38](#). The scripts are intended for the bash shell.



Installing the bash shell on HP-UX, IBM AIX, and Solaris is required for using the buildomatic scripts.

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

7.5.1 Example Buildomatic Configuration

All upgrade configuration is handled by a single file that is named `default_master.properties`. 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`.

7.5.1.1 MySQL Example

This example uses MySQL (the same general logic applies to other databases).

You must rename and copy the sample file to this location: `<js-install-4.2>/buildomatic`.

1. Locate the `mysql_master.properties` file:

Database	Master Properties File
MySQL	<code><js-install-4.2>/buildomatic/sample_conf/mysql_master.properties</code>

2. Copy the file to `<js-install-4.2>/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>

7.5.1.2 PostgreSQL Example

This example uses PostgreSQL (the same general logic applies to other databases).

You must rename and copy the sample file to this location: `<js-install-4.2>/buildomatic`.

1. Locate the `postgresql_master.properties` file:

Database	Master Properties File
PostgreSQL	<code><js-install-4.2>/buildomatic/sample_conf/postgresql_master.properties</code>

2. Copy the file to <js-install-4.2>/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>

7.6 Upgrading to JasperReports Server 4.2



There are no database changes between 4.0.0, 4.1.0, and 4.2.0. If you are upgrading from 4.0.0 or 4.1.0 to 4.2 use the steps described in [Chapter 8, “Upgrading from 4.0.0 or 4.1.0 to 4.2,” on page 61](#).

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 JasperServer 3.7.0 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-4.2>/buildomatic</code>	
<code>js-upgrade-newdb-ce.bat <path>/js-3.7-export.zip</code>	(Windows) Upgrade jasperserver war file, drop and recreate database, import 3.7.0 data.
<code>./js-upgrade-newdb-ce.sh <path>/js-3.7-export.zip</code>	(Linux) Upgrade jasperserver war file, drop and recreate database, import 3.7.0 data.



You can use a fully specified path or a relative path. If the path you specify has spaces in it, you should use quotation marks: “<path>/js-3.7-export.zip”



On MySQL, if you receive an error about packet size, see section [A.3.4, “Maximum Packet Size in MySQL,” on page 79](#).

7.6.1 Auto-Upgrade Test Mode

The auto-upgrade scripts can be run in a test mode using the `test` option. For example, in Windows enter:

```
cd <js-install-4.2>/buildomatic
js-upgrade-newdb-ce.bat <path>/js-3.7-export.zip test
```

In test mode, the js-upgrade scripts will check your default_master.properties settings. The application server location will be validated and the capability to connect to the specified database will be 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.

7.6.2 Auto-Upgrade Script Help

You can run the auto-upgrade scripts with a help option, as shown in the following example:

```
js-upgrade-newdb-ce.bat --help
```

or

```
js-upgrade-newdb-ce.sh --help
```

7.6.3 Output Log Location

The auto-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
```

7.6.4 Errors

If you encounter errors during the auto-upgrade script execution, you should start by looking at the output log to see if you can spot any errors. Additionally, you should refer to the Troubleshooting section [Appendix A, “Troubleshooting,” on page 77](#). The information in this section applies to both auto-upgrade scripts and the auto-install scripts.

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

7.7 Starting JasperReports Server 4.2

You may now start your Tomcat, JBoss, or GlassFish application server. Your database should already be running.

7.8 Logging into JasperReports Server 4.2

If your application server and JasperReports Server 4.2 were started cleanly, you can now prepare to login.

7.8.1 Clearing Your Browser Cache

Before you log into 4.2, make sure and clear your Browser cache. JavaScript files, which enable UI elements of JasperReports Server, are typically cached by the Browser. The cache should be cleared to ensure that the most current files are used.

For JasperReports Server 4.2, the UI has been significantly enhanced. It will be very important to clear the browser cache.

Your end users should also be reminded to clear their Browser caches before logging in.

7.8.2 Logging into JasperReports Server

Login using the following URL, user ID, and password:

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

User ID	Password	Description
jasperadmin	<your-password>	Administrator for the default organization



If you updated your sample data in the sections above, your jasperadmin password might be reset to jasperadmin. You should change it as soon as possible.

Your JasperReports Server instance has now been upgraded to 4.2. If there are problems starting up or logging in, refer to troubleshooting section [A.3, “Database Connectivity Errors,” on page 78](#).

7.9 Additional Notes on JasperReports Server Upgrade

7.9.1 Handling JasperReports Server Customizations

If you made modifications or customizations to your JasperServer 3.7.0 application, 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 the older 3.7.0 environment and re-integrated into the new 4.2 environment.

7.9.2 Clearing the Application Server Work Directory

Application servers have work directories where JSP 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 is a good practice to double-check (in case a permission or other problem prevents the clearance of /work).

To clear the /work directory in Tomcat:

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

7.9.3 Clearing the Application Server Temp Directory

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 directory. It is a good practice to clear this /temp folder to avoid any conflicts after the upgrade is complete.

To clear the /temp directory in Tomcat, for instance, you would do the following:

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

7.9.4 Clearing the Repository Cache 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,” you should 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 this table, run a SQL command similar to the following:

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



You can clear your server repository cache manually using the above command (or a similar command).

7.9.5 Updating the XML/A Connections (Optional)

When you upgrade your sample data to 4.2, your XML/A connection sample data will be 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 XML/A connection may fail due to a mismatched password.

If you would like to update your XML/A connections, refer to section [6.9, “Notes on Updating XML/A Connection Definitions,”](#) on page 50.

7.9.6 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 steps as part of the upgrade process.

1. You will need to 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. Once this folder is deleted, you can deploy the new portlet WAR.

7.10 Older Manual Upgrade Steps

This section has some of the older, manual upgrade steps that were in place before the auto-upgrade shell scripts were implemented for JasperReports Server release 4.0.0. These are here as a reference. It is recommended that you use the auto-upgrade steps described at the beginning of this upgrade chapter.

Older buildomatic targets upgrade steps are:

Commands	Description
<code>cd <js-install-4.2>/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 your jasperserver db. Make sure it is backed up.
<code>js-ant import-minimal-ce</code>	
<code>js-ant import-upgrade</code> <code>-DimportFile="<path-and-filename>"</code>	The -DimportFile should point to the <path> and <filename> of the js-3.7-export.zip 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.
<code>js-ant import-sample-data-upgrade-ce</code>	This step is optional; it loads the 4.2 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 3.7.0 war file, deploy the 4.2 war file.



On MySQL, if you receive an error about packet size, see section [A.3.4, “Maximum Packet Size in MySQL,”](#) on page 79.

CHAPTER 8 UPGRADING FROM 4.0.0 OR 4.1.0 TO 4.2

This chapter covers how to upgrade JasperReports Server 4.0.0 or 4.1.0 to the Community Project version of JasperReports Server 4.2.



There are no database changes between 4.0.0, 4.1.0 and 4.2, so use this upgrade procedure for all database types. The procedure in this chapter is the recommended upgrade path.

This section uses an auto-upgrade shell script to carry out the upgrade. This script updates the JasperServer repository database resources with the new 4.2 resources and deploys the 4.2 WAR file to your application server.

This chapter contains the following sections:

- ♦ **Upgrade Steps**
- ♦ **Backing Up Your JasperReports Server 4.0.0 or 4.1.0 Instance**
- ♦ **Preparing the JasperReports Server 4.2 WAR File Distribution**
- ♦ **Configuring Buildomatic for Your Database and Application Server**
- ♦ **Upgrading to JasperReports Server 4.2**
- ♦ **Starting JasperReports Server 4.2**
- ♦ **Logging into JasperReports Server 4.2**
- ♦ **Additional Information on Post-Upgrade Steps**
- ♦ **Running Buildomatic DB Upgrade Steps Manually**

8.1 Upgrade Steps

These are the general steps used in this section.

1. Back up your 4.0.0 or 4.1.0 JasperServer instance.
2. Download and setup the 4.2 JasperReports Server WAR file distribution zip.
3. Run the auto-upgrade script.

If your instance of JasperServer 4.0.0 or 4.1.0 has any custom modifications or extensions, you will need to keep track of these and re-integrate them into your 4.2 instance after the upgrade is complete.

8.2 Backing Up Your JasperReports Server 4.0.0 or 4.1.0 Instance

First you must backup your JasperReports Server WAR file and your 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.

The following instructions are for the MySQL and PostgreSQL databases. For other databases, consult your DB administration documentation for back up information.

8.2.1 Backing Up Your JasperReports Server WAR File

Back up the jasperserver directory in Tomcat to a backup directory. (We will use 4.1.1 as the example.)

1. Go to the <tomcat> directory.
2. Make a new directory named js-4.1.1-war-backup.
3. Copy <tomcat>/webapps/ jasperserver to <tomcat>/js-4.1.1-war-backup.
4. Delete the <tomcat>/webapps/jasperserver directory.

8.2.2 Backing Up Your Jasperserver Database

Go to the location where you originally unpacked the 4.1.1 WAR file distribution zip. (We will use 4.1.1 as the example.)

1. Go to the <js-install-4.1.1> (the location of your original unpacked 4.1 WAR file distribution).
2. Run the following commands for either MySQL or PostgreSQL:

- ♦ MySQL

Windows: `mysqldump --user=root --password=<password> jasperserver > js-db-4.1.1-dump.sql`

Linux: `mysqldump --user=root --password=<password> --host=127.0.0.1 jasperserver > js-db-4.1.1-dump.sql`



If you receive an error about packet size, see section [A.3.4, “Maximum Packet Size in MySQL,”](#) on page 79.

- ♦ PostgreSQL

```
cd <js-install-4.1.1>
```

```
pg_dump --username=postgres jasperserver > js-db-4.1.1-dump.sql
```

8.3 Preparing the JasperReports Server 4.2 WAR File Distribution

We will use the buildomatic auto-upgrade scripts included in the 4.2 WAR file distribution ZIP release package in order to carry out the upgrade. Follow the steps in the sections listed below to obtain and unpack the WAR file distribution ZIP file:

1. The WAR file distribution comes in a file named jasperreports-server-cp-4.2.1-bin.zip in the compressed ZIP format. To download the WAR file distribution, go to JasperForge.org.
2. Extract all files from jasperreports-server-cp-4.2.1-bin.zip. Choose a destination, such as Program Files on Windows, /home/<user> on Linux, and /Applications on Mac OSX.

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

<js-install-4.2>

8.4 Configuring Buildomatic for Your Database and Application Server

This upgrade procedure is based on using the buildomatic scripts which are included with the WAR File Distribution ZIP release package. The buildomatic scripts are based on the ant utility and require the Java Development Kit (JDK) to run. If

you don't have Java available in your environment, see section [7.9.1, "Handling JasperReports Server Customizations," on page 59](#).

The scripts are intended for the bash shell.



Installing the bash shell on HP-UX, IBM AIX, and Solaris is required for using the buildomatic scripts.

Follow the configuration steps that match your database and application server in section [5.6, "Running the Import and Export Utilities," on page 36](#). The following section contains example configurations using the MySQL and PostgreSQL databases.

8.4.1 Example Buildomatic Configuration

All upgrade configuration is handled by a single file that is named `default_master.properties`. 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`.

8.4.1.1 MySQL Example

This example uses MySQL (the same general logic applies to other databases).

You must rename and copy the sample file to this location: `<js-install-4.2>/buildomatic`

1. Locate the `mysql_master.properties` file:

Database	Master Properties File
MySQL	<code><js-install-4.2>/buildomatic/sample_conf/mysql_master.properties</code>

2. Copy the file to `<js-install-4.2>/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-tomcat-6.0.26 (for example) dbUsername=root dbPassword=password dbHost=localhost</pre>

8.4.1.2 PostgreSQL Example

This example uses PostgreSQL (the same general logic applies to other databases).

You must rename and copy the sample file to this location: `<js-install-4.2>/buildomatic`

1. Locate the `postgresql_master.properties` file:

Database	Master Properties File
PostgreSQL	<code><js-install-4.2>/buildomatic/sample_conf/postgresql_master.properties</code>

2. Copy the file to `<js-install-4.2>/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	<code>appServerType=tomcat6 [tomcat7, tomcat6, jboss, glassfish2, glassfish3]</code> <code>appServerDir=c:\\apache-tomcat-6.0.26 (for example)</code> <code>dbUsername=postgres</code> <code>dbPassword=postgres</code> <code>dbHost=localhost</code>

8.5 Upgrading to JasperReports Server 4.2

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 JasperServer 4.0.0 or 4.1.0 WAR file before proceeding.

1. Stop your application server.
2. Start your database server.
3. Run these commands to upgrade the JasperReports Server 4.0.0 or 4.1.0 to the Community Project version JasperReports Server 4.2:

Commands	Description
<code>cd <js-install-4.2>/buildomatic</code>	
<code>js-upgrade-samedb-ce.bat</code>	Upgrade jasperserver war file, add 4.2 repository resources into the jasperserver database (for Windows)
<code>./js-upgrade-samedb-ce.sh</code>	Upgrade jasperserver war file, add 4.2 repository resources into the jasperserver database (for Linux)

8.5.1 Auto-upgrade Test

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

```
cd <js-install-4.2>/buildomatic
js-upgrade-samedb-ce.bat test
```

In test mode, the `js-upgrade` scripts will check your `default_master.properties` settings. The application server location will be validated and the capability to connect to the specified database will be 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.

8.5.2 Output Log Location

The auto-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
```

8.5.3 Errors

If you encounter errors during the auto-upgrade script execution, you should start by looking at the output log to see if you can spot any errors. Additionally, you should refer to the Troubleshooting section [Appendix A, “Troubleshooting,” on page 77](#). The information in this section applies to both auto-upgrade scripts and the auto-install scripts.

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

8.6 Starting JasperReports Server 4.2

You may now start your Tomcat, JBoss, or GlassFish application server. Your database should already be running.

8.7 Logging into JasperReports Server 4.2

If your application server and JasperReports Server 4.2 were started cleanly, you can now prepare to login.

8.7.1 Clearing Your Browser Cache

Before you log into 4.2, make sure and clear your browser cache. JavaScript files, which enable UI elements of JasperReports Server, are typically cached by the browser. The cache should be cleared to ensure that the most current files are used.

For JasperReports Server 4.2, the UI has been significantly enhanced. It will be very important to clear the browser cache.

Your end users should also be reminded to clear their browser caches before logging in.

8.7.2 Logging into JasperReports Server

You may now log into JasperReports Server using the same URL and credentials that you used before the upgrade.

8.8 Additional Information on Post-Upgrade Steps

After upgrading, clear the application server `/work` and `/temp` directories as described in this section. There is additional information on optional steps that can be done after the main upgrade steps are complete. See section [7.9, “Additional Notes on JasperReports Server Upgrade,” on page 59](#).

8.8.1 Clearing the Application Server Work Directory

Application servers have work directories where JSP 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 is a good practice to double-check (in case a permission or other problem prevents the clearance of `/work`).

To clear the `/work` directory in Tomcat:

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

8.8.2 Clearing the Application Server Temp Directory

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` directory. It is a good practice to clear this `/temp` folder to avoid any conflicts after the upgrade is complete.

To clear the `/temp` directory in Tomcat, for instance, you would do the following:

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

8.8.3 Clearing the Repository Cache 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,” you should 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 this table, run a SQL command similar to the following:

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



You can clear your jasperserver repository cache manually using the above command (or a similar command).

8.9 Running Buildomatic DB Upgrade Steps Manually

The auto-upgrade scripts (`js-upgrade-samedb.bat/.sh`) execute buildomatic Ant targets in order to carry out the upgrade. Here are the key buildomatic targets executed by the auto-upgrade scripts:

Commands	Description
<code>cd <js-install-4.2>/buildomatic</code>	
<code>js-ant upgrade-4.1-4.2-ce</code>	Execute SQL script for database upgrade to 4.2. Executes script <code>buildomatic/install_resources/sql/<dbType>/upgrade-<dbType>-4.1.0 - 4.2-ce.sql</code> (Note: This upgrade script is a dummy script, but this manual step is kept in place for consistency.)
<code>js-ant import-minimal-for-upgrade-ce</code>	Loads themes and other core resources for 4.2.
<code>js-ant import-sample-data-upgrade-ce</code>	(Optional) This step is optional. Loads the 4.2 sample data.
<code>js-ant deploy-webapp-ce</code>	Delete existing 4.0.0 or 4.1.0 war file, deploy 4.2 war file.

CHAPTER 9 UPGRADE NOTES FOR JASPERSERVER 3.7.0

If you are currently running JasperReports Server 3.7.0, then you can follow the set of steps in **Chapter 7, “Upgrading from 3.7.0 to 4.2,” on page 53** in order to upgrade to 4.2.

If you are running a JasperReports Server version earlier than 3.7.0, then you will first have to upgrade to 3.7.0 before you can upgrade to 4.2. In order to upgrade to 3.7.0, you should download the WAR File Distribution zip package for 3.7.0 (jasperserver-CE-3.7.0-bin.zip).

9.1 Upgrade from JasperServer 3.5.0

If your current instance is JasperServer 3.5.0, you must first upgrade to version 3.7.0 before upgrading to 4.2. The steps to carry out a 3.5.0 to 3.7.0 upgrade are documented in the JasperServer Installation Guide for the 3.7.0 release. You will need to download the JasperServer 3.7.0 release package to get the relevant files and documentation. To download the JasperServer 3.7.0 WAR file distribution zip package, go to the JasperForge.org downloads area.

CHAPTER 10 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, as well as 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.

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**

10.1 Backing Up Your JasperReports Server Database

As a precaution, you must back up your jasperserver database in case there is any 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
```

For MySQL databases, refer to your product documentation for details.

10.2 Stopping Your Application Server

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

10.3 Running the Repository Export Utility

The repository export utility writes out all of 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.bat --everything-ce --output-dir js-backup-catalog
Linux:   js-export.sh --everything-ce --output-dir js-backup-catalog
```

For information on running the export utility, refer to [Chapter 11, “Configuring the Import-Export Utilities,” on page 73](#).

10.4 Specifying Encryption Settings in the JasperReports Server WAR

JasperReports Server uses the Spring configuration and security to enable and configure encryption. These options can allow you to have a strong encryption setup. This section is focused on the minimal configuration necessary 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 the following:
`<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 the following:
`<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.

10.4.1 Specifying Encryption Settings - Reference Table

The information in the [Table 10-1](#) is a summary of the available password encryption configuration options described in the previous section.

Table 10-1 Password Encryption Configuration Options

Configuration File
...\WEB-INF\applicationContext-security.xml

Table 10-1 Password Encryption Configuration Options, continued

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.

10.5 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 [10.1, “Backing Up Your JasperReports Server Database,”](#) on page 69. 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 [10.4, “Specifying Encryption Settings in the JasperReports Server WAR,”](#) on page 70.

10.6 Recreating the JasperReports Server Database

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

10.6.1 Dropping and Recreating 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-ce-create.ddl  
\i quartz.ddl
```

10.6.2 Dropping and Recreating 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-ce-create.ddl;  
mysql>source quartz.ddl;
```

10.7 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.bat --input-dir js-backup-catalog`

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

For information on running the import utility, see [Chapter 11, “Configuring the Import-Export Utilities,”](#) on page 73.

10.8 Starting the Application Server

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

10.9 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 11 CONFIGURING THE IMPORT-EXPORT UTILITIES

The import and export utilities let you 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 in order to load the sample data into the repository.

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 3.7.0](#)

11.1 Introduction

The import-export functionality can be run using the auto-configured buildomatic Ant scripts or it can be run using the import-export shell scripts: `js-import.bat` and `js-export.bat` (Windows) or `js-import.sh` and `js-export.sh` (Linux and Mac OSX).

The following table contains examples of using the buildomatic Ant scripts and the import-export shell scripts:

Example Command	Description
<code>cd <js-install>/buildomatic</code>	Change to buildomatic folder
[buildomatic] <code>js-ant export-everything-ce -DexportFile=js-catalog-exp.zip</code>	Export using buildomatic Ant script
[shell script] <code>js-export.sh --everything-ce --output-file=js-catalog-exp.zip</code>	Export using import-export shell script

These two ways of running import-export commands are being merged so that all database configuration work will be done automatically by the buildomatic/default_master.properties. However, as of JasperReport Server version 4.2, this merging work is not complete. This means that if you are running the `js-import.sh/bat` or `js-export.sh/bat` shell scripts, you might have to do manual configuration depending on the database used.

In particular, the JDBC drivers are not in place for the MySQL database.

And you will need to make sure that there is a buildomatic/default_master.properties file in place before you run the `js-export` and `js-import` scripts.

The following sub-sections describe how to configure import-export if you are running from the shell scripts for different database types.



In Release 4.0, the `js-import.sh/bat` and `js-export.sh/bat` shell scripts have been moved to the `<js-install>/buildomatic` folder. (The old location was `<js-install>/scripts`.)

11.2 Import-Export Configuration Files

In the `buildomatic` folder, you will find the following files that make up the main parts of the import-export utility. These are the files to use or to modify to make configuration changes.

File or Location	Purpose
<code><js-install>/buildomatic/js-import.bat</code> and <code>.sh</code>	Import scripts for Windows and Linux, respectively
<code><js-install>/buildomatic/js-export.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

11.3 Changing Your Configuration Settings

If you are running the `js-import.bat/.sh` or `js-export.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 doing a manual installation from the WAR file distribution you must modify the `default_master.properties` configuration file to include your database settings and perform other configuration tasks.

11.3.1 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`.

Do the following:

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

11.3.2 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>buildomatic/conf_source/ieCe/lib`
Copy your DB type JDBC driver to this folder.

11.3.3 Checking the js.jdbc.properties File

Table 11-1 and **Table 11-2** contain sample settings for each database.

The `js.jdbc.properties` file will be automatically configured by buildomatic. 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`.

If your repository contains international characters, you may need to perform additional configuration for the import and export utilities. See section **A.9, “Exporting a Repository That Contains UTF-8,”** on page 81.

Table 11-1 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.MySQLDialect 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>

11.3.4 Checking the js.quartz.properties File

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

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

Table 11-2 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>

11.4 Deploying a JDBC Database Driver

In order for the import-export shell scripts to run, they will 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`

11.5 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.bat --help
         js-export.bat --help
Linux:   js-import.sh --help
         js-export.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, “Exporting a Repository That Contains UTF-8,” on page 81](#).

Please refer to the command line help for more information about command options for the import and export utilities.

11.5.1 Import-Export Access Events Option

Use the option `--include-access-events` to import or export of access event records stored in the JasperServer repository database.

11.6 Configuring the Import-Export Utility for JasperServer 3.7.0

You may need to configure your 3.7.0 import-export utility as to upgrade to 4.2.

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/or to handle database password changes you have made in your system, you would update the following files in the same manner described in the previous sections:

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

Additionally, you will need to copy the appropriate JDBC driver to the following folder (using the same copy information found in sections above):

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

APPENDIX A TROUBLESHOOTING

This appendix contains the following sections:

- **Installation Error Using PostgreSQL 8.1**
- **Installer Freezes**
- **Database Connectivity Errors**
- **Error Running a Report**
- **Database Error after Changing Database Port Number**
- **Case Sensitivity for Table and Column Names**
- **Java Out of Memory Error**
- **Error Running Scheduled Report**
- **Exporting a Repository That Contains UTF-8**
- **Importing Scheduled Jobs with Update Option**
- **Glassfish Modifications**
- **JBoss Modifications**
- **PostgreSQL: Job Scheduling Error**
- **Error Running Buildomatic Scripts**
- **Troubleshooting on Solaris**
- **Disabling User Session Persistence in Application Servers**
- **Linux Installer Issue with Unknown Host Error**
- **Problem Starting JasperReports Server on the Mac**
- **Session Error Using JasperReports Server and Tomcat 7**

A.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 auto-install or buildomatic scripts, there is no problem.

To workaround 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.2 Installer Freezes

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.3 Database Connectivity Errors

The most common problems encountered with a new JasperReports Server instance are database configuration problems. This section contains information that may help resolve such issues.

A.3.1 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.3.1.1 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.3.2 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 as described in A.3.3)
JBoss:   <jboss>/server/default/deploy/js-postgresql-ds.xml or js-oracle-ds.xml or js-<database name>-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.3.3 Context.xml under Tomcat: Special Case

If you deploy JasperServer to Tomcat multiple times, 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.

To change the context.xml file to fix a database problem:

1. Edit this `context.xml`:
`<tomcat>/webapps/jasperserver/META-INF/context.xml`
2. Delete the `jasperserver.xml` file:
`<tomcat>/conf/Catalina/localhost/jasperserver.xml`
 Deleting this file ensures that Tomcat uses the `context.xml` file

A.3.4 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.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 = 4M
```

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.4 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.
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.5 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.3.2, “Configuration File Locations,” on page 79](#).

A.6 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 Java Out of Memory Error

If you encounter a Java out of memory error, it is suggested that you increase your Java heap size setting. See section 5.3, **“Starting JasperReports Server,” on page 33**. As a minimum, add `-Xms128m -Xmx512m` 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 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.9 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 Error During JasperServer 1.2 Export

Upgrading typically requires doing an export operation on your database. If you are using MySQL and get a null pointer exception such as the following:

```
java.lang.NullPointerException
ResourceExporter.exportResource(ResourceExporter.java:258)
```

it may be due to an incorrect character in the file `js.jdbc.properties`. 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.10 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 workaround this problem, do not specify the `--update` option on your `js-import.bat/sh` command line.

A.11 Glassfish Modifications

A.11.1 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-ce` script in [step 5](#) in section [5.2, “Installing the WAR File Using Auto-install Scripts,” on page 31](#). 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.11.2 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. In order to allow the JVM options to be set, you will need to uncomment the `glassfishPort` property in the following buildomatic property file:

```
<js-install>/buildomatic/build_conf/default/app.srv.properties
```

from

```
#glassfishPort=4848
```

to

```
glassfishPort=4848
```

A.12 JBoss Modifications

A.12.1 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 41](#).

A.12.2 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

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.12.3 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.13 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
```

You should 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.14 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.14.1 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.

The following error referring to the tools.jar might occur:

```
[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 1.5 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.
```

To solve this problem, 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 1.5 JDK bundled in the previously installed version, as described in section [7.9.1, “Handling JasperReports Server Customizations,”](#) on page 59.

A.14.2 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.14.3 Older Apache Ant Version

As of the release of JasperReports Server 4.0, Apache Ant version 1.8.1 or higher is required. There are improvements to error handling routines in the buildomatic auto-install scripts which required the higher level of Ant. So, if you are using your own version of Ant, be sure that it is at this higher level:

```
ant -version
```

A.15 Troubleshooting on Solaris

When running the bundled Apache Ant scrips on the Solaris platform, you may see the following error:

```
ANT_HOME=../apache-ant: is not an identifier
```

The bundled Ant scripts are intended for the bash shell and may cause this error when run in the Bourne shell (sh). To avoid the error, run all js-ant targets in the bash shell explicitly, for example:

```
bash js-ant create-js-db
```

A.16 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.17 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
```

If you look inside of this log, or look at the error messages displayed on the console, and if you see errors such as the ones listed below, you might have the issue where the hosts file is not properly configured. Here are some errors you might see:

```
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!
```

In this case, you should fix the hosts issue and reinstall JasperReports Server.

The `/etc/hosts` file should normally have entries that look similar to the following:

```
127.0.0.1    localhost.localdomain    localhost
172.17.5.0   myserver.mydomain.com     myserver
```

A.18 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-cp-<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-cp-<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-cp-<ver>.
3. To Start:

```
./ctlscript postgresql start  
./ctlscript tomcat start
```
4. To shutdown:

```
./ctlscript stop
```

or

```
./ctlscript tomcat stop  
./ctlscript postgresql stop
```

A.19 Session Error Using JasperReports Server and Tomcat 7

On versions of Tomcat 7 before 7.0.19, a session error might occur while running reports. To workaround this problem, add the crossDomainSessionSecurity parameter to the dwr servlet in <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>debug</param-name>  
    <param-value>true</param-value>  
  </init-param>  
  <init-param>  
    <param-name>crossDomainSessionSecurity</param-name>  
    <param-value>>false</param-value>  
  </init-param>  
</servlet>
```