



JASPERREPORTS SERVER COMMUNITY PROJECT INSTALLATION GUIDE

RELEASE 4.1.0

<http://www.jasperforge.org>

Copyright © 2011 Jaspersoft Corporation. All rights reserved. Printed in the U.S.A. Jaspersoft, the Jaspersoft logo, JasperAnalysis, JasperServer, JasperETL, JasperReports, JasperReports Server, and iReport, 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 0611-JSO41-25 of the *JasperReports Server Installation Guide*.

TABLE OF CONTENTS

Chapter 1 Introduction	9
1.1 Conventions	10
1.2 Java Version Supported	10
1.3 JasperReports Server Distributions	10
1.4 Installer Distribution Support	11
1.4.1 Naming of 32-bit and 64-bit Installers	11
1.4.2 Installer Distribution Components	11
1.4.3 Installing with Existing Components	12
1.4.4 Running Components as Windows Services	12
1.5 WAR File Binary Distribution Support	12
1.6 Release Notes	13
1.7 Prerequisites for Installation	13
1.8 System Requirements	13
1.9 Support for Internationalization	14
Chapter 2 Installing JasperReports Server	15
2.1 Installation Steps	15
2.1.1 Welcome	16
2.1.2 Accepting the License Agreement	16
2.1.3 Choosing an Installation Directory	16
2.1.4 Selecting Tomcat Configuration	16
2.1.5 Selecting MySQL Configuration	16
2.1.6 Installing Sample Data	17
2.1.7 Installing Jaspersoft iReport Designer	17
2.1.8 Ready to Install	18
2.1.9 Installation Complete Screen	18
2.1.10 Logging into JasperReports Server	18
2.2 Post-Installation Steps	18
2.2.1 Updates Made by the Installer During Installation	18
2.2.2 Installer Output Log File Location	19

2.2.3	Checking your Java JVM Options	19
Chapter 3	Starting and Stopping JasperReports Server	21
3.1	Using the Start/Stop Menu With Windows	21
3.2	Using Start/Stop Scripts Without Bundled Installation with Windows	22
3.3	Start/Stop Scripts for Linux	22
3.4	Using Start/Stop Apps With Mac OSX	22
3.5	Logging into JasperReports Server	23
3.6	Starting the Included Jaspersoft iReport Designer	23
3.7	JasperReports Server Log Files	24
Chapter 4	Uninstalling JasperReports Server	25
4.1	Windows	25
4.2	Linux	25
4.3	Mac OSX	25
4.4	Uninstall Survey	26
Chapter 5	Installing the WAR File Distribution	27
5.1	Applications Supported by the WAR File Distribution	27
5.2	Obtaining the WAR File Distribution Zip	28
5.3	Unpacking the WAR File Distribution Zip	28
5.4	Introduction to Buildomatic Scripts	28
5.5	Pre-Installation Steps	28
5.5.1	Checking Your Java Installation	28
5.5.2	About Bundled Apache Ant	29
5.5.3	Checking Your Application Server	29
5.5.4	Checking Your Database Server	29
5.6	Configuring the Buildomatic Scripts	29
5.6.1	Creating your Default Master Properties File	30
5.7	Installing JasperReports Server	30
5.7.1	Windows Auto-Install	30
5.7.2	Linux Auto-Install	31
5.7.3	Auto-Install with Minimal Data	31
5.7.4	Output Log Location	31
5.7.5	Errors	31
5.8	Setting Java JVM Options	31
5.9	Starting JasperReports Server	32
5.10	Logging into JasperReports Server	32
5.10.1	JasperReports Server Heartbeat	32
5.11	Troubleshooting Your JasperReports Server Configuration	33
5.11.1	JasperReports Server Startup Problems	33
5.11.2	Error Running a Report	33
5.11.3	Error Running Auto-Install Scripts (js-install-ce.bat/sh)	33
5.11.4	Auto-Install Script "Test" Mode	33
5.11.5	Auto-upgrade Scripts	34

5.11.6	Auto-Install Script Output Log File	34
5.12	Running the Import and Export Utilities	34
5.12.1	Running Export from Buildomatic	34
5.12.2	Running Import from Buildomatic	35
5.12.3	Running the Import Export Shell Scripts	36
5.13	Pre-Test Validation mode of Auto-Install Scripts	36
5.13.1	Additional Command for Buildomatic Settings	36
5.14	Deploying to GlassFish	37
5.15	Manual Buildomatic Install Steps	37
Chapter 6	Additional Installation Information	39
6.1	Setting JVM Options for Application Servers	39
6.1.1	Tomcat and JBoss JVM Options	39
6.1.2	Bundled Tomcat as a Windows Service JVM Options	40
6.1.3	Existing Tomcat as a Windows Service JVM Options	41
6.1.4	GlassFish JVM Options	41
6.2	Additional Buildomatic Configuration Information	42
6.2.1	Buildomatic: Generated Property Files	42
6.2.2	Buildomatic: SQL Scripts Location	43
6.2.3	Buildomatic: Database Creation Statements Location	43
6.2.4	Buildomatic: JDBC Driver Locations	44
6.2.5	Buildomatic: Change your Deployed JDBC Driver	44
6.2.6	Buildomatic: JasperReports Server WAR File Location	45
6.2.7	Buildomatic: Sample Data Catalog ZIP Files	45
6.3	Additional Notes on Databases	45
6.3.1	Notes on the MySQL Database	45
6.4	Notes on the Hibernate Properties File	46
6.5	Notes on Database Connections for Tomcat	47
6.6	Notes on Data Source Definitions for JBoss	47
6.6.1	Notes on Extra JBoss Configuration Steps	47
6.7	Notes on Database Connections for GlassFish	48
6.8	Report Scheduling Configuration with Quartz	48
6.8.1	Mail Server Configuration Settings	48
6.8.2	Database settings for the Quartz Driver Delegate Class	49
6.8.3	Settings for the Report Scheduler Web URI	49
6.8.4	Settings for Import-Export	49
6.9	Notes on Updating XML/A Connection Definitions	50
Chapter 7	Upgrade from 3.7.0 to 4.1.0	51
7.1	Standard Upgrade Steps	51
7.2	Backing Up Your JasperServer 3.7.0 Instance	52
7.3	Export Your 3.7 Repository Data	52
7.3.1	Export Using Buildomatic Scripts	52
7.3.2	Export Using js-export Script	53
7.4	Preparing the JasperReports Server 4.1.0 WAR File Distribution	53

7.5	Configuring Buildomatic for Your Database and Application Server	53
7.5.1	Example Buildomatic Configuration (using MySQL)	53
7.6	Upgrading to JasperReports Server 4.1.	54
7.6.1	Output Log Location	54
7.6.2	Errors	54
7.7	Starting JasperReports Server 4.1.0	55
7.8	Logging into JasperReports Server 4.1.0	55
7.8.1	Clearing Your Browser Cache	55
7.8.2	Logging into JasperReports Server	55
7.9	Additional Notes on JasperReports Server Upgrade	55
7.9.1	Handling JasperReports Server Customizations	55
7.9.2	Clearing the Application Server Work Directory	55
7.9.3	Clearing the Application Server Temp Directory	56
7.9.4	Clearing the Repository Cache Table	56
7.9.5	Updating the XML/A Connections (Optional)	56
7.9.6	Upgrading the Liferay Portal	56
7.10	Older Manual Upgrade Steps	56
7.10.1	Manual Upgrade Steps	57
Chapter 8	Upgrade from 4.0.0 to 4.1.0	59
8.1	Upgrade Steps	59
8.2	Backing Up Your JasperServer 4.0.0 Instance	60
8.2.1	Backing Up Your JasperServer WAR File	60
8.2.2	Backing Up Your JasperServer Database	60
8.3	Preparing the JasperReports Server 4.1.0 WAR File Distribution	60
8.4	Configuring Buildomatic for Your Database and Application Server	60
8.4.1	Example Buildomatic Configuration (using MySQL)	60
8.5	Upgrading to JasperReports Server 4.1.0	61
8.5.1	Output Log Location	61
8.5.2	Errors	61
8.6	Starting JasperReports Server 4.1.0	62
8.7	Logging into JasperReports Server 4.1.0	62
8.7.1	Clearing Your Browser Cache	62
8.7.2	Logging into JasperReports Server	62
8.8	Additional Information on Post-Upgrade Steps	62
8.8.1	Clearing the Repository Cache Table	62
8.9	Running Buildomatic DB Upgrade Steps Manually	63
Chapter 9	Upgrade Notes for JasperServer 3.7.0	65
9.1	Upgrade from JasperServer 3.5.0	65
Chapter 10	Changing Password Encryption in JasperReports Server	67
10.1	Backing Up Your JasperReports Server Database	67
10.2	Stopping Your Application Server	68
10.3	Running the Repository Export Utility	68

10.4	Specifying Encryption Settings in the JasperReports Server WAR	68
10.4.1	Specifying Encryption Settings - Reference Table	68
10.5	Specifying Encryption Settings for the Import Utility	69
10.6	Recreating the JasperReports Server Database	69
10.6.1	Dropping and Recreating in MySQL	70
10.6.2	Dropping and Recreating in PostgreSQL	70
10.7	Importing Your Repository Data	70
10.8	Starting the Application Server	70
10.9	Logging into JasperReports Server	70
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	First Create a default_master.properties File	74
11.3.2	Here are the Configuration Locations	74
11.3.3	Check your js.jdbc.properties	75
11.3.4	If PostgreSQL Check your js.quartz.properties	75
11.4	Deploying a Database Driver	75
11.5	Running Import or Export	76
11.5.1	Import-Export Updates for 4.0	76
11.6	Configuring the Import-Export Utility for JasperServer 3.7.0	76
Appendix A	Troubleshooting	77
A.1	Installer Freezes	77
A.2	Database Connectivity Errors	78
A.2.1	Testing the Database Connection	78
A.2.2	Configuration File Locations	78
A.2.3	Context.xml under Tomcat: Special Case	78
A.2.4	Connect to Installed/Bundled Version of MySQL	79
A.2.5	Maximum Packet Size in MySQL	79
A.3	Error Running a Report	79
A.4	Database Error after Changing MySQL Port Number	80
A.5	Case Sensitivity for Table and Column Names	80
A.6	Java Out of Memory Error	80
A.7	Error Running Scheduled Report	80
A.8	Exporting a Repository That Contains UTF-8	80
A.8.1	Error During JasperServer 1.2 Export	80
A.9	Importing Scheduled Jobs with Update Option	81
A.10	JBoss Modifications	81
A.10.1	JBoss 4.2 XML/A Connection Fix	81
A.10.2	JBoss Large INFO Log Message on Analysis Drill-through	81
A.10.3	JBoss 4.0 Log4j Error on Startup	81
A.10.4	JBoss 5.0.1 and 5.1.x Error	82
A.11	PostgreSQL: Job Scheduling Error	82

A.12	Error Running Buildomatic Scripts	82
A.12.1	Missing Java JDK	82
A.12.2	Forgot to Copy the File ant-contrib.jar	83
A.12.3	Older Apache Ant Version	83
A.13	Troubleshooting on Solaris	83
A.14	Disabling User Session Persistence in Application Servers	83
A.15	Linux Installer Issue with Unknown Host Error	84
A.16	Problem Starting JasperReports Server on the Mac	84

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**
- **Java Version Supported**
- **JasperReports Server Distributions**
- **Installer Distribution Support**
- **WAR File Binary Distribution Support**
- **Release Notes**
- **Prerequisites for Installation**
- **System Requirements**
- **Support for Internationalization**

1.1 Conventions

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

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

1.2 Java Version Supported

JasperReports Server supports both Java 1.5 and Java 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.

1.4 Installer Distribution Support

The installers support the following operating systems:

Platform	Versions supported
Windows	XP Vista Windows 7
Linux	Red Hat Enterprise Linux SUSE Ubuntu And additional Linux distributions
Mac OSX	10.5 (Leopard) 10.6 (Snow Leopard)

1.4.1 Naming of 32-bit and 64-bit Installers

As of JasperReports Server 4.1.0, native 64-bit installers are supported. The 64-bit installer will put 64-bit versions of Java 6 and MySQL 5.1 onto your system for increased speed and performance.

The installer file naming has been updated for release 4.1.0 in order to distinguish the 32-bit installer from the 64-bit installer.

Installer Type	Naming
32-bit installer	jasperreports-server-cp-4.1.0-windows-x86-installer.exe jasperreports-server-cp-4.1.0-linux-x86-installer.run jasperreports-server-cp-4.1.0-osx-x86-installer.app.zip
64-bit installer	jasperreports-server-cp-4.1.0-windows-x64-installer.exe jasperreports-server-cp-4.1.0-linux-x64-installer.run jasperreports-server-cp-4.1.0-osx-x64-installer.app.zip
Note:	“x86” is shorthand referring to the 386, 486, and 586 CPU architecture.

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

1.4.2 Installer Distribution Components

The installer is designed so that it is easy to get JasperReports Server up and running quickly. The server requires the Java environment, an application server, and database to run. The installer distribution contains bundled versions of these components:

Component	Description
JasperReports Server Application	WAR file and configuration support scripts.
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 web application container.
MySQL Database	Database server. You can use the bundled version or an existing version. If using an existing MySQL, it must be on the local machine.

Component	Description
Apache Tomcat	Web application container: You can use the bundled version or an existing version. If using an existing Tomcat, it must be on the local machine.
JasperReports Server Documentation	Found in the <js-install>/docs directory.

1.4.3 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. For instance, if you already have Tomcat on your computer you can choose an “existing” Tomcat. If you would like the installer to install Tomcat for you, you can choose the “bundled” Tomcat. Both Apache Tomcat and the MySQL database can be independently used as bundled or existing instances. For information on the specific versions of third party applications that are supported by the installer refer to the JasperReports Server release notes for the distribution you are using. The release notes are found in the root of the installation directory.

Note: If you use an existing Tomcat or existing MySQL, they must be on the local machine.

1.4.4 Running Components as Windows Services

Starting with release 4.0, the Windows installer now installs MySQL 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” MySQL 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 MySQL and Tomcat services under the following names:

- ♦ jasperreportsMySQL
- ♦ jasperreportsTomcat

1.5 WAR File Binary Distribution Support

The WAR file binary distribution is the package you would use to do a manual installation of the JasperReports Server application. The WAR file supports many more applications than are supported by the installers. By using the WAR file to install the server, you can use a database other than MySQL and an application server other than Apache Tomcat.

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

Since version 4.0, there are “auto-install” scripts that will handle the installation tasks by putting your local settings in a single property 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.

Contents of the WAR file binary distribution are:

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

1.6 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.

1.7 Prerequisites for Installation

JasperReports Server relies on third-party products, such as application servers and relational databases. Unless you use the ones included with the installer, these third party products must be installed and configured before beginning an installation. Refer to the sections below that relate to your preferred application server and database.

1.8 System Requirements

The following table contains the minimum and recommended resources for a full installation, including MySQL 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	10 GB free	40 GB +
RAM		2 GB	4 GB +
Processor		1.5 GHz single Pentium, UltraSparc II, or equivalent	2.5 GHz + multi-core Pentium for Windows, Mac, and Linux

1.9 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 MySQL 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:

- [Installation Steps](#)
- [Post-Installation Steps](#)

2.1 Installation Steps

When you run the installation executable, you are prompted to specify information about the third party applications that JasperReports Server relies on. These third party applications are Apache Tomcat and the MySQL database.



When you run the installer using an existing database instance, the database must be running at install time. When you run the installer using an existing Apache Tomcat, it is recommended that the Tomcat instance be stopped,

To begin, run the installer. The installed application server and database will be on the same host.

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

```
jasperreports-server-cp-4.1.0-windows-x86-installer.exe    (32 bit)
jasperreports-server-cp-4.1.0-windows-x64-installer.exe    (64 bit)
```

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

```
./jasperreports-server-cp-4.1.0-linux-x86-installer.run    (32 bit)
./jasperreports-server-cp-4.1.0-linux-x64-installer.run    (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**.

On Mac OSX, the download file has a ZIP format. Typically, after download, the installer will be found in your <user>/Downloads folder. And it will already be unpacked. So, after the download is complete, double-click the following:

```
jasperreports-server-cp-4.1.0-osx-x86-installer.app        (32 bit)
jasperreports-server-cp-4.1.0-osx-x64-installer.app        (64 bit)
```

2.1.1 Welcome

The first step introduces the installer and allows you to continue or exit. Click **Next**.

Note: If you are installing a 32-bit installer on to 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.

2.1.2 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.

2.1.3 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-4.1.0

Linux: <USER_HOME>/jasperreports-server-cp-4.1.0

Mac OSX /Applications/jasperreports-server-cp-4.1.0

2.1.4 Selecting 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**.

2.1.5 Selecting MySQL Configuration

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

The first option is to choose a bundled MySQL. If you choose this option, the installer puts an instance of MySQL 5 onto your system. Click **Next**. The default MySQL port 3306 will be used. The installer will also create a MySQL database user with administrator privileges and credentials of jasperdb/password. If the installer finds that port 3306 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 MySQL configuration:

Parameter	Default Value and Description
Port	3306 - User must choose an alternate port if 3306 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 same password is used for the root database user.

The second option is to choose an existing MySQL. If you already have an instance of MySQL running on your local system, then you can choose this option. Choose the existing MySQL option and click **Next**. You are prompted for the location of MySQL, and the port to use. Note that the MySQL instance must reside on your local machine (i.e. localhost or 127.0.0.1). Enter the correct location for MySQL or click **Browse** to locate and select another location. Click **Next**. You are prompted for the root database account password of the MySQL root administrative user. The database administrative user account “root” is used by default. Enter the root database user password and click **Enter**.

The first set of values below are values to be entered. The second set is a description of values used by the installer if installing to an existing installation of MySQL:

Parameter	Default Value and Description
Binary Directory	The directory where the mysql and mysqladmin binaries are located.
Port	The port number that MySQL uses (default is 3306).
IP or Host Name	The value is hard coded to 127.0.0.1. Note that your existing MySQL instance must reside on the local machine.
MySQL Root Password	Password of the database administrative user: root. The installer cannot handle special characters at the end of a password string. Incompatible characters include: & ; \$
Defaults Used	Hardcoded Default Values Used or Created
MySQL Root User Name	The default administrative database user is “root”
Database User Name	jasperdb - The installer creates this database user which is used to connect to jasperserver database.
Database User Password	password - The installer creates this password for the jasperdb database user.



To improve system security, Jaspersoft recommends that you change all 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 MySQL to the mysql.user.password table.

2.1.6 Installing Sample Data

JasperReports Server can be installed with sample data that can help you evaluate its features. Sample data and resources included are the following:

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

2.1.7 Installing Jaspersoft iReport Designer

Jaspersoft iReport Designer (hereafter called “iReport”) is the leading GUI-based JasperReports Library creation tool. It has the capability of communicating directly with a JasperReports Server instance and can thus 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 click **Yes**.

2.1.8 Ready to Install

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

2.1.9 Installation Complete Screen

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 MySQL. 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 on logging in, see section 3.5, “[Logging into JasperReports Server,](#)” on page 23.



Checking the **Launch JasperReports Server Now** check box may or may not start the server, depending on your Tomcat and MySQL configuration choices. The 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 21.

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 5.10.1, “[JasperReports Server Heartbeat,](#)” on page 32.

You can later enable or disable the heartbeat by modifying the jasperserver/WEB-INF/applicationContext-logging.xml file.

2.1.10 Logging into JasperReports Server

You should now be ready to log into the server. For information on default login credentials, go to section 3.5, “[Logging into JasperReports Server,](#)” on page 23.



When you complete the evaluation or testing of your JasperReports Server instance, you should change the administrative passwords and remove any sample end-users. Leaving the default passwords and end-users weakens the security of your installation.

2.2 Post-Installation Steps

2.2.1 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/setclasspath.bat Linux: bin/setclasspath.sh	Modifies JAVA_OPTS to add -Djs.license.directory
Windows: bin/setenv.bat Linux: bin/setenv.sh	This file gets newly created. Sets increased Java memory allocation values to JAVA_OPTS. For additional settings, refer to section 5.8, “Setting Java JVM Options,” on page 31.
Tomcat 5: common/lib Tomcat 6: lib	Adds MySQL JDBC driver.

Updates made to the MySQL database

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

MySQL 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.

2.2.2 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>/install.log`.

For additional license configuration options, refer to sections 5.8, “Setting Java JVM Options,” on page 31.

2.2.3 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.

To check your Java JVM settings refer to section 5.8, “Setting Java JVM Options,” on page 31.

CHAPTER 3 STARTING AND STOPPING JASPERREPORTS SERVER

This chapter contains the following sections:

- **Using the Start/Stop Menu With Windows**
- **Using Start/Stop Scripts Without Bundled Installation with Windows**
- **Start/Stop Scripts for Linux**
- **Using Start/Stop Apps With Mac OSX**
- **Logging into JasperReports Server**
- **Starting the Included Jaspersoft iReport Designer**
- **JasperReports Server Log Files**

3.1 Using the Start/Stop Menu With Windows

If you chose to install a “bundled” Tomcat and a “bundled” MySQL, then you can use the Windows Start menu items to start and stop JasperReports Server.

To start or stop JasperReports Server, do the following:

- From the Windows Start menu:
Click **Start > All Programs > JasperReports Server CP 4.1.0 > Start or Stop Services > Start Service**.
Click **Start > All Programs > JasperReports Server 4.1.0 > Start or Stop Services > Stop Service**.
- Here is some extra information on the installed Tomcat and MySQL components. If you look at the Windows Services Tool Panel, you can see entries for these components which are installed as Windows Services.
Go to the Windows Services Tool Panel:
Look for: jasperreportsTomcat
Look for: jasperreportsMySQL
Note that by default, these components are started automatically. This means that if your system is rebooted, JasperReports Server will be automatically restarted. You can set these components to manual if you don’t wish them to automatically start.
- When JasperReports Server is running, in the Windows Task Manager you can see the following processes:
mysqlld.exe
tomcat6.exe

3.2 Using Start/Stop Scripts Without Bundled Installation with Windows

If you used your own existing installation for one of either Tomcat or MySQL 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 MySQL, the scripts and menus specified in the previous section would only start and stop the MySQL application. For the existing Tomcat, you would use the existing start and stop scripts provided by the Tomcat application.

3.3 Start/Stop Scripts for 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 mysql start|stop
./ctlscript.sh tomcat start|stop
```

3.4 Using Start/Stop Apps With Mac OSX

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

```
/Applications/jasperreports-server-4.1.0
```

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
```

Support for the Dock:

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

```
jasperServerStart.app
jasperServerStop.app
jasperServerLogin.app
```

When JasperReports Server is running, in the Activity Monitor, you can see the following processes:

```
java
mysqld.bin
```

Note: JasperReports Server under Mac can also be started and stopped from the Terminal shell command line using the same commands documented in the section above for Linux.

3.5 Logging into JasperReports Server



When you complete the evaluation or testing of your JasperReports Server instance, you should change the administrative passwords and remove any sample end-users. Leaving the default passwords and end-users weakens the security of your installation.

This section assumes that JasperReports Server is running. If it isn't, start it as described in the section above.

Log into JasperReports Server by entering the correct URL in your browser's address field and supplying the correct user name and password. JasperReports Server supports Firefox, Internet Explorer, Chrome, and Safari. The URL depends upon your application server. For the 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 when you connect to JasperReports Server.

Windows

Under Windows, you can launch the login page from the desktop of its host by clicking **Start > All Programs > JasperReports Server CP 4.1.0 > JasperReports Server Login**.

Mac OSX

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

`<js-install>/jasperServerLogin`

For example: `/Applications/jasperreports-server-4.1.0/jasperServerLogin`

Support for the Dock:

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

If the login page appears, JasperReports Server has started properly. You may now login with the following credentials. The default password should be changed for better system security.

User ID	Password	Description
jasperadmin	jasperadmin	Administrator for the default organization

If you installed the sample data, this sample end-user is also created. This user is a non-administrative user who has fewer system privileges than an administrative user. For better system security, delete this user when you stop using them:

User ID	Password	Description
joeuser	joeuser	Sample end-user

3.6 Starting the Included Jaspersoft iReport Designer

If you chose to install iReport as part of the JasperReports Server installation, you may start iReport from the Windows Start menu. To do this, click **Start > All Programs > JasperReports Server CP 4.1.0 > Start iReport Designer**.

For Mac OSX, from Finder, you can double-click the `iReport-mac.dmg` file found in the root of the JasperReports Server installation. This will bring up a new window where you can double-click the iReport Designer application.

To start on Mac OSX:

Double-click `iReport-mac.dmg`

Double-click Jaspersoft iReport Designer

To start on Linux:

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

3.7 JasperReports Server Log Files

Log files contain important information about how JasperReports Server is running. The log output goes to 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.

You can also change the logging levels while you are running JasperReports Server. Browse to the web page http://<hostname>:8080/jasperserver/log_settings.html. However, the settings on this page affect only the current session of JasperReports Server. Logging levels revert to their settings 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

Under Windows, click **Start > All Programs > JasperReports Server CP 4.1.0 > Uninstall JasperReports Server** to uninstall the server.

Note for Windows XP:

When uninstalling under Windows XP, it is typical to get a popup window that asks a question about which user account to run as. In this case, you will need to uncheck the check box that says “Protect my computer and data from unauthorized program activity”. Otherwise, the uninstaller will not execute.

4.2 Linux

Under Linux, the `<js-install>` folder includes an executable that removes JasperReports Server from the host. From the command line as the root user (or any user with sufficient privileges), enter:

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

You are prompted whether to remove JasperReports Server. On your keyboard, press Y then press Enter to remove JasperReports Server from this computer.

4.3 Mac OSX

Under Mac OSX, use Finder to go to the `<js-install>` folder and click the uninstall script. Do the following steps:

1. Navigate to the `<js-install>` folder.

For example: `/Applications/jasperreports-server-4.1.0`

2. Click the uninstall app to launch the uninstaller.

4.4 Uninstall Survey

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

CHAPTER 5 INSTALLING THE WAR FILE DISTRIBUTION

In addition to the installer binaries, the JasperReports Server application is distributed as a stand-alone WAR file distribution. This distribution is packaged as a ZIP file. Customers 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.

This chapter contains the following sections:

- **Applications Supported by the WAR File Distribution**
- **Obtaining the WAR File Distribution Zip**
- **Unpacking the WAR File Distribution Zip**
- **Introduction to Buildomatic Scripts**
- **Pre-Installation Steps**
- **Configuring the Buildomatic Scripts**
- **Installing JasperReports Server**
- **Setting Java JVM Options**
- **Starting JasperReports Server**
- **Logging into JasperReports Server**
- **Troubleshooting Your JasperReports Server Configuration**
- **Running the Import and Export Utilities**
- **Pre-Test Validation mode of Auto-Install Scripts**
- **Deploying to GlassFish**
- **Manual Buildomatic Install Steps**

5.1 Applications Supported by the WAR File Distribution

The instructions in this chapter support the following configurations:

Database	App Server
MySQL	Apache Tomcat
PostgreSQL	JBoss
	GlassFish

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 Obtaining the WAR File Distribution Zip

The WAR file distribution comes in a file named `jasperreports-server-cp-4.1.0-bin.zip` in the compressed ZIP format. To download the WAR file distribution, go to JasperForge.org.

5.3 Unpacking the WAR File Distribution Zip

Once you have downloaded the WAR file distribution, you need to unpack it in order to access the files it contains.

Choose a top level directory location to unpack the ZIP file to. The ZIP file creates the directory `jasperreports-server-cp-4.1.0-bin`.

Unpack to a directory such as Program Files or the root of the hard drive in Windows or your home directory in Linux. The resulting location given in the following table will be referred to as `<js-install>` in this document:

Operating System	Example Location	Referenced As
Windows	C:\Program Files\jasperreports-server-cp-4.1.0-bin or C:\jasperreports-server-cp-4.1.0-bin	<js-install>
Linux	/home/<user>/jasperreports-server-cp-4.1.0-bin or /opt/jasperreports-server-cp-4.1.0-bin*	<js-install>

* Default location for the root user.

5.4 Introduction to Buildomatic Scripts

The WAR file distribution contains a set of scripts known as the buildomatic scripts. These scripts automatically handle the configuration and deployment of JasperReports Server.

These scripts are found in the buildomatic directory. They rely on the Apache Ant build tool and the Java JVM for execution.

Since JasperReports Server release 4.0, there are “auto-install” shell scripts included that automate the installation steps. These scripts are located here:

`<js-install>/buildomatic/js-install-ce.bat`

`<js-install>/buildomatic/js-install-ce.sh`

5.5 Pre-Installation Steps

5.5.1 Checking Your Java Installation

JasperReports Server is a Java application that requires either Java 1.5 or Java 1.6. Earlier Java versions such as Java 1.4 will not work. You should check your installed Java version to see that it is at least Java 1.5. Additionally, JasperServer is certified to run with the Sun Java Development Kit (JDK). Unfortunately, there are known bugs currently with other Java implementations such as OpenJDK.

The buildomatic scripts are based on Apache Ant and they required the Java JDK. Therefore, you will need to verify that you have the Java Development Kit (JDK) installed and not merely the Java Runtime Environment (JRE). The JDK has additional tools and utilities required by Apache Ant.

You should also make sure that you have your `JAVA_HOME` variable set.

To check your Java version from the command line run:

```
java -version
```

5.5.2 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 scripts come with Windows and Linux batch scripts that are pre-configured to use the bundled version of Apache Ant. The buildomatic scripts are called from the command line in the following manner:

Windows: `js-ant <target-name>`

Linux: `./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.13, “Troubleshooting on Solaris,”](#) on page 83.

5.5.3 Checking Your Application Server

To run JasperReports Server you will need to have an application server installed. The buildomatic scripts support automatic deployment to the Tomcat, JBoss, and GlassFish application servers.

In the configuration step for the buildomatic scripts, you will need to specify the application server you are using and the application server's home directory. Valid values will be:

`tomcat5`

`tomcat6`

`jboss`

`glassfish2`

`glassfish3`

5.5.4 Checking Your Database Server

To run JasperReports Server you will need to have a database available. The buildomatic scripts support automatic installation to the MySQL and PostgreSQL databases. When running the buildomatic scripts or performing any manual installation, your database server should be running.

In the configuration step for the buildomatic scripts, you will need to supply, at a minimum, the DB username, DB password, and DB hostname information for your database.

5.6 Configuring the Buildomatic Scripts

Since JasperReports Server release 4.0, there are “auto-install” shell scripts included that utilize buildomatic to automate the installation steps. These scripts are located here:

`<js-install>/buildomatic/js-install-ce.bat`

`<js-install>/buildomatic/js-install-ce.sh`

You simply need to setup a properties file with your local settings.

5.6.1 Creating your Default Master Properties File

The buildomatic scripts read a file called `default_master.properties` in order to gather your application server path and your database settings. You must create the default master properties file from one of the database-specific sample files provided.

1. Copy the file for your database:

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

2. And rename the file to:
`<js-install>/buildomatic/default_master.properties`
3. Make the buildomatic directory your current directory.
`cd <js-install>/buildomatic`
4. Edit the `default_master.properties` file to add the settings that are specific to your database and your application server. The table below gives examples for each supported database. Be sure to replace the `appServerDir` property value with the path to your installed application server.

Database	Sample Property Values
MySQL	<pre>appServerType=tomcat6 [tomcat, jboss, glassfish, skipAppServerCheck] appServerDir=c:\\apache-tomcat-6.0.26 [for example] dbUsername=root dbPassword=password dbHost=localhost</pre>
PostgreSQL	<pre>appServerType=tomcat6 [tomcat, jboss, glassfish, skipAppServerCheck] appServerDir=c:\\apache-tomcat-6.0.26 [for example] dbUsername=postgres dbPassword=postgres dbHost=localhost</pre>

5.7 Installing JasperReports Server

Now that your `default_master.properties` file has been edited, you can install JasperReports Server.

Since JasperReports Server 4.0 there are Windows batch and Linux shell scripts that automatically handle the full installation of JasperReports Server. These are referred to as the “auto-install” scripts.



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 5.11.3, “[Error Running Auto-Install Scripts \(js-install-ce.bat/sh\)](#),” on page 33.

5.7.1 Windows Auto-Install

Make sure your database is running and run the following commands:

Commands	Description
<code>cd <js-install>/buildomatic</code>	Go to the buildomatic scripts directory.
<code>js-install-ce.bat</code>	Install JasperReports Server, create sample data and sample databases (foodmart and sugarcrm).

You may now proceed to the sections below on the Java JVM Options and Starting JasperReports Server. For troubleshooting, see section 5.11, “[Troubleshooting Your JasperReports Server Configuration](#),” on page 33.

5.7.2 Linux Auto-Install

Make sure your database is running and run the following commands:

Commands	Description
<code>cd <js-install>/buildomatic</code>	Go to the buildomatic scripts directory.
<code>js-install-ce.sh</code>	Install JasperReports Server, create sample data and sample databases (foodmart and sugarcrm).

You may now proceed to the sections below on the Java JVM Options and Starting JasperReports Server. For troubleshooting, see section 5.11, “[Troubleshooting Your JasperReports Server Configuration](#),” on page 33.

5.7.3 Auto-Install with Minimal Data

If you want to do a “minimal” installation that does not include any sample data, run the auto-install script with the `minimal` option (Windows or Linux):

Commands	Description
<code>cd <js-install>/buildomatic</code>	Go to the buildomatic scripts directory.
<code>js-install-ce.bat minimal</code>	Install JasperReports Server with minimal data (for Windows).
<code>js-install-ce.sh minimal</code>	Install JasperReports Server with minimal data (for Linux).

You may now proceed to the sections below on the Java JVM Options and Starting JasperReports Server.

5.7.4 Output Log Location

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, you can open the output log file.

The output log file is located here:

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

5.7.5 Errors

If you encounter errors during the auto-install 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 5.11, “[Troubleshooting Your JasperReports Server Configuration](#),” on page 33.

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

5.8 Setting Java JVM Options

JasperReports Server requires that your Java JVM runtime options be set to values larger than the typical Java default values.

For Tomcat and JBoss these values can be directly set by editing the shell scripts which start the application server. See section 6.1.1, “[Tomcat and JBoss JVM Options](#),” on page 39 for detailed steps.

For GlassFish, you can set the JVM options using the asadmin utility or by editing the domain.xml config directly. See section 6.1.4, “GlassFish JVM Options,” on page 41 for detailed steps.

5.9 Starting JasperReports Server

To run JasperReports Server start your application server with one of the following commands:

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

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

GlassFish: asadmin start-domain domain1

If your application server is set up as a Windows service, launch it through the Windows Start menu, for example **Start > All Programs > Apache Tomcat > ...**

To view the JasperReports Server application logs, see section 3.7, “JasperReports Server Log Files,” on page 24.

5.10 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 the system:

User ID	Password	Description
jasperadmin	jasperadmin	Administrator for the default organization

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

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



When you complete the evaluation or testing of your JasperReports Server instance, you should change the administrative password and remove any sample end-users. Leaving the default passwords and end-users weakens the security of your installation.

5.10.1 JasperReports Server Heartbeat

Upon first logging into 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 `jasperserver/WEB-INF/applicationContext-logging.xml` file. To disable, set the `enabled` property to `false` as shown below:

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

5.11 Troubleshooting Your JasperReports Server Configuration

5.11.1 JasperReports Server Startup Problems

When trying to run a new JasperReports Server instance, the most typical issue that users encounter are problems with the database configuration.

These problems are typically related to having incorrect configurations within the database configuration files or in the application server configuration files.

For more information on resolving these types of errors, refer to troubleshooting section [A.2, “Database Connectivity Errors,” on page 78](#).

5.11.2 Error Running a Report

If you have trouble running reports in your new JasperReports Server instance, refer to troubleshooting section [A.3, “Error Running a Report,” on page 79](#).

5.11.3 Error Running Auto-Install Scripts (`js-install-ce.bat/sh`)

If you received an error when you executed the `js-install-ce.bat` or `js-install-ce.sh` script, hopefully the error message was clear enough to allow you to make changes to eliminate the error. Common errors would be such things as typos in the path for the application server, or misspelling the hostname or password for the database.

The output log is found at:

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

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 make the edit and re-run the `js-install` script. The `js-install` script will always use the most current values found in the `default_master.properties` file.

To help isolate errors, the auto-install scripts can be run in “test” mode. See subsequent sections for details.

5.11.4 Auto-Install Script “Test” Mode

The auto-install scripts can be run in a “test” mode using the `test` option. This can help debug issues such as a mis-typed database password. Your system will not be altered when executing in test mode.

Run auto-install Script in Validation Test Mode

To run the auto-install scripts in test mode, do the following:

Commands	Description
<code>cd <js-install>/buildomatic</code>	Go to the buildomatic scripts directory
<code>js-install-ce.bat test</code>	Run auto-install script in test mode (Windows)
<code>js-install-ce.sh test</code>	Run auto-install script in test mode (Linux)

In test mode, the js-install scripts will check your settings to see if any setting can be determined to be incorrect. The application server location will be validated and the ability to connect to the specified database will be validated.

Note if 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.

5.11.5 Auto-upgrade Scripts

The same logic listed above for the “auto-install” scripts also applies to the “auto-upgrade” scripts. These scripts can also be run in test mode.

5.11.6 Auto-Install Script Output Log File

When an auto-install or auto-upgrade script is executed it will generate an output log. This log is located at:

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

5.12 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.

5.12.1 Running Export from Buildomatic

The `export` target for ant has the following syntax:

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

Linux: `./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 quotes (") and can contain more than one export option, as shown in the Windows examples below:

```
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 quotes (") and other characters such as the | separator for organization names must be escaped with a backslash (\). In addition, when giving a list of usernames, it must be enclosed in single quotes ('), 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'\"
```

5.12.2 Running Import from Buildomatic

The `import` target for ant has the following syntax:

```
Windows: js-ant import-ce -DimportFile=<filename> [-DimportArgs="<import-options>"]
Linux:    ./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 will be handled as a directory. The `importArgs` argument is optional, it can contain more than one import option. On Linux, all double quotes (") must be escaped with a backslash (\).

The following examples on Windows are typical import commands:

```
js-ant import-help-ce

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-ce

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

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

5.12.3 Running the Import Export Shell Scripts

The import-export shell scripts used to be found in the scripts folder. They are now found in the buildomatic folder.

You can find them here:

```
<js-install>/buildomatic/js-export.bat/.sh
<js-install>/buildomatic/js-import.bat/.sh
```

The options available from these scripts can be displayed by typing:

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

These scripts are shell scripts and are distinct from the “buildomatic import-export functionality” which is “auto-configured” (using the default_master.properties file).

To use these shell scripts, you must normally configure them yourself to, for instance, use the correct JDBC driver for your database. If use these shell scripts from a binary installer installation then they should already be properly configured.

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

5.13 Pre-Test Validation mode of Auto-Install Scripts

The auto-install scripts can run in a test (validation) mode. In this mode, no changes will be made to you system.

Commands	Description
<code>cd <js-install>/buildomatic</code>	Go to the buildomatic scripts directory
<code>js-install-ce.bat test</code>	Run a validation test of the JasperReports Server install (for Windows)
<code>js-install-ce.sh test</code>	Run a validation test of the JasperReports Server install (for Linux)

5.13.1 Additional Command 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).

5.14 Deploying to GlassFish

The GlassFish application server must first be running as described in [5.5.3, “Checking Your Application Server,” on page 29](#). This information is only for the case where you are not using the “auto-install” scripts.

Check that GlassFish is running then run:

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

When deploying to GlassFish with the previous command, the ant target will perform the following actions:

- JVM options will be set as described in section 5.8, “Setting Java JVM Options,” on page 31.
- The GlassFish application server will be stopped. You will restart the server in a subsequent step.

5.15 Manual Buildomatic Install Steps

These are the older manual target execution steps (before the implementation of the auto-install shell scripts in JasperReports Server version 4.0). You can use this section if you are not able to use the auto-install scripts.

Older manual buildomatic install target steps:

Commands	Description
<code>cd <js-install>/buildomatic</code>	Go to the buildomatic scripts directory
<code>js-ant create-js-db</code> <code>js-ant create-sugarcrm-db</code> <code>js-ant create-foodmart-db</code>	Creates the JasperReports Server and sample databases.
<code>js-ant load-sugarcrm-db</code> <code>js-ant load-foodmart-db</code> <code>js-ant update-foodmart-db</code>	(Optional) Loads sample data into the sample databases.
<code>js-ant init-js-db-ce</code> <code>js-ant import-minimal-ce</code>	Initializes database, loads core application data.
<code>js-ant import-sample-data-ce</code>	(Optional) Loads the demos that use the sample data.
<code>js-ant deploy-webapp-ce</code>	Configures and deploys the WAR file to Tomcat, JBoss, or Glassfish



On 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.13, “Troubleshooting on Solaris,” on page 83](#).

CHAPTER 6 ADDITIONAL INSTALLATION INFORMATION

This chapter contains the following sections:

- [Setting JVM Options for Application Servers](#)
- [Additional Buildomatic Configuration Information](#)
- [Additional Buildomatic Configuration Information](#)
- [Additional Notes on Databases](#)
- [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 with Quartz](#)
- [Notes on Updating XML/A Connection Definitions](#)

6.1 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.

6.1.1 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 analysis 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.

JVM Options on Windows	
Tomcat file	<tomcat>/bin/setenv.bat (or <tomcat>/bin/setclasspath.bat)
JBoss file	<jboss>/bin/run.bat

JVM Options on Windows, continued	
Options for Java 1.5 and Java 1.6	set JAVA_OPTS=%JAVA_OPTS% -Xms1024m -Xmx2048m -XX:PermSize=32m -XX:MaxPermSize=128m -Xss2m -XX:+UseConcMarkSweepGC -XX:+CMSClassUnloadingEnabled
Additional options for Java 1.6 or 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	
Tomcat file	<tomcat>/bin/setclasspath.sh or <tomcat>/bin/setenv.sh
JBoss file	<jboss>/bin/run.sh
Options for Java 1.5 and Java 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 or 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:

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
<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"
<tomcat>/bin/setenv.bat or <tomcat>/bin/setenv.sh	JAVA_OPTS setting can go anywhere in this file.

6.1.2 Bundled Tomcat as a Windows Service JVM Options

As of release 4.0.0, 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 MySQL and Tomcat after completing this work. You can use the normal JasperReports Server menu items to stop and start the services.

6.1.3 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.

6.1.4 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.

6.1.4.1 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.

6.1.4.2 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
```

6.2 Additional Buildomatic Configuration Information

The Ant-based buildomatic scripts contain support files that allow for the setup and configuration of a number of databases and application servers. Here are some pointers to the locations and content of some of these files.

6.2.1 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.)

6.2.2 Buildomatic: SQL Scripts Location

Buildomatic comes with SQL scripts and other utilities that support a number of databases. Here is where these files are found:

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

Here is an example of some of the key files (same pattern for additional databases):

```
<js-install>/buildomatic/install_resources/sql/mysql/js-create.ddl
<js-install>/buildomatic/install_resources/sql/mysql/js-drop.ddl
<js-install>/buildomatic/install_resources/sql/mysql/quartz.ddl
<js-install>/buildomatic/install_resources/sql/mysql/sugarcrm.zip
```



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

6.2.3 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, domains, and other data are stored. This database is normally named `jasperserver`.

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

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

```
<js-install>/buildomatic/conf_source/db/
mysql/scripts.properties
postgresql/scripts.properties
(same pattern for additional databases)
```

6.2.4 Buildomatic: JDBC Driver Locations

Buildomatic has default JDBC drivers for each supported database. These JDBC drivers are located here:

Database	Buildomatic JDBC Driver Location
MySQL	<js-install>/buildomatic/conf_source/db/mysql/jdbc/mysql-connector-java-5.1.10.jar
PostgreSQL	<js-install>/buildomatic/conf_source/db/postgresql/jdbc/postgresql-9.0-801.jdbc3.jar

To change the jdbc driver used, you can update your default_master.properties file and add the following information:

For example to specify a different default PostgreSQL driver:

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

The buildomatic scripts will automatically copy the appropriate JDBC driver to your application server when you run the `deploy-webapp-ce` target. Here are some typical locations where you can expect the JDBC driver to be copied:

```
Tomcat 5: <tomcat>/common/lib
Tomcat 6: <tomcat>/lib
JBoss:    <jboss>/server/default/lib
GlassFish: <glassfish>/domains/domain1/lib/ext
```

6.2.5 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.

However, you may find, for instance, that there is a different or more up to date JDBC driver that you would prefer to use. You can change the driver used by updating your default_master.properties file:

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

You will 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>
```

The buildomatic scripts will look in the default jdbc folder location that is associated with your DB type:

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

Change the deployed JDBC driver example for PostgreSQL:

Say, for example, you would like to use the PostgreSQL JDBC 4 driver. You would like this driver to be automatically deployed to your application server when you run the `deploy-webapp-ce` target. However, we can see by the settings in the following file that the default driver that will be used will be the following:

```
<js-install>/buildomatic/conf_source/db/postgresql/db.properties
maven.jdbc.artifactId=9.0
maven.jdbc.version=801.jdbc3
So the driver used is: postgresql-9.0-801.jdbc3.jar
```

To change the driver used, edit your default_master.properties file, add the following lines:

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

Now, when you run `deploy-webapp-ce`, the JDBC 4 driver will be used.

Note: For Import-Export: You should also manually update the JDBC driver version found in the `<js-install>/buildomatic/conf_source/ieCe/lib` folder. (This is only in the case where you plan to run the `js-import-ce.bat/.sh`, `js-export-ce.bat/.sh` shell scripts.)

To deploy other JDBC JAR files:

You can use the same logic above to deploy other JDBC drivers. You would put them in the same location as the existing JDBC drivers for your DB type. Then, make sure that the property settings and the JDBC driver name exactly match.

6.2.6 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/mysql/jdbc/mysql-connector-java-5.1.10.jar`
Copied to `<tomcat>/lib`

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

6.3 Additional Notes on Databases

This section provides additional information on the MySQL Database. There are commands to:

- Manually create and initialize the JasperReports Server database.
- Manually import the default users and organization.
- Manually create and load the sample databases.

6.3.1 Notes on the MySQL Database

The MySQL client software, `mysql.exe` or `mysql`, can be used to interact with the MySQL database. The example commands below have been tested at Jaspersoft. The commands to be used on your MySQL instance may be different.

These commands are run from the Windows or Linux command line.

6.3.1.1 Manual Creation of the JasperReports Server Database

Please check your database user documentation for how to set up a database and how to create a database user.

Run the following commands:

```
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@localhost identified by 'password';
mysql>flush privileges; (reload privilege tables)
mysql>use jasperserver;
mysql>source js-create.ddl
mysql>source quartz.ddl
mysql>exit
```



If you are going to access MySQL on a remote server you should run an additional grant statement:

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

6.3.1.2 Manual Import of Default Minimal Data

First, follow the instructions in [Chapter 11, “Configuring the Import-Export Utilities,”](#) on page 73.

Note: The commands shown below can be run from the buildomatic ant import task instead.

Then run the following commands to perform the import:

```
cd <js-install>/buildomatic
js-import-ce --input-zip ../buildomatic/install_resources/export/js-catalog-mysql-
minimal-ce.zip
```

The next (optional) command loads the sample data resources. No need to run this command if you do not create and load the sample databases from the next section.

```
js-import-ce --input-zip ../buildomatic/install_resources/export/js-catalog-mysql-ce.zip
```

6.3.1.3 Manual Creation of Sample Databases

Run the following commands:

```
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;
mysql>use foodmart;
mysql>source foodmart-mysql.sql; (first make sure the file is unzipped)
mysql>exit
```

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.

Here is an example hibernate property value:

```
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.10, “JBoss Modifications,” on page 81](#).

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 with Quartz

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
<code>report.scheduler.mail.sender.host</code>	The name of the computer hosting the mail server.
<code>report.scheduler.mail.sender.username</code>	The name of the user in the mail server that JasperReports Server can use.
<code>report.scheduler.mail.sender.password</code>	The password of the mail server user.
<code>report.scheduler.mail.sender.from</code>	The address that appears in the From field on email notifications.
<code>report.scheduler.mail.sender.protocol</code>	The protocol that the mail server uses. JasperReports Server only supports SMTP. Note: Your entry must be lower case. For example: <code>smtp</code>
<code>report.scheduler.mail.sender.port</code>	The port number that the mail server uses. For SMTP, the default is typically 25 (values other than 25 may not work in earlier JasperServer versions).

Configuration File		
<app-server>/<deployment>/WEB-INF/applicationContext-report-scheduling.xml		
Property	Bean	Description
javaMailProperties key="mail.smtp.auth"	reportScheduler MailSender	If your mail server requires authentication, change this property from false to true.

6.8.2 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 PostgreSQL database it needs a non-default setting.



If you installed using buildomatic these settings are handled 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	MySQL	org.quartz.impl.jdbcjobstore.StdJDBCDelegate
	PostgreSQL	org.quartz.impl.jdbcjobstore.PostgreSQLDelegate

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:

Configuration File		
<app-server>/<deployment>/WEB-INF/js.quartz.properties		
Property	App Server	Example Value
report.scheduler. web.deployment.uri	Apache Tomcat	http://localhost:8080/jasperserver
	JBoss	http://localhost:8080/jasperserver
	GlassFish	http://localhost:8080/jasperserver

To manually edit, you would edit the following file:

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 and the Quartz table prefix 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:

Configuration File	
<js-install>/buildomatic/conf_source/ieCe/js.quartz.properties	
Property	Description
quartz.delegateClass	Set to the same value as described in section 6.8.2, “ Database settings for the Quartz Driver Delegate Class ,” on page 49.
quartz.tablePrefix	Set to the same value as described in section 6.8.4, “ Settings for Import-Export ,” 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.

If you have Jaspersoft OLAP enabled (via your license), 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 UPGRADE FROM 3.7.0 TO 4.1.0

It is typical for JasperReports Server to have database schema updates with new major or minor releases. However, in the case of release 4.1.0, there are no database changes between 4.0.0 and 4.1.0.

Because of this, the best procedure to upgrade from 4.0.0 to 4.1.0 is the procedure described in [Chapter 8, “Upgrade from 4.0.0 to 4.1.0,” on page 59](#).

Because there are no database changes, it is also possible to upgrade from release 3.7.0 to release 4.1.0 in one set of steps. This chapter describes the recommended procedure for upgrading from 3.7.0 to 4.1.0.



There are no database changes between 4.0.0 to 4.1.0. To upgrade from 4.0.0 to 4.1.0 use the steps described in [Chapter 8, “Upgrade from 4.0.0 to 4.1.0,” on page 59](#).

These steps use the JasperReports Server WAR File Distribution ZIP release package and the included buildomatic scripts for the upgrade procedure.

This chapter contains the following sections:

- [Standard Upgrade Steps](#)
- [Backing Up Your JasperServer 3.7.0 Instance](#)
- [Export Your 3.7 Repository Data](#)
- [Preparing the JasperReports Server 4.1.0 WAR File Distribution](#)
- [Configuring Buildomatic for Your Database and Application Server](#)
- [Upgrading to JasperReports Server 4.1.](#)
- [Starting JasperReports Server 4.1.0](#)
- [Logging into JasperReports Server 4.1.0](#)
- [Additional Notes on JasperReports Server Upgrade](#)
- [Older Manual Upgrade Steps](#)

7.1 Standard Upgrade Steps

This section lists the standard upgrade steps. These general steps are always available with 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 4.1.0 JasperReports Server WAR file distribution zip.

4. Run 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 4.1.0 instance after the upgrade is complete.

7.2 Backing Up Your JasperServer 3.7.0 Instance

First you must 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 database. 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. `cd <js-install-3.7>` (the location of your original unpacked 3.7.0 WAR file distribution)
- b. Run the following command:

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.2.5, “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.

7.3 Export Your 3.7 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. One is using the buildomatic scripts (if you originally installed using buildomatic). Or you will 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. `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 4.1.0 upgrade.

7.3.2 Export Using js-export Script

To use the `js-export.bat/.sh` script, you will move to the `<js-install-3.7>/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. `cd <js-install-3.7>/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 4.1.0 upgrade.

7.4 Preparing the JasperReports Server 4.1.0 WAR File Distribution

We will use the buildomatic scripts included in the 4.1.0 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. Follow steps in section 5.2, “[Obtaining the WAR File Distribution Zip,](#)” on page 28.
2. Follow steps in section 5.3, “[Unpacking the WAR File Distribution Zip,](#)” on page 28.

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

`<js-install-4.1.0>`

7.5 Configuring Buildomatic for Your Database and Application Server



There are no database changes between 4.0 to 4.1. If you are upgrading from 4.0.0 to 4.1.0, use the steps described in [Chapter 8, “Upgrade from 4.0.0 to 4.1.0,”](#) on page 59 instead.

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.

Follow the configuration steps that match your database and application server in section 5.6, “[Configuring the Buildomatic Scripts,](#)” on page 29. Below is an example configuration using the MySQL database.

7.5.1 Example Buildomatic Configuration (using MySQL)

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`.

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

You must rename and copy the sample file to this location: `<js-install-4.1>/buildomatic`.

1. Locate the `mysql_master.properties` file:

Database	Master Properties File
MySQL	<code><js-install-4.1>/buildomatic/sample_conf/mysql_master.properties</code>

2. Copy the file to `<js-install-4.1>/buildomatic`.

3. Rename the file default_master.properties.
4. Edit default_master.properties for your database and application server:

Database	Sample Property Values
MySQL	appServerType=tomcat6 (or tomcat, jboss, glassfish) appServerDir=c:\\apache-tomcat-6.0.26 (for example) dbUsername=root dbPassword=password dbHost=localhost

7.6 Upgrading to JasperReports Server 4.1.



There are no database changes between 4.0.0 to 4.1.0. If you are upgrading from 4.0.0 to 4.1.0 use the steps described in [Chapter 8, “Upgrade from 4.0.0 to 4.1.0,” on page 59](#) instead.

Now that your buildomatic scripts have been configured, you can complete the upgrade. Run the following commands:



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.

Commands	Description
<code>cd <js-install-4.1>/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, data, 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.2.5, “Maximum Packet Size in MySQL,” on page 79](#).

7.6.1 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
```

7.6.2 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 [5.11, “Troubleshooting Your JasperReports Server Configuration,” on page 33](#). 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 4.1.0

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

7.8 Logging into JasperReports Server 4.1.0

If your application server and JasperReports Server 4.1.0 were started cleanly, you can now prepare to login.

7.8.1 Clearing Your Browser Cache

Before you log into 4.1, 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 4.1, 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 4.10. If there are problems on startup or login refer to troubleshooting section [A.2, “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 4.1.0 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. Before you update to a new WAR file or license, the `/work` directory should be cleared. The buildomatic `deploy-webapp` target automatically clears the application server's `work` directory but it is a good practice to double-check (in case of permission issues, etc).

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

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 4.1.0, 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 from the top of this upgrade chapter.

7.10.1 Manual Upgrade Steps

Older buildomatic targets upgrade steps:

Commands	Description
<code>cd <js-install-4.1>/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>	<p>The -DimportFile should point to the <path> and <filename> of the js-3.7-export.zip file you created earlier.</p> <p>On Windows, you must use double quotes (") if your path or filename contains spaces. On Linux, you must use double quotes escaped with a backslash (\") in this case.</p>
<code>js-ant import-sample-data-upgrade-ce</code>	This step is optional; it loads the 4.1.0 sample data. The old sample data is overwritten, so you may need to redo certain changes such as configuring the sample data sources for your database.
<code>js-ant deploy-webapp-ce</code>	Delete existing 3.7.0 war file, deploy 4.1.0 war file.



On MySQL, if you receive an error about packet size, see section [A.2.5, “Maximum Packet Size in MySQL,”](#) on page 79.

CHAPTER 8 UPGRADE FROM 4.0.0 TO 4.1.0

This chapter covers how to upgrade a JasperReports Server 4.0.0 to JasperReports Server 4.1.0



There are no database changes between 4.0.0 and 4.1.0 So this upgrade procedure will work 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 will update your JasperServer repository database resources with the new 4.1 resources, and it will deploy the 4.1 WAR file to your application server.

This chapter contains the following sections:

- **Upgrade Steps**
- **Backing Up Your JasperServer 4.0.0 Instance**
- **Preparing the JasperReports Server 4.1.0 WAR File Distribution**
- **Configuring Buildomatic for Your Database and Application Server**
- **Upgrading to JasperReports Server 4.1.0**
- **Starting JasperReports Server 4.1.0**
- **Logging into JasperReports Server 4.1.0**
- **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 JasperServer instance.
2. Download and setup the 4.1.0 JasperReports Server WAR file distribution zip.
3. Run auto-upgrade script.

If your instance of JasperServer 4.0.0 has any custom modifications or extensions, you will need to keep track of these and re-integrate them into your 4.1 instance after the upgrade is complete.

8.2 Backing Up Your JasperServer 4.0.0 Instance

First you must 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 database. For other databases, consult your DB administration documentation for back up information.

8.2.1 Backing Up Your JasperServer WAR File

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

1. Go to the <tomcat> directory.
2. Make a new directory named js-4.0.0-war-backup.
3. Copy <tomcat>/webapps/ jasperserver to <tomcat>/js-4.0.0-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 4.0.0 WAR file distribution zip.

1. Go to the <js-install-4.0.0> (the location of your original unpacked 4.0.0 WAR file distribution).
2. Run the following command:

Windows: `mysqldump --user=root --password=<password> jasperserver > js-db-4.0.0-dump.sql`

Linux: `mysqldump --user=root --password=<password> --host=127.0.0.1 jasperserver > js-db-4.0.0-dump.sql`



If you receive an error about packet size, see section [A.2.5, “Maximum Packet Size in MySQL,”](#) on page 79.

8.3 Preparing the JasperReports Server 4.1.0 WAR File Distribution

We will use the buildomatic scripts included in the 4.1 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. Follow steps in section [5.2, “Obtaining the WAR File Distribution Zip,”](#) on page 28.
2. Follow steps in section [5.3, “Unpacking the WAR File Distribution Zip,”](#) on page 28.

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

<js-install-4.1>

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.

Follow the configuration steps that match your database and application server in section [5.6, “Configuring the Buildomatic Scripts,”](#) on page 29. Below is an example configuration using the MySQL database.

8.4.1 Example Buildomatic Configuration (using MySQL)

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`.

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

You must rename and copy the sample file to this location: `<js-install-4.1.0>/buildomatic`.

1. Locate the `mysql_master.properties` file:

Database	Master Properties File
MySQL	<code><js-install-4.1>/buildomatic/sample_conf/mysql_master.properties</code>

2. Copy the file to `<js-install-4.1.0>/buildomatic`.
3. Rename the file to `default_master.properties`.
4. Edit `default_master.properties` for your database and application server:

Database	Sample Property Values
MySQL	<pre>appServerType=tomcat6 (or tomcat5, jboss, glassfish) appServerDir=c:\\apache-tomcat-6.0.26 (for example) dbUsername=root dbPassword=password dbHost=localhost</pre>

8.5 Upgrading to JasperReports Server 4.1.0

Now that your buildomatic scripts have been configured, you can complete the upgrade. Run these commands to upgrade JasperReports Server 4.0.0 to JasperReports Server 4.1.0:



Make sure you have backed up your `jasperserver` database before proceeding.

Make sure you have backed up your JasperServer 4.0.0 WAR file before proceeding.

Commands	Description
<code>cd <js-install-4.1.0>/buildomatic</code>	
<code>js-upgrade-samedb-ce.bat</code>	(Windows) Upgrade jasperserver war file, add 4.1.0 repository resources into the jasperserver database.
<code>js-upgrade-samedb-ce.sh</code>	(Linux) Upgrade jasperserver war file, add 4.1 repository resources into the jasperserver database.

8.5.1 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`

8.5.2 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 [5.11, “Troubleshooting Your JasperReports Server Configuration,” on page 33](#). 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 4.1.0

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

8.7 Logging into JasperReports Server 4.1.0

If your application server and JasperReports Server 4.1.0 were started cleanly, you can now prepare to login.

8.7.1 Clearing Your Browser Cache

Before you log into 4.1.0, 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. 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

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 55](#).

8.8.1 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-ce.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-4.1.0>/buildomatic</code>	
<code>js-ant upgrade-4.0-4.1-ce</code>	Execute SQL script for database upgrade to 4.1. Executes script buildomatic/install_resources/sql/<dbType>/upgrade-<dbType>-4.0.0-4.1.0-ce.sql (Note: For 4.1.0, 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 4.1.0
<code>js-ant import-sample-data-upgrade-ce</code>	(Optional) This step is optional. Loads the 4.1.0 sample data.
<code>js-ant deploy-webapp-ce</code>	Delete existing 4.0.0 war file, deploy 4.1.0 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 9 “Upgrade from 3.7.0 to 4.1.0” in order to upgrade to 4.1.0

If you are running a JasperReports Server version older than 3.7.0, then you will first have to upgrade to 3.7.0 before you can upgrade to 4.1.0. In order to upgrade to 3.7.0, you should download the WAR File Distribution zip package for 3.7.0 (jasperreports-server-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 4.1.0. The steps to carry out a 3.5.0 to 3.7.0 upgrade are documented in the JasperServer Installation Guide for the 3.5.0 release. You will need to download the JasperServer 3.5.0 release package to get the relevant files and documentation. To download the JasperServer 3.5.0 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 MySQL database, go to the <js-install> directory and run the following command:

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



If you installed JasperReports Server from the installer, you may specify `--user=jasperdb` in this command. If you receive an error about packet size, see section [A.2.5, “Maximum Packet Size in MySQL,”](#) on page 79.

For the PostgreSQL database, 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 format files. The output of the export operation is known as an export catalog.

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

```
Windows: js-export-ce.bat --everything-ce --output-dir js-backup-catalog
Linux:   js-export-ce.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 below is a summary of the options described in the section above.

The following table describes the available password encryption configuration options:

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

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

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 67](#). 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. Perform the same modifications to this file as in the procedure in section 10.4, “[Specifying Encryption Settings in the JasperReports Server WAR,” on page 68](#).

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 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;
```

10.6.2 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
```

10.7 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 sections above.

To import your backup catalog to the repository:

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

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

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

For information 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.

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

This chapter contains the following sections:

- [Introduction](#)
- [Import-Export Configuration Files](#)
- [Changing Your Configuration Settings](#)
- [Deploying a 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 shell based `js-import-ce.sh/bat` and `js-export-ce.sh/bat` scripts.

The different command styles would look like the following:

Example Command	Description
<code>cd <js-install>/buildomatic</code>	Change to buildomatic folder
<code>[buildomatic] js-ant export-everything-ce -DexportFile=js-catalog-exp.zip</code>	Export using buildomatic Ant script
<code>[shell script] js-export.sh --everything-ce --output-file=js-catalog-exp.zip</code>	Export using 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 4.0, this merging work is not complete. This means that if you are running the `js-import-ce.sh/bat` or `js-export-ce.sh/bat` shell scripts, you might have to do manual configuration depending on the database used.

In particular, the JDBC drivers will not be in place for non-MySQL databases.

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

11.3 Changing Your Configuration Settings

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

When you install JasperReports Server from the installer binary, the import and export shell scripts are automatically configured. However, if you are doing a manual installation from the WAR file distribution you must modify the following configuration file to include your database settings.

11.3.1 First Create a `default_master.properties` File

If you don't have a `<js-install>/buildomatic/default_master.properties` file in place then you should create one. For instance, copy and rename `buildomatic/sample_conf/mysql_master.properties` to `buildomatic/default_master.properties`. Then edit `default_master.properties` for you local settings. For more information see section [5.6.1, “Creating your Default Master Properties File,” on page 30](#).

Do the following:

- Edit: `<js-install>/buildomatic/default_master.properties`.
- Run: `js-ant gen-config`.

11.3.2 Here are the Configuration Locations

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

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

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

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

- `<js-install>/buildomatic/conf_source/ieCe/lib` (copy your DB type JDBC driver to this folder)

11.3.3 Check your js.jdbc.properties

The following tables give sample settings for each database.

This file will be automatically configured by buildomatic. You can double-check the file by looking here:

`<js-install>/buildomatic/build-conf/default/js.jdbc.properties`

You may specify an encrypted password instead of the clear-text password by default.

If your repository contains international characters, you may need to perform additional configuration for the import and export utilities. See section [A.8, “Exporting a Repository That Contains UTF-8,” on page 80](#).

Table 11-1 JDBC Settings in the js.jdbc.properties File

Database	Sample Property Values
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>
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>

11.3.4 If PostgreSQL Check your js.quartz.properties

This file will be automatically configured by buildomatic. You can double-check the file by looking here:

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

If your repository contains international characters, you may need to perform additional configuration for the import and export utilities. See section [A.8, “Exporting a Repository That Contains UTF-8,” on page 80](#).

11.5.1 Import-Export Updates for 4.0

As of JasperReports Server 4.0, there is a new option for import and export. This option is “`--include-access-events`”.

Specifying this option will allow the import or export of access event records stored in the JasperServer repository database.

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 part of the upgrade to 4.1.0 process.

In 3.7, the import-export shell scripts and configurations are located in the `<js-install-3.7>/scripts` folder.

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

```
<js-install-3.7>/scripts/config
<js-install-3.7>/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 as is described in the sections above:

```
<js-install-3.7>/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>/scripts/lib
```

APPENDIX A TROUBLESHOOTING

This appendix contains the following sections:

- **Installer Freezes**
- **Database Connectivity Errors**
- **Error Running a Report**
- **Database Error after Changing MySQL 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**
- **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**

A.1 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.2 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.2.1 Testing the Database Connection

The simplest database configuration problem is an incorrect user name or password. If you encounter database problems upon startup or login, check the user name and password by logging directly into your RDBMS as described in the following sections.

You can connect to your database using the database configuration settings that are found in JasperReports Server. This validates the database hostname, port, username, and password that are being used.

If you are having trouble logging into JasperReports Server on the login page, you can check the users and passwords that exist by viewing the contents of the `jasperserver.JIUser` table.

A.2.1.1 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.2.2 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:	<code><tomcat>/webapps/jasperserver/META-INF/context.xml</code>	
	<code><tomcat>/webapps/jasperserver/WEB-INF/hibernate.properties</code>	
	<code><tomcat>/apache-tomcat/webapps/jasperserver/WEB-INF/web.xml</code>	(JNDI config)
	<code><tomcat>/apache-tomcat/config/Catalina/localhost/jasperserver.xml</code>	(delete: see below)
JBoss:	<code><jboss>/server/default/deploy/js-mysql-ds.xml</code>	
	<code><jboss>/server/default/deploy/jasperserver.war/WEB-INF/hibernate.properties</code>	
	<code><jboss>/server/default/deploy/jasperserver.war/WEB-INF/web.xml</code>	
	<code><jboss>/server/default/deploy/jasperserver.war/WEB-INF/jboss-web.xml</code>	
GlassFish:	<code><glassfish>/domains/domain1/autodeploy/jasperserver.war/WEB-INF/hibernate.properties</code>	
	<code><glassfish>/domains/domain1/autodeploy/jasperserver.war/WEB-INF/js.quartz.properties</code>	
	<code><glassfish>/domains/domain1/config/domain.xml</code>	

A.2.3 Context.xml under Tomcat: Special Case

If you do multiple deploys of JasperServer to Tomcat, the `context.xml` (database connection configuration) can be superseded by a file in this location: `<tomcat>/conf/Catalina/localhost/jasperserver.xml` file.

When JasperServer is deployed, the `context.xml` will be copied to `<tomcat>/conf/Catalina/localhost/jasperserver.xml` (Tomcat does this by default).

Now, if you make changes to your `<tomcat>/webapps/jasperserver/META-INF/context.xml`, Tomcat will not “see” them. Instead, the `jasperserver.xml` will be used. This is confusing, but is the way that Tomcat operates.

If you edit your `context.xml` to fix a database problem:

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

Make sure and delete the jasperserver.xml file:

```
<tomcat>/conf/Catalina/localhost/jasperserver.xml      (delete this file)
```

A.2.4 Connect to Installed/Bundled Version of MySQL

These steps are for connecting under Linux.

If you have installed JasperReports Server using the bundled version of MySQL, you may want to connect to MySQL with the `mysql` command line application to examine the database. In order to connect, you will need to specify the socket that MySQL is using, as specified in the file `<install-dir>/jasperctl.sh`.

1. Get the socket file location by using the Linux `ps` (process status) command:

```
ps -ef | grep mysql
```

2. This displays lots of information. Look for the `--socket` value, for example:

```
... /home/devuser/jasperreports-server-cp-4.1.0/mysql/bin/mysqld_safe --port=3306 \
--socket=/home/devuser/jasperreports-server-cp-4.1.0/mysql/tmp/mysql.sock ...
```

3. Then run a command similar to the following:

```
/home/devuser/jasperreports-server-cp-4.1.0/mysql/bin/mysql -u jasperdb -p \
--socket=/home/devuser/jasperreports-server-cp-4.1.0/mysql/tmp/mysql.sock
```

A.2.5 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.3 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.4 Database Error after Changing MySQL Port Number

The default port for MySQL is 3306. If you entered a different port when you installed MySQL, the JasperReports Server installer configures them to communicate properly. If the MySQL 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.2.2, “Configuration File Locations,” on page 78](#).

A.5 Case Sensitivity for Table and Column Names

Some databases are case-sensitive with respect to table names and will consider “customer” and “Customer” to be two different tables. If JasperReports Server is using a case-sensitive database, it’s important that the table names specified in query strings in the JRXML file of a saved report match the actual table names found in the database. A mismatch may occur if you are transferring data from one database to another, which may cause the capitalization of table names to change.

Under Windows MySQL, table and column names are *not* case-sensitive.

Under Linux MySQL, table and column names are case-sensitive. Linux MySQL can be configured to be non-case-sensitive by setting the configuration parameter `lower_case_table_names` to 1 in the `my.ini` or `my.cnf` file. For more information search the MySQL documentation for a section about identifier case sensitivity.

Table and column names in PostgreSQL are case-sensitive.

A.6 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 [5.8, “Setting Java JVM Options,” on page 31](#). It is recommended that you add `-Xms128m -Xmx512m` to your `JAVA_OPTS` setting, but you may increase that to `-Xms512m -Xmx1024m` or larger if your server can support higher settings.

This Java option is set within the application server, so you must set it then restart your application server.

A.7 Error Running Scheduled Report

If you setup a scheduled report, chose to run it, and chose to save it as HTML or RTF, the report size can potentially get quite large. If you are running MySQL and you get the following error:

```
JDBC exception on Hibernate data access
org.hibernate.exception.GenericJDBCException: could not insert
```

The problem may be the default size of the MySQL “blob” datatype. You can increase the size of this datatype by updating your `my.ini` or `my.cnf` MySQL configuration file with the following setting:

```
max_allowed_packet=32M
```

A.8 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.8.1 Error During JasperServer 1.2 Export

Upgrading typically requires doing an export operation on your database. If you 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 `scripts/ji-export-util/jdbc.properties`. Check the URL in this file; it should look like the following:

```
jdbc:mysql://localhost:3306/jasperserver?useUnicode=true&characterEncoding=UTF-8
```

Note the ampersand & character. It is incorrect if it appears as `&`. The `&` is only correct in an HTML or XML context. It is incorrect in a properties file.

A.9 Importing Scheduled Jobs with Update Option

There is a JasperReports Server bug where if you import a set of resources that contain Report Jobs these jobs will not be loaded into the JasperReports Server repository if you are using the “--update” option of the import tool.



To workaround this problem, you should make sure to not specify the “--update” option on your `js-import-ce.bat/sh` command line.

A.10 JBoss Modifications

A.10.1 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 6.1, “Setting JVM Options for Application Servers,” on page 39](#).

A.10.2 JBoss Large INFO Log Message on Analysis Drill-through

JBoss has an internal mechanism to track and log information on unclosed JDBC connections. JasperServer Analysis leaves a connection open for performance reasons when doing an analysis 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.10.3 JBoss 4.0 Log4j Error on Startup

An error occasionally seen in JBoss 4.0 (tested on 4.0.5) has the following exception message:

```
log4j:ERROR "org.jboss.logging.util.OnlyOnceErrorHandler
```

JBoss is normally distributed with the log4j facility enabled. Log4j is initialized at JBoss startup. JasperReports Server also includes and uses log4j. When JBoss loads the JasperReports Server WAR file, the `OnlyOnceErrorHandler` exception can occur. This error is not fatal, but it can cause confusion when seen in the JBoss server log.

To remove this error, you can delete the JasperReports Server version of the log4j.jar file:

```
<jboss>/server/default/deploy/jasperserver.war/WEB-INF/lib/log4j-1.2.12
```

A.10.4 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.11 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:

```
org.quartz.jobStore.driverDelegateClass = org.quartz.impl.jdbcjobstore.PostgreSQLDelegate
```

A.12 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.12.1 Missing Java JDK

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

```
[exec] [ERROR] BUILD FAILURE
[exec] [INFO] -----
[exec] [INFO] Compilation failure
[exec] Unable to locate the Javac Compiler in:
[exec]   c:\Program Files\Java\jdk1.6.0_10\jre\..\lib\tools.jar
[exec] Please ensure you are using JDK 1.5 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.
```

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

If you are upgrading JasperReports Server, you can also use the Java 1.5 JDK bundled in the previously installed version, as described in [7.9.1, “Handling JasperReports Server Customizations,” on page 55](#).

A.12.2 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.12.3 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.13 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.14 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 and 6.0, 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.15 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.16 Problem Starting JasperReports Server on the Mac

Jaspersoft has seen some issues where the Tomcat included with the JasperReports Server is not shutdown properly. This could be related to the machine being shutdown while Tomcat is running.

When the Tomcat scripts start Tomcat, they write a "pid" (Process ID) file to the Tomcat folder. Tomcat uses this to determine whether the Tomcat instance is already running. When Tomcat is shutdown, this pid file is removed.

However, if the pid file is not removed on shutdown, Tomcat will fail to start up.

You may see this when you double-click the `jasperServerStart.app` startup. It will seem like JasperReports Server is starting up but it never actually starts up.

In order to recover from this issue, you will need to manually delete the pid file.

Delete catalina.pid using Finder:

1. Navigate to the `<js-install>/tomcat/temp` folder.

For instance: /Applications/jasperreports-server-<ver>/tomcat/temp

2. Delete catalina.pid.

Delete the catalina.pid file using Terminal shell:

1. Open a Terminal shell (Finder > Go > Utilities > Terminal Icon)
2. Navigate to the <js-install>/tomcat/temp folder.
For instance: /Applications/jasperreports-server-<ver>/tomcat/temp.
3. Enter the following command:

```
rm catalina.pid
```

To start and stop the MySQL and Tomcat components separately from the command line shell:

1. Open a Terminal shell (Finder > Go > Utilities > Terminal Icon).
2. Navigate to the <js-install> folder.
For instance: /Applications/jasperreports-server-<ver>.
3. To Start:

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

```
./ctlscript stop
```

or

```
./ctlscript tomcat stop  
./ctlscript mysql stop
```

