JasperReports® Server Community Project Installation Guide

Release 6.4

http://www.jaspersoft.com

This is version 0217-JSO64-39 of the *JasperReports Server Community Project Installation Guide*.

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION

TIBCO JasperReports® Server builds on TIBCO JasperReports® Library as a comprehensive family of Business Intelligence (BI) products, providing robust static and interactive reporting, report server, and data analysis capabilities. These capabilities are available as either stand-alone products, or as part of an integrated end-to-end BI suite utilizing 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.

This section describes functionality that can be restricted by the software license for JasperReports Server. If you don't see some of the options described in this section, your license may prohibit you from using them. To find out what you're licensed to use, or to upgrade your license, contact Jaspersoft.

The heart of the TIBCO Jaspersoft® BI Suite is the server, which provides the ability to:

- Easily create new reports based on views designed in an intuitive, web-based, drag and drop Ad Hoc Editor.
- Efficiently and securely manage many reports.
- Interact with reports, including sorting, changing formatting, entering parameters, and drilling on data.
- Schedule reports for distribution through email and storage in the repository.
- Arrange reports and web content to create appealing, data-rich Jaspersoft Dashboards that quickly convey business trends.

For users interested in multi-dimensional modeling, we offer Jaspersoft® OLAP, which runs as part of the server.

While the Ad Hoc Editor lets users create simple reports, more complex reports can be created outside of the server. You can either use Jaspersoft® Studio or manually write JRXML code to create a report that can be run in the server. We recommend that you use Jaspersoft Studio unless you have a thorough understanding of the JasperReports file structure.

You can use the following sources of information to learn about JasperReports Server:

- Our core documentation describes how to install, administer, and use JasperReports Server and Jaspersoft Studio. Core documentation is available as PDFs in the doc subdirectory of your JasperReports Server installation. You can also access PDF and HTML versions of these guides online from the Documentation section of the Jaspersoft Community website.
- Our Ultimate Guides document advanced features and configuration. They also include best practice recommendations and numerous examples. You can access PDF and HTML versions of these guides online from the Documentation section of the Jaspersoft Community website.

- Our Online Learning Portal lets you learn at your own pace, and covers topics for developers, system administrators, business users, and data integration users. The Portal is available online from the Professional Services section of our website.
- Our free samples, which are installed with JasperReports Library, Jaspersoft Studio, and JasperReports Server, are available and documented online. Please visit our GitHub repository.
- If you have a subscription to our professional support offerings, please contact our Technical Support team when you have questions or run into difficulties. They're available on the web at and through email at http://support.tibco.com and js-support@tibco.com.

JasperReports Server is a component of both a community project and commercial offerings. Each integrates the standard features such as security, scheduling, a web services interface, and much more for running and sharing reports. Commercial editions provide additional features, including Ad Hoc views and reports, advanced charts, dashboards, Domains, auditing, and a multi-organization architecture for hosting large BI deployments.

This chapter contains the following sections:

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 by the binary installer.
<js-install>	For manual installations, the directory where you unpack the WAR file distribution TIB_js-jrs_cp_6.4.0_bin.zip. See **2.5, "Choosing an Installation Directory," on page 17** for the default values.
<glassfish>	The directory where GlassFish is installed.
<java>	The directory where Java is installed.
<jboss>	The directory where JBoss is installed.
<postgresql>	The directory where PostgreSQL is installed. If you use our bundled instance of PostgreSQL, it's in the <js-install> directory.
<tomcat>	The directory where Apache Tomcat is installed. If you use our bundled instance of Tomcat, it's in <js-install> directory.

1.2 Supported Platforms

For a list of supported JDK/JVMs, application servers, databases, operating systems, and browsers, see the *TIBCO Jaspersoft Platform Support* document on the Documentation section of the Jaspersoft Community website.

1.3 Using IBM JDK 1.7

If you are using the IBM JDK 1.7, you need to set OWASP to use the correct Pseudo-random Number Generator (PRNG). To do this before installation, you can modify the WAR file as follows:

1. The WAR file is an archive format in a single file.
   a. Extract the `Websphere.jrs.csrfguard.properties` file using the following command:

```
cd <js-install>
"%JAVA_HOME%/bin/jar" xf jasperserver-pro.war WEB-INF/csrf/Websphere.jrs.csrfguard.properties
```

   This creates the `WEB-INF/csrf` folder in the current location and places the extracted file there.
   b. Rename the file from `Websphere.jrs.csrfguard.properties` to `jrs.csrfguard.properties` using the following command:

```
mv ./WEB-INF/csrf/Websphere.jrs.csrfguard.properties ./WEB-INF/csrf/jrs.csrfguard.properties
```

2. After you have modified the file, replace it in the WAR file archive using the following commands.

```
cd <js-install>
"%JAVA_HOME%\bin\jar" uf jasperserver.war WEB-INF/csrf/jrs.csrfguard.properties
```

1.4 JasperReports Server Distributions

JasperReports Server has two main distribution packages.

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

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

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

1.4.1 Installer Support

There are native installers for Linux, Macintosh, and Windows. See the *Jaspersoft Platform Support* document for the list of supported operating systems.

1.4.1.1 Installer Naming for 32-bit and 64-bit

The 32-bit and 64-bit installers are distinguished by file name.

Installer Type	Naming
32-bit installer (Linux only)	`TIB_js-jrs_cp_6.4.0_installer-linux-x86.run`

64-bit installer	`TIB_js-jrs_cp_6.4.0_installer-win-x64.exe`
	`TIB_js-jrs_cp_6.4.0_installer-linux-x64.run`
	`TIB_js-jrs_cp_6.4.0_installer-osx-x64.app.zip`

Note: x86 is shorthand referring to the 386, 486, and 586 CPU architecture.	

The 64-bit installer will put 64-bit versions of Java 8 and PostgreSQL 9 onto your system for increased speed and performance.

Note: The 64-bit installer will *not* work on a 32-bit system. The 32-bit installer will work on a 64-bit system, but we don't recommend it.

1.4.1.2 Installer Distribution Components

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

Component	Description
JasperReports Server Application	WAR file and configuration support scripts.
JasperReports Server Documentation	Found in the <js-install>/docs directory.
Apache Tomcat 8	Web application container. You can use the bundled version or an existing version.
Java 1.8 Runtime	Runs the web application container.
PostgreSQL 9 Database	Database server. You can use the bundled version or an existing version.
PhantomJS	Scriptable headless WebKit, required for exporting dashboards. We also recommend configuring JasperReports Server to use PhantomJS for graphical reports that are run in the background or scheduled. See the *JasperReports Server Community Project Administrator Guide* for more information.

1.4.1.3 Installing with Existing Components

The installer can use either bundled or existing instances of both the Apache Tomcat application server and the PostgreSQL database:

- If you want to use an existing Tomcat, it must be on the local machine.
- If you want to use an existing PostgreSQL, it can be on a local or remote machine. If it's on a remote Linux machine, configure PostgreSQL to allow remote connections as described in **2.7.4, "Enabling Connections to a Remote Host," on page 20**.

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

1.4.1.4 Running Components as Windows Services

The Windows installer installs PostgreSQL and Tomcat as Windows Services. You can manage these Services in the Windows Control Panel:

Control Panel > System and Security > Administrative Tools > Services

You'll find the PostgreSQL and Tomcat services by the following names:

- `jasperreportsPostgreSQL`
- `jasperreportsTomcat`

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

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

1.4.1.5 Installer on Windows 7

Due to a known issue with PhantomJS on Windows 7, you should not install PhantomJS using the bundled installer on Windows 7. Instead, to use PhantomJS, install JasperReports Server without PhantomJS and then install PhantomJS separately and configure JasperReports Server to use PhantomJS. See the *JasperReports Server Community Project Administrator Guide* for more information on configuring all JasperReports Server with PhantomJS.

1.4.2 WAR File Binary Distribution Support

Use the WAR file binary distribution package to install the JasperReports Server application if you can't use the installer. The WAR file supports additional applications not supported by the installer. If you want to use a database other than PostgreSQL and/or an application server other than Apache Tomcat, install JasperReports Server using the WAR file.

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

The application server should reside on the local machine, but the target database can be on a remote server. Using a remote PostgreSQL database on some Linux platforms requires a change to its configuration file, as described in **2.7.4, "Enabling Connections to a Remote Host," on page 20**.

The WAR file distribution includes `js-install` shell scripts (for Linux and Windows) that automate much of the installation using a single properties file. These scripts are:

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

The main contents of the WAR file binary distribution are shown in the following table.

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

1.4.2.1 About Bundled Apache Ant

The War File Distribution ZIP includes Apache Ant version 1.9.4. The buildomatic Ant scripts come with Windows and Linux batch scripts pre-configured to use the bundled version of Apache Ant. You call the buildomatic Ant scripts from the command line in the following manner:

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

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

If you want to run your own version of Ant, version 1.8.1 or higher is required.

The bundled Apache Ant has an additional jar (ant-contrib.jar) that enables conditional logic in Ant. If you're running your own Ant, copy this 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 **A.4, "Bash Shell for Solaris, IBM AIX, HP UX and FreeBSD," on page 55**.

1.5 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 in that distribution.

1.6 System Requirements

The following table contains the minimum and recommended resources for a full installation that includes PostgreSQL and an application server. The values are based on our own testing. You may find that JasperReports Server can run on systems with fewer resources or slower systems than stated in the minimum resources column. At the same time, it's 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	~1.3 Gigabytes	10GB free	40GB +
RAM		4GB	8GB +
Processor		2 core minimum	2.5GHz + multi-core Pentium for Windows, Mac, and Linux

1.7 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. UTF-8 is configured by default in the bundled Tomcat and PostgreSQL software. If you use any other software, refer to the *JasperReports Server Community Project Administrator Guide* for instructions about configuring software to support UTF-8.

CHAPTER 2 INSTALLING JASPERREPORTS SERVER

This chapter describes how to install JasperReports Server using the installer executable. This chapter contains the following sections:

2.1 Pre-installation Steps

When you run the installation executable, you can choose to install the bundled Apache Tomcat application server and PostgreSQL database or use an existing Tomcat and PostgreSQL.

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

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

The bundled installer is not meant for use in Enterprise Production environments.

2.2 Starting the Installer

In Windows, you'll need Administrative privileges to run the installer executable file. Right-click the binary installer file and select “Run as administrator” from the context menu.

`TIB_js-jrs_cp_6.4.0_installer-win-x64.exe` (64 bit only)

The Windows installer will get an error installing the PostgreSQL database if the Windows user does not have sufficient Administrative privileges and if the installer is not started by right-clicking to use “Run as administrator”.

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:

`./TIB_js-jrs_cp_6.4.0_installer-linux-x86.run` (32 bit)

`./TIB_js-jrs_cp_6.4.0_installer-linux-x64.run` (64 bit)

In Mac OSX, the installer is a .zip file. After download, you should find the installer already unpacked in your <user>/Downloads folder. Double-click the following:

`TIB_js-jrs_cp_6.4.0_installer-osx-x64.app` (64 bit only)

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

The welcome screen introduces the installer. Click **Next**.

If you're installing a 32-bit installer onto a 64-bit operating system you may get a popup reminder that a 64-bit installer is available. You can continue the 32-bit installation if you choose to.

On Windows you'll get an error installing the PostgreSQL database if you don't have Administrative privileges and if you don't start the installer by right-clicking to use “Run as administrator”.

2.3 Accepting the License Agreement

You must accept the license agreement or exit the installer.

When prompted, read the agreement and click **I accept the agreement** then click **Next**.

If installing from the command line, you must page through several screens of text to read the full agreement.

2.4 Choosing Installation Type

You can choose a full install of all components and sample data or a custom install that lets you choose the components you want and to take or leave the sample data.

Install All Option:

This option copies a Bundled version of the Apache Tomcat package and a Bundled version of the PostgreSQL database to your file system; adds all sample data (Reports, Data Sources, OLAP Views, etc) to your

JasperReports Server; and creates additional sample databases. The installer looks for open Tomcat ports from 8080 up and for open PostgreSQL ports from 5432 up.

After you choose this option, you can choose the installation directory for JasperReports Server. All files and components can then be installed with no further information required.

Custom Install:

With the custom install, you have the following choices: install a Bundled Tomcat or use an Existing Tomcat, install a Bundled PostgreSQL or use an Existing PostgreSQL, choose ports for Tomcat and PostgreSQL, and choose whether or not to install sample data.

2.5 Choosing an Installation Directory

When you're prompted to choose the JasperReports Server installation directory, you can accept the default directory or click **Browse** and choose a different location.

On the command line, press Enter to accept the default or enter a directory at the prompt to choose a different location.

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

Windows:	C:\Jaspersoft\jasperreports-server-cp-
Linux:	<USER_HOME>/jasperreports-server-cp-
Linux (as root)	/opt/jasperreports-server-cp-
Mac OSX	/Applications/jasperreports-server-cp-

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

2.6 Selecting a Tomcat Configuration

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

- **I want to use the bundled Tomcat**
  If you choose this option, the installer puts an instance of Tomcat 8 onto your system. Later, after choosing a bundled or existing database, you're prompted for the server port and shutdown port Tomcat will use. You can accept the default values or enter alternate values.
- **I want to use an existing Tomcat**
  If you have an instance of Tomcat on your system, you can choose this option. Later, after choosing a bundled or existing database, you're prompted for the location of Tomcat.
  - Browse to the folder where you installed Tomcat.

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

2.7 Selecting a PostgreSQL Configuration

JasperReports Server requires a database. The installer is pre-configured to run with the PostgreSQL database. You have two options:

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

2.7.1 Choosing the Bundled PostgreSQL

If you choose to install the bundled PostgreSQL, the installer puts PostgreSQL 9 onto your system. The default PostgreSQL port is 5432. If port 5432 is in use, the installer will prompt you to pick an alternate port. The installer sets the PostgreSQL administrator password to **postgres** and creates a PostgreSQL database user with administrator privileges and the credentials **jasperdb/password**.

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

Parameter	Default Value and Description
Binary Directory	The directory where the postgres and pgAdmin3 binaries are located.
Port	The port number PostgreSQL uses (default is 5432). Choose an alternate port if 5432 is in use.
IP or Host Name	The IP address or name of the machine where PostgreSQL is installed. The default value is 127.0.0.1.
PostgreSQL Administrative Password	Password of the database administrative user: postgres. The installer cannot handle special characters at the end of a password string. Incompatible characters include: & ; $
Database User Name	Hard coded default: jasperdb - The installer creates this user which is used to connect to the JasperReports Server database
Database User Password	Hard coded default: password - The installer uses this password for the jasperdb user.
Additional notes for Linux	If your Linux installation does not have a locale setting that supports UTF-8 encoding, your Bundled PostgreSQL instance will be initialized using a temporary locale (--locale=C). This will allow the PostgreSQL initdb to succeed with the desired UTF-8 database encoding.

2.7.2 Choosing an Existing PostgreSQL on a Local Host

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

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

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

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

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

2.7.3 Using an Existing PostgreSQL on a Remote Host

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

```
psql --version
```

or

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

For instance: `C:/Jaspersoft/PostgreSQL/9.0/bin/psql --version`

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

- Using your local PostgreSQL client tools, enter this command:

```
psql -U postgres -h <remote-host> -d postgres
```

or

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

You might also need to enable connections as described below.

2.7.4 Enabling Connections to a Remote Host

On most platforms, the default PostgreSQL installation doesn't allow remote connections for security reasons. You need to enable remote connections as described in this documentation:

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

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

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

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

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

   ```
   host all 192.168.12.10/32 trust
   ```
3. Allow TCP/IP connections to the remote PostgreSQL server instance by making the following change to the postgresql.conf file on the remote machine:

   From: `listen_addresses = 'localhost'`

   To: `listen_addresses = '*'`
4. Restart PostgreSQL.
5. Using your local PostgreSQL client tools, verify that you can connect to the target remote PostgreSQL from the local installation machine, as described in **2.7.3, "Using an Existing PostgreSQL on a Remote Host," on page 19**.

2.8 Installing Sample Data

During installation, you'll be prompted to install sample databases and sample reports. We provide these resources to help you evaluate the many features of JasperReports Server. This sample data includes:

- SugarCRM 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.

Our documentation provides tutorials that use this sample data. We strongly recommend that you install it.

2.9 Completing the Installation

After you've installed the files, you'll see several post-installation options on the final screen:

- **View Release Notes** - If you choose to view the release notes, you'll have to exit the release notes text viewer before JasperReports Server will launch.
- **Launch JasperReports Server Now** (for bundled Tomcat and PostgreSQL only) If you're installing on Linux, don't close the terminal window running the start script.

If you choose not to **Launch** JasperReports Server **Now**, the bundled components won't be started. If you have only one bundled component, it won't be started unless you use the Start/Stop menus or scripts.

- **Opt-in for JasperServer Heartbeat** - Sends anonymous system and version information to Jaspersoft using HTTPS.

2.10 Post-installation Steps

2.10.1 Updates Made by the Installer During Installation

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

Updates made to the application server

If you installed to an existing Tomcat, the installer attempted the following updates to the Tomcat environment:

File or Directory	Updates
Windows: bin/setclasspath.bat Linux and Mac OSX: bin/setclasspath.sh	Modifies JAVA_OPTS to add -Djs.license.directory. (Commercial installer only)
All platforms: lib	Adds PostgresQL JDBC drivers to this directory.

Updates made to the PostgreSQL database

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

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

2.10.2 Installer Output Log File Location

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

2.10.3 Setting your Java JVM Options

You need to set your Java JVM options. There are number of files where you can do this; refer to **4.1, "Setting JVM Options for Application Servers," on page 37**.

2.11 Starting and Stopping the Server

- **Start/Stop Menu — Windows**
- **Start/Stop Scripts — Linux**
- **Start/Stop Apps — Mac OSX**

2.11.1 Start/Stop Menu — Windows

This section describes start and stop procedures that vary depending on whether you installed the bundled Tomcat and PostgreSQL or used your own Tomcat and PostgreSQL.

2.11.1.1 Start/Stop Menus — Bundled Tomcat and PostgreSQL

If you installed the bundled Tomcat and PostgreSQL, use the Windows Start menu to start and stop JasperReports Server.

- Click **Start > All Programs >JasperReports Server> Start or Stop Services** then select Start Service or Stop Service.

2.11.1.2 Additional Information about the Bundled Tomcat and PostgreSQL

JasperReports Server Windows Service Names:

PostgreSQL and Tomcat, installed as Windows Services, are listed in the Windows Services Panel as:

- `jasperreportsPostgreSQL`
- `jasperreportsTomcat`

Preventing JasperReports Server from starting up automatically:

By default, the bundled services start automatically on a reboot, which also starts JasperReports Server. To change the startup mode for the services from automatic to manual:

- In the Windows Services Panel, select `jasperreportsTomcat`
- Right-click the `jasperreportsTomcat` service, and select properties
- Change the Startup type drop-down setting from Automatic to Manual
- Do the same for the `jasperreportsPostgreSQL` service

To Start JasperReports Server from the Windows Services Panel:

- Open the Windows Services Panel
- Select `jasperreportsPostgreSQL`, click Start

- Select `jasperreportsTomcat`, click Start

To Start JasperReports Server from the CMD Shell:

- Open a Windows CMD Shell
- Navigate to the root of the <js-install> folder (C:\Jaspersoft\jasperreports-server-cp-)
- `servicerun START`
- `servicerun STOP` (to shutdown JasperReports Server)

Running Processes:

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

- postgres.exe
- tomcat7.exe

2.11.1.3 Start/Stop Scripts — No Bundled Applications

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

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

JasperReports Server requires database and application servers to be started in this order:

1. Database server.
2. Application server.

2.11.2 Start/Stop Scripts — Linux

This section describes start and stop procedures that vary depending on whether you installed the bundled Tomcat and PostgreSQL or used your own Tomcat and PostgreSQL.

2.11.2.1 Manual Start/Stop

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

Start JasperReports Server:

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

Stop JasperReports Server:

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

To start and stop individual components:

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

2.11.2.2 Auto Start/Stop with Bundled Tomcat and PostgreSQL

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

<js-install>/scripts/linux/jasperserver

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

Once installed, these services start automatically when you reboot, which also starts JasperReports Server.

2.11.3 Start/Stop Apps — Mac OSX

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

/Applications/jasperreports-server-cp-

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

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

jasperServerStart.app

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

jasperServerStop.app

The Mac lists the following information in the Activity Monitor:

- java

  or

  org.apache.catalina.startup.Bootstrap
- postgres

2.11.3.1 Start/Stop Apps — Mac Dock

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

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

2.11.3.2 Start/Stop JasperReports Server — Mac Terminal Shell

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

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

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

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

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

2.12 Logging into JasperReports Server

To log into JasperReports Server on any operating system:

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

   http://<hostname>:8080/jasperserver

   - <hostname> is the name or IP address of the computer hosting JasperReports Server.
   - 8080 is the default port number for the Apache Tomcat application server. If you used a different port when installing your application server, specify its port number instead of 8080.

   The login page appears.
4. Log in using the following credentials:

User ID	Password	Description
jasperadmin	jasperadmin	Administrator for the default organization

If you installed the sample data, these additional sample end-users are also created. These users are non-administrative users with fewer system privileges.

User ID	Password	Description
joeuser	joeuser	Sample end-user
demo	demo	Sample end-user for the SuperMart Dashboard demonstration

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

To log into JasperReports Server on Windows:

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

To log into JasperReports Server on Mac OSX:

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

/Applications/<js-install>/jasperServerLogin

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

To use the Dock to log into JasperReports Server:

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

2.13 Log Files

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

Tomcat:	<tomcat>/webapps/jasperserver/WEB-INF/logs/jasperserver.log
JBoss:	<jboss>/server/default/deploy/jasperserver.war/WEB-INF/logs/jasperserver.log
GlassFish:	<glassfish>/domains/domain1/autodeploy/jasperserver.war/WEB-INF/logs/jasperserver.log

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

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

1. Browse to http://<hostname>:8080/jasperserver/log_settings.html. The Log Settings page appears.
2. Change logging levels using the drop-down menus.

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

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

2.14 Uninstalling the Server

If you install JasperReports Server using the installer executable, you can uninstall it programmatically.

2.14.1 Windows

To uninstall JasperReports Server on Windows 7:

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

2.14.2 Linux

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

To uninstall JasperReports Server:

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

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

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

2.14.3 Mac OSX

To use Finder to uninstall JasperReports Server:

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

2.14.4 Uninstall Survey

After running the uninstaller, you're prompted to take an uninstall survey from Jaspersoft. Survey answers are anonymous and help us improve our products. 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. If none of the reasons apply, enter a short explanation. Thank you for your feedback.

CHAPTER 3 INSTALLING THE WAR FILE DISTRIBUTION

For production environments, use the stand-alone WAR file distribution to install the JasperReports Server application. Download the WAR file distribution from the Jaspersoft community site (http://community.jaspersoft.com). The WAR file distribution comes in a file named TIB_js-jrs_cp_6.4.0_bin.zip in compressed ZIP format.

This chapter contains the following sections:

- **Applications Supported by the WAR File Distribution**
- **Installing the WAR File Using js-install Scripts**
- **Starting the Server**
- **Logging into the Server**
- **Troubleshooting Your Server Configuration**
- **Installing the WAR File Manually**

3.1 Applications Supported by the WAR File Distribution

3.1.1 Database and Application Server Support

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

Database	Application Server	Instructions Located In
PostgreSQL MySQL	Apache Tomcat JBossEAP/Wildfly GlassFish	This chapter.

Jaspersoft recommends that you use Apache Tomcat with PostgreSQL as your repository, unless you have a strong reason to use another configuration. For version information about these databases and application servers refer to the release notes in the root of the unpacked distribution ZIP.

3.1.2 Operating System Support for Bash Shell

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

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

3.2 Installing the WAR File Using js-install Scripts

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

Prerequisites for installing the WAR file:

1. Install a supported version of the Java Development kit (JDK). See the *TIBCO Jaspersoft Platform Support* document on the Documentation section of the Jaspersoft Community website for a list.
2. Create and set the `JAVA_HOME` system environment variable to point to the Java JDK location.
3. Locate or install one of the following application servers:
   - Apache Tomcat 6, 7, or 8
   - JBoss EAP 6.x or 7.x or Wildfly 8.x, 9.x, 10.x
   - Glassfish 4.1 using the default domain (domain1)
     If you use a custom domain with GlassFish, see **A.9.6, "GlassFish Modifications," on page 63**.
4. Locate or install the PostgreSQL or MySQL database.

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

For an optional pre-install validation test, run `js-install-ce.bat/sh test`. See **3.5.3.1, "js-install Script Test Mode," on page 34** for more information.

To install the WAR file using js-install scripts:

The scripts are intended for the bash shell.

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

1. Extract all files from `TIB_js-jrs_cp_6.4.0_bin.zip`. Choose a destination, such as `C:\Jaspersoft` on Windows, `/home/<user>` on Linux, or `/Users/<user>` on Mac.

   The directory, `TIB_js-jrs_cp_6.4.0_bin`, appears in the file location you choose.
2. Copy the `<database>_master.properties` file for your database from `sample_conf` and paste it to `buildomatic`:
   - Copy from — `<js-install>/buildomatic/sample_conf/`
   - Paste to — `<js-install>/buildomatic`

   For example, if your database is PostgreSQL, copy `postgresql_master.properties` to `<js-install>/buildomatic`.
3. Rename the file you copied to `default_master.properties`.
4. Edit the `default_master.properties` file to add the settings for your database and application server. **Table 3-1** lists sample property values for each supported database.

Table 3-1 Sample Values for the default_master.properties File

Database	Sample Property Values

PostgreSQL	`appServerType=tomcat [jboss-eap-6, wildfly, glassfish, skipAppServerCheck]`
	`appServerDir=c:\\Program Files\\Apache Software Foundation\\Tomcat 8.0.36`
	`dbHost=localhost`
	`dbUsername=postgres`
	`dbPassword=postgres`

MySQL	`appServerType=tomcat [jboss-eap-6, wildfly, glassfish, skipAppServerCheck]`
	`appServerDir=c:\\Program Files\\Apache Software Foundation\\Tomcat 8.0.36`
	`dbUsername=root`
	`dbPassword=password`
	`dbHost=localhost`

Note the following:

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

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

On Linux, if Tomcat is installed using apt-get, yum, or rpm, see **A.9.5, "Tomcat Installed Using apt-get/yum," on page 62**.

5. Password encryption

   The `default_master.properties` file has a property setting to enable encryption of passwords that reside on the file system. This applies to all files found under the buildomatic folder, as well as the connection pooling file used by Apache Tomcat (context.xml). Currently, password encryption for connection pooling supports only the Tomcat application server.

   To enable encryption on the file system, uncomment the `encrypt` property so it looks like this:

   ```
   encrypt=true
   ```

For more information about the encryption functionality, refer to the *JasperReports Server Security Guide*.

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

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

   If you encounter errors during the `js-install` script execution, see **3.5.3, "Error Running js-install Scripts (js-install-ce.bat/sh)," on page 34**.
7. Set Java JVM Options (required), as described in **4.1, "Setting JVM Options for Application Servers," on page 37**.

To view the output log, look in: `<js-install>/buildomatic/logs/js-install-ce-<date>.log`

3.3 Starting the Server

To run JasperReports Server:

Start your application server using one of these commands:

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

To view the JasperReports Server application logs, see **2.13, "Log Files," on page 26**.

3.4 Logging into the Server

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

http://<hostname>:8080/jasperserver

Example:

http://localhost:8080/jasperserver

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

The login page appears after compiling the necessary JSP files (this will take a few moments).

Use the following credentials to log into JasperReports Server:

User ID	Password	Description
`jasperadmin`	`jasperadmin`	Administrator for the default organization

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

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

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

3.4.1 JasperReports Server Heartbeat

After your initial login, you're asked to opt in to the JasperReports Server Heartbeat. The heartbeat helps Jaspersoft understand customer installation environments to improve our products. If you choose to enable the heartbeat, an HTTPS call at server startup time sends information like this to Jaspersoft:

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

You can manually enable or disable the heartbeat by modifying the following property file `jasperserver/WEB-INF/js.config.properties`. To disable the heartbeat, set the `heartbeat.enabled` property to false:

```
heartbeat.enabled=false
```

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

3.5 Troubleshooting Your Server Configuration

This section helps you troubleshoot the most common installation problems.

3.5.1 Startup Problems

If you encounter a problem trying to run a new JasperReports Server, an incorrect database configuration is the likely culprit. Another common cause is a mistake in the application server configuration files. For information about resolving these types of errors, see **Appendix A, "Troubleshooting," on page 53**.

3.5.2 Error Running a Report

If you have trouble running reports in your new JasperReports Server instance, see "**Error Running a Report**" in **Appendix A, "Troubleshooting," on page 53**.

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

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

To troubleshoot problems running js-install scripts:

1. Open the output log file located in:
   `<js-install>/buildomatic/logs/js-install-<date>-<number>.log`
2. Try to find the first error encountered by the `js-install` steps.
   - Go to the end of the output log.
   - Scroll back through lines of error messages until you find the first error logged. Typically, this error causes more errors later in the log.
   - Finding the original error is the way to understand the problem. However, this can often be tricky because Java stack traces in conjunction with the Spring application component framework can make the error output quite long.
3. Incorrect settings in the `default_master.properties` file cause most problems, which you can correct by editing your `default_master.properties` settings. Common errors are:
   - Typos in the path for the application server
   - Misspelling the hostname or password for the database

To recreate your default_master.properties settings:

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

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

3.5.3.1 js-install Script Test Mode

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

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

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

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

1. Navigate to the buildomatic directory:
   `cd <js-install>/buildomatic`

2. Enter the following command to run the `js-install` script in `test` mode:
   `./js-install-ce.sh test`

3.5.4 Problem Connecting to a Cloud Database Instance

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

3.6 Installing the WAR File Manually

You may need to install the WAR file manually when you cannot use the `js-install` scripts.

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

To install the WAR file distribution using manual buildomatic steps:

1. Start your database server.
2. Stop your application server.
3. Create and edit a `default_master.properties` file to add the settings in for your database and application server as described in **3.2, "Installing the WAR File Using js-install Scripts," on page 30**.
4. Open a Command Prompt as Administrator on Windows or open a terminal window on Linux or Mac. Run the following commands:

Table 3-2 Buildomatic Targets to Execute to Install the WAR File

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

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

On non-Linux Unix platforms, the js-ant commands may not be compatible with all shells. If you have errors, use the `bash` shell explicitly. For more information, see **A.4, "Bash Shell for Solaris, IBM AIX, HP UX and FreeBSD," on page 55**.

If you encounter an error when running `create-sugarcrm-db`, `create-foodmart-db`, or `create-js-db`, you can create the JasperReports Server database manually using the database administration tool for your particular database type. To create the JasperReports Server database manually for PostgreSQL or MySQL, see **Appendix B, "Manually Creating the JasperReports Server Database," on page 69**.

If you have previously installed the databases, you can drop the old versions and then recreate the databases. To do this, run the following drop commands before running the commands in **Table 3-3**

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

Commands	Description
`js-ant drop-sugarcrm-db` `js-ant drop-foodmart-db`	(Optional) Deletes the sample databases.
`js-ant drop-js-db`	(WARNING) This will delete the JasperReports Server repository database. Only run this command if you intend to recreate the `jasperserver` database

5. Set Java JVM Options (required) as described in **4.1, "Setting JVM Options for Application Servers," on page 37**.

CHAPTER 4 JVM OPTIONS AND WORKING WITH JDBC DRIVERS

This chapter contains the following sections:

4.1 Setting JVM Options for Application Servers

Java Virtual Machine (JVM) runtime parameters normally need to be explicitly set so that the memory settings have values larger than the default settings. The options and values depend on your version of Java and the application server you use. For a list of supported JDK/JVMs and application servers, see the *TIBCO Jaspersoft Platform Support* document on the Documentation section of the Jaspersoft Community website.

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

4.1.1 Tomcat and JBoss JVM Options

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

The following example settings are for 64-bit systems. For 32-bit systems, see **"Setting your Java JVM Options" on page 22**.

JVM Options on Windows (64 bit)	
Options for all app servers	set JAVA_OPTS=%JAVA_OPTS% -Xms1024m -Xmx2048m -Xss2m set JAVA_OPTS=%JAVA_OPTS% -XX:+UseConcMarkSweepGC set JAVA_OPTS=%JAVA_OPTS% -XX:+CMSClassUnloadingEnabled

JVM Options on Windows (64 bit)	
Additional options for JDK 1.7	set JAVA_OPTS=%JAVA_OPTS% -XX:PermSize=32m -XX:MaxPermSize=512m
Additional options for JDK 1.8	set JAVA_OPTS=%JAVA_OPTS% -XX:MetaspaceSize=128m
Additional options for JBoss	set JAVA_OPTS=%JAVA_OPTS% -Djavax.xml.soap.MessageFactory=org.apache.axis.soap.MessageFactoryImpl set JAVA_OPTS=%JAVA_OPTS% -Djavax.xml.soap.SOAPConnectionFactory=org.apache.axis.soap.SOAPConnectionFactoryImpl set JAVA_OPTS=%JAVA_OPTS% -Djavax.xml.soap.SOAPFactory=org.apache.axis.soap.SOAPFactoryImpl
Additional option for JBoss EAP 6.1	set JAVA_OPTS=%JAVA_OPTS% -Djavax.xml.transform.TransformerFactory=org.apache.xalan.processor.TransformerFactoryImpl

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

JVM Options on Linux and Mac OSX (64 bit)	
Options for all app servers, all JDKs	export JAVA_OPTS="$JAVA_OPTS -Xms1024m -Xmx2048m -Xss2m" export JAVA_OPTS="$JAVA_OPTS -XX:+UseConcMarkSweepGC" export JAVA_OPTS="$JAVA_OPTS -XX:+CMSClassUnloadingEnabled"
Additional options for JDK 1.7	export JAVA_OPTS="$JAVA_OPTS -XX:PermSize=32m -XX:MaxPermSize=512m "
Additional options for JDK 1.8	export JAVA_OPTS="$JAVA_OPTS -XX:MetaspaceSize=128m "
Additional options for JBoss	export JAVA_OPTS="$JAVA_OPTS -Djavax.xml.soap.MessageFactory=org.apache.axis.soap.MessageFactoryImpl" export JAVA_OPTS="$JAVA_OPTS -Djavax.xml.soap.SOAPConnectionFactory=org.apache.axis.soap.SOAPConnectionFactoryImpl" export JAVA_OPTS="$JAVA_OPTS -Djavax.xml.soap.SOAPFactory=org.apache.axis.soap.SOAPFactoryImpl"
Additional option for JBoss EAP 6.1	export JAVA_OPTS="$JAVA_OPTS -Djavax.xml.transform.TransformerFactory=org.apache.xalan.processor.TransformerFactoryImpl"

You can set JVM options multiple ways. Sections **4.1.2** - **4.1.4** present step-by-step instructions for performing this task. Alternatively, you can add your `JAVA_OPTS` settings to any of the following files.

File	Add JVM Options After This Line on Windows
<tomcat>\bin\setclasspath.bat	`set JAVA_ENDORSED_DIRS=%BASEDIR%\common\endorsed`
<tomcat>\bin\setenv.bat	`JAVA_OPTS` setting can go anywhere in this file.
<jboss>\bin\standalone.conf.bat	Find the existing `JAVA_OPTS` line, remove the default memory settings from this line, and add a new line with the recommended `JAVA_OPTS` after this line. (For example, for JBoss EAP 6.1 on JDK 1.7, remove `Xms1303M -Xmx1303M -XX:MaxPermSize=256"` and add the recommended settings on a new line.)

File	Add JVM Options After This Line on Linux
<tomcat>/bin/setclasspath.sh	`JAVA_ENDORSED_DIRS="$BASEDIR"/common/endorsed`
<tomcat>/bin/setenv.sh	`JAVA_OPTS` setting can go anywhere in this file.
<jboss>/bin/standalone.conf	Find the existing `JAVA_OPTS` line, remove the default memory settings from this line, and add a new line with the recommended `JAVA_OPTS` after this line. (For example, for JBoss EAP 6.1 on JDK 1.7, remove `Xms1303M -Xmx1303M -XX:MaxPermSize=256` and add the recommended settings on a new line.)

4.1.2 Changing JVM Options for Tomcat as a Windows Service

If you installed JasperReports Server to use Tomcat running as a Windows service, you can set Java options on the Java Tab of the Tomcat Properties dialog:

1. Launch the Tomcat configuration application. If you installed the bundled Tomcat, you can do this by going to the <js-install>/apache-tomcat/bin directory and double-clicking the jasperreportsTomcat.exe file. (If you have multiple instances of JasperReports Server installed, the file name will be of the form jasperreportsTomcatnum<number>.exe, for example, jasperreportsTomcatnum2.exe.) If you installed Tomcat using an existing Windows service, look for an .exe file in the same location, with the same name as your Tomcat service, or select the service from the Windows Start menu:

   Start > Programs > Apache Tomcat > Configure Tomcat (Run as administrator)
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 tables 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, on JDK 1.7, add options as follows:

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

These example settings are for 64-bit systems. For 32-bit systems, see **"Setting your Java JVM Options" on page 22**.

5. Click `Apply`, then click `OK`.
6. Stop and restart Tomcat.

4.1.3 Changing JVM Options for Bundled Tomcat on Linux

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

1. Open the following file for editing:

   ```
   cd <js-install>/apache-tomcat/scripts/ctl.sh
   ```

2. Look for the `start_tomcat()` function and locate the `JAVA_OPTS` variable inside it.
3. Modify the `JAVA_OPTS` values according to the tables above. For example, on JDK 1.7:

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

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

4. Save and close the `ctl.sh` file.
5. Stop and restart PostgreSQL and Tomcat as described in **2.11, "Starting and Stopping the Server," on page 22**.

4.1.4 Changing GlassFish JVM Options

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

4.1.4.1 Setting GlassFish JVM Options with the asadmin Command

1. First make sure your GlassFish instance is up and running, then enter the command as a single line. For example, on JDK 1.7:

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

2. Restart the application server using the following commands:

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

When running the `asadmin create-jvm-options` command, you may see error messages like this:

```
[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.

4.1.4.2 Setting GlassFish JVM Options by Editing domain.xml

1. Open the `<glassfish>/domains/domain1/config/domain.xml` configuration file for editing.
2. Add the appropriate lines to the section `java-config`. For example, on JDK 1.7:

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

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

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

4.2 Working With JDBC Drivers

This section describes how to set up your installation to use a driver other than the default driver.

4.2.1 Open Source JDBC Drivers

For open source JDBC drivers, buildomatic is set up to use a single default driver. If you want to use a driver other than the default driver, you can modify the buildomatic property files that determine the default JDBC driver.

The buildomatic JDBC driver property files are set up to point to a specific driver jar. This allows for multiple driver jar files in the same `buildomatic/conf_source/db/<dbType>/jdbc` folder. During the installation procedure only the default driver jar is copied to your application server.

If you want to use a newer JDBC driver version or a different JDBC driver, you can modify the buildomatic properties seen in your `default_master.properties` file.

4.2.1.1 PostgreSQL Example

The buildomatic/conf_source/db/postgresql/jdbc folder contains these driver files:

```
postgresql-9.2-1002.jdbc3.jar
postgresql-9.2-1002.jdbc4.jar
```

If, for instance, you want to change the default driver used by PostgreSQL from type jdbc4 to jdbc3, edit your default_master.properties file:

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

Uncomment and change:

```
# maven.jdbc.version=9.2-1002.jdbc4
```

To:

```
maven.jdbc.version=9.2-1002.jdbc3
```

When you next run a buildomatic command, such as deploy-webapp-ce, the jdbc3 driver will be copied to your application server.

4.2.1.2 MySQL Example

The `buildomatic/conf_source/db/mysql/jdbc` folder contains this driver file:

```
mariadb-java-client-1.1.2.jar
```

If, for instance, you want to use a JDBC driver built and distributed by the MySQL project, such as `mysql-connector-java-5.1.30-bin.jar`, you first need to download the driver from the MySQL Connector/J download location:

```
https://dev.mysql.com/downloads/connector/j/
```

Next, change your buildomatic configuration properties to point to this new driver.

Edit your default_master.properties file:

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

Uncomment and change:

```
# jdbcDriverClass=com.mysql.jdbc.Driver
# maven.jdbc.groupId=mysql
# maven.jdbc.artifactId=mysql-connector-java
# maven.jdbc.version=5.1.30-bin
```

To:

```
jdbcDriverClass=com.mysql.jdbc.Driver
maven.jdbc.groupId=mysql
maven.jdbc.artifactId=mysql-connector-java
maven.jdbc.version=5.1.30-bin
```

4.2.1.3 Glassfish Load Balancing Example

For Glassfish, the URL must be edited from the admin console.

1. Open the Glassfish admin console. The default location is `http://hostname:4848`
2. Go to **Resources > JDBC > JDBC Connection Pools** and select the connection pool for the data sources you want.
3. Select the Additional Properties tab and set the **url** property. For example:

```
jdbc:tibcosoftware:oracle//server1:1521;ServiceName=SERVICE;AlternateServers=(server-
2:1521,server3:1521,server4:1521);LoadBalancing=true
```

4.2.2 Application Server Copy-to Locations

When the `deploy-webapp-ce` buildomatic target is executed it copies the JDBC driver to the following default locations:

Tomcat:	<tomcat>/lib
JBoss:	<jboss>/standalone/deployments
Wildfly:	<wildfly>/standalone/deployments
GlassFish:	<glassfish>/domains/domain1/lib/ext

4.3 Locating and Changing Buildomatic Configuration Files

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

4.3.1 Regenerating Buildomatic Settings

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

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

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

If you want to explicitly regenerate your configuration, run the following buildomatic targets:

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

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

4.3.2 Locating Buildomatic-Generated Property Files

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

installation.

The generated property files are in this location:

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

Some of the key configuration files are:

`js.jdbc.properties`

`js.quartz.properties`

`js-glassfish-ds.xml`

`js-jboss-ds.xml`

`maven_settings.xml` - (used for source code build)

More generated property files are:

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

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

`META-INF/context.xml`

`WEB-INF/hibernate.properties`

`WEB-INF/js.quartz.properties`

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

4.3.3 Buildomatic Location for JasperReports Server WAR File

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

`<js-install>/jasperserver.war`

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

- <js-install>/jasperserver.war
  Unpacked and copied to <tomcat>/webapps/jasperserver/*
- <js-install>/buildomatic/build_conf/default/webapp/META-INF/context.xml
  Copied to <tomcat>/webapps/jasperserver/META-INF/context.xml
- <js-install>/buildomatic/build_conf/default/webapp/WEB-INF/hibernate.properties
  Copied to <tomcat>/webapps/jasperserver/WEB-INF/hibernate.properties
- <js-install>/buildomatic/build_conf/default/webapp/WEB-INF/js.quartz.properties
  Copied to <tomcat>/webapps/jasperserver/WEB-INF/js.quartz.properties
- <js-install>/buildomatic/build_conf/db/postgres/jdbc/postgresql-9.2-1002.jdbc4.jar
  Copied to <tomcat>/lib

4.3.4 Buildomatic Location for SQL Scripts

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

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

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

<js-install>/buildomatic/install_resources/sql/postgresql/js-create.ddl
<js-install>/buildomatic/install_resources/sql/postgresql/quartz.ddl
<js-install>/buildomatic/install_resources/sql/postgresql/upgrade-postgresql-6.0.0-6.1.0-ce.sql
<js-install>/buildomatic/install_resources/sql/postgresql/js-drop.ddl
<js-install>/buildomatic/install_resources/sql/postgresql/drop-quartz.ddl

You can run these scripts manually by copying them to the location of your database client software.

4.3.5 Buildomatic Location for Database Creation Scripts

For most databases the buildomatic scripts can create the metadata repository database used by JasperReports Server. This is the database that stores data defining users, roles, data sources, reports, OLAP views, domains, and other data. 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 located in the following directory:

<js-install>/buildomatic/conf_source/db/<db_name>/scripts.properties

4.3.6 Buildomatic Location for Sample Data Catalog ZIP Files

Buildomatic includes export files that hold the JasperReports Server sample data (with 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:

js-catalog-<db_name>-minimal-ce.zip
js-catalog-<db_name>-ce.zip

4.3.7 Hibernate Properties Settings

After you run buildomatic to generate your configuration files, your `hibernate.properties` settings are in the following directory:

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

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

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

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

Hibernate property values are:

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

4.3.8 Database Connection Configuration Files

4.3.8.1 Tomcat

When you've set up the buildomatic configuration for your database, the Tomcat context.xml will be automatically created with the appropriate settings for JasperReports Server.

When you run the buildomatic target `deploy-webapp-ce`, 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

Older versions of Tomcat will 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 can be confusing for Tomcat users who try to change their database settings. If you change your settings, delete the file in this location:

<tomcat>/conf/Catalina/localhost/jasperserver.xml

4.3.8.2 JBoss

When you've set 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 you run the buildomatic target `deploy-webapp-ce`, 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>/standalone/deployments/jasperserver.war/WEB-INF/js-jboss7-ds.xml

When JasperReports Server is running under JBoss, a couple of INFO log messages and an XML/A connection error may occur depending on the version of JBoss you're running.

For more information, refer to troubleshooting section **A.9.7, "JBoss Modifications," on page 64**.

4.3.8.3 Glassfish

After you've set 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 you run the buildomatic target `deploy-webapp-ce`, 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 data source definition manually, run a command similar to the following:

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

4.4 Configuring Report Scheduling

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

In a deployed JasperReports Server instance, you'll find the js.quartz.properties file in this location:

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

For mail server configuration, you'll find an additional property setting for authentication in this file:

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

The following configurations are discussed in this section:

- Mail Server Configuration
- Quartz Driver Delegate Class
- Report Scheduler Web URI
- Quartz Table Prefix
- Settings for import-export
- Setting Properties in the default_master.properties File

4.4.1 Mail Server Configuration Settings

You can specify email addresses to notify when a report completes. To do this, configure JasperReports Server to contact an email server as shown in the following table.

Configuration File	
<app-server>/<deployment>/WEB-INF/js.quartz.properties	
Property	**Description**
`report.scheduler.mail.sender.host`	The name of the computer hosting the mail server
`report.scheduler.mail.sender.username`	The name of the mail server user JasperReports Server can use
`report.scheduler.mail.sender.password`	The password of the mail server user
`report.scheduler.mail.sender.from`	The address for in the **From** field on email notifications
`report.scheduler.mail.sender.protocol`	The protocol that the mail server uses. JasperReports Server supports only SMTP. Note: Your entry must be lower case (`smtp`)
`report.scheduler.mail.sender.port`	The port number the mail server uses. The default is typically 25 (other ports may not work in earlier JasperReports Server 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`.

4.4.2 Database Settings for the Quartz Driver Delegate Class

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

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

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

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

4.4.3 Settings for the Report Scheduler Web URI

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

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

To set this value manually, edit this file:

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

Change the properties as shown in the following table.

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`

4.4.4 Settings for the Quartz Table Prefix

For databases that support schemas, you can set the Quartz table prefix to include the schema, if you use one.

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

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

Property	Description
`quartz.tablePrefix`	The prefix for the quartz table, including any schema name.

4.4.5 Settings for Import-Export

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

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

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

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

Change the following properties:

Property	Description
`quartz.delegateClass`	Set to the same value as described in **4.4.2, "Database Settings for the Quartz Driver Delegate Class," on page 48**.
`quartz.tablePrefix`	Set to the same value as described in **4.4.4, "Settings for the Quartz Table Prefix," on page 49**

4.4.6 Setting Properties in the default_master.properties File

You can modify the `default_master.properties` file to configure JasperReports Server functionality. Uncomment the properties you want to have them take effect upon installation. The properties are documented directly in the `default_master.properties` file:

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

You'll find a sample master.properties here (in the case of PostgreSQL):

```
<js-install>/buildomatic/sample_conf/postgresql_master.properties
```

When you execute the `js-install-ce.sh/bat` script (or the underlying `deploy-webapp-ce` ant target), these properties will be set in the deployed JasperReports Server in the `js.quartz.properties` file.

4.4.6.1 Report Scheduler Email Properties

You can set the following properties to configure the Report Scheduler email (default values are shown):

```
quartz.mail.sender.host=mail.localhost.com
quartz.mail.sender.port=25
quartz.mail.sender.protocol=smtp
quartz.mail.sender.username=admin
quartz.mail.sender.password=password
quartz.mail.sender.from=admin@localhost.com
quartz.web.deployment.uri=http://localhost:8080/jasperserver
```

4.4.6.2 Diagnostic Properties

The following properties configure the Diagnostic functionality:

```
diagnostic.jmx.usePlatformServer = false
diagnostic.jmx.port = 10990
```

```
diagnostic.jmx.name = jasperserver
diagnostic.jmx.rmiHost = localhost
```

Look at the descriptions of the properties in the default_master.properties file and also refer to the *JasperReports Server Community Project Administrator Guide* for more information on these settings.

4.5 Updating XML/A Connection Definitions

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

If you have Jaspersoft OLAP enabled (via your license), JasperReports Server can make XML/A connections over the Web Services interface. These connections need a user account for authentication. You may have different usernames and passwords than the defaults in the sample data. Additionally, your application server hostnames and port may be different than the default values. In such cases, the connections and resources that rely on them will fail.

The sample connections are:

- Foodmart Sample XML/A connection
- SugarCRM Sample XML/A connection

To validate and update these resources:

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

APPENDIX A TROUBLESHOOTING

This appendix contains the following sections:

A.1 Binary Installer Freezes

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

A.1.1 Installer Log Files

If you get an error when running the JasperReports Server installer on any platform, look at the log file created by the installer. This log records the status and completion of installer operations. If a specific error occurred, you may find an explicit error message. Even without an explicit error message, the log file should help you locate the cause of the error.

You'll find the installer log for your platform in the following location:

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

If you've tried multiple installs, make sure you view the most recent install log.

A.2 Error Running Buildomatic Scripts

The buildomatic scripts depend on both Java and Apache Ant. Two common configuration errors are possible when attempting an installation using these scripts (if you're not using the included, bundled Apache Ant).

A.2.1 Missing Java JDK

If you have the Java JRE (Java Runtime Environment) instead of the JDK, you won't have all the required utilities. In particular, you may see an error referring to the tools.jar, 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\jdkx.x.x_xx\jre\..\lib\tools.jar
[exec] Please ensure you are using JDK x.x 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 Oracle web site.

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

If you're using your own version of Ant and your Ant instance doesn't have the ant-contrib.jar in the lib directory, you'll get an error similar to the following:

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

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

A.2.3 Failure with '$' Character in Passwords in Buildomatic Scripts

If your password in buildomatic scripts includes two of more '$' characters in a row, Ant will not accept it. This issue does not occur when dollar signs are separated by other characters. For example, `$pas$word$` or `pas$word$` will not fail.

If you have two consecutive dollar signs, you'll need to escape each with three more dollar signs. For example, if your password is `pa$$word`, enter it as `pa$$$$$$$$word` in the configuration file. Once you do this, JasperReports Server will set all data connections to `pa$$word`.

A.2.4 Older Apache Ant Version

As of release 6.1 of JasperReports Server, Apache Ant version 1.9.4 or later is recommended. The earliest compatible version is Ant 1.8.1.

Older versions of Ant will cause an error similar to the following:

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

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

```
ant -version
```

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

```
echo $CLASSPATH
```

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

A.3 Unable to Edit Files on Windows 7

In some cases, you may want to manually edit files in your C:/Jaspersoft directory during or after installation. For security reasons, Windows 7 doesn't allow normal processes to change files in many folders including the Program Files folder, for instance. When you attempt to edit these files, you may see an error like this:

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

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

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

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

The bash shell is required to execute the `js-install` shell scripts described in **Chapter 3, "Installing the WAR File Distribution," on page 29**. The following `js-install` and `js-upgrade` scripts are in the buildomatic folder:

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

The bash shell is not included by default in all Unix platforms. When the bash shell is not available, you'll need to download and install the bash shell specific to your platform.

Alternatively, you can manually run the same "buildomatic" Ant targets that are run by the `js-install` script. These Ant targets are listed in **"Troubleshooting Your Server Configuration" on page 33**.

Also, make sure you've updated your local Ant to include `ant-contrib.jar`, which supports conditional logic in Ant. Copy the `ant-contrib.jar` to your `<ant_home>/lib` folder from:

```
buildomatic/extra-jars/ant-contrib.jar.
```

For more information see **A.2.2, "Forgot to Copy the File ant-contrib.jar," on page 54**.

If you try using the Ant that's included with the JasperReports Server WAR file Distribution ZIP package, you may get the same non-bash syntax error. You may get the error below, for example:

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

If you have the bash shell installed, you can try executing the js-ant command by calling bash explicitly, for example:

```
bash js-ant help-install
```

A.5 Linux Installer Issue with Unknown Host Error

If your Linux server doesn't have proper hostname entries in the `/etc/hosts` file, you may get installer errors.

The installer carries out an import operation to load the core minimal data into the repository database. This import operation can fail if the host is not configured.

If the import operation fails during installation, the installation will also fail. However, there should be an installation.log in the root of the installation folder to help debug the problem. The `installation.log` is located here:

```
<js-install>/installation.log
```

An improperly configured hosts file typically causes error messages like these:

```
Caused by: java.net.NoRouteToHostException: No route to host
com.mysql.jdbc.exceptions.jdbc4.CommunicationsException: Communications link failure
ERROR Cache:145 - Unable to set localhost. This prevents creation of a GUID
java.net.UnknownHostException
org.quartz.SchedulerException: Couldn't get host name!
```

To fix the /etc/hosts file:

1. Include entries that look like these:

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

   For instance:

   ```
   127.0.0.1       localhost.localdomain          localhost
   172.17.5.0      myhost.jaspersoft.com          myhost
   ```

2. You can also double check the file `/etc/sysconfig/network` (if it exists). In this file it would be similar to the following:

   ```
   HOSTNAME=myhost
   ```

3. After fixing the `/etc/hosts` file, reinstall JasperReports Server.

A.6 Installation Error with Windows Path

If the path of the war archive exceeds the maximum length allowed by Windows, you'll get an error message like the one shown below.

```
BUILD FAILED
c:\jaspers\war_file_installations\war_mysql_500\jasperreports-server-5.0-bin\buildomatic\bin\db-com-
mon.xml:871:
The following error occurred while executing this line:
c:\jaspers\war_file_installations\war_mysql_500\jasperreports-server-5.0-bin\buildomatic\bin\import-
export.xml:264:
The following error occurred while executing this line:
c:\jaspers\war_file_installations\war_mysql_500\jasperreports-server-5.0-bin\buildomatic\bin\import-
export.xml:158:
java.io.IOException: Cannot run program "C:\Program
Files\Java\jdkx.x.x_xx\jre\bin\java.exe": CreateProcess error=206, The filename
or extension is too long
        at java.lang.ProcessBuilder.start(ProcessBuilder.java:460)
        at java.lang.Runtime.exec(Runtime.java:593)
```

You'll need to move the war archive to reduce the path length. More information is available from Microsoft at: http://msdn.microsoft.com/en-us/library/windows/desktop/aa365247(v=vs.85).aspx.

A.7 Mac OSX Issues

A.7.1 Problem Starting JasperReports Server on Mac

We have seen some issues caused by the improper shutdown of the Tomcat included with JasperReports Server. This may be caused by shutting the machine down 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. JasperReports Server seems to be starting up, but it never actually does.

To recover from this issue, manually delete the pid file.

Delete catalina.pid using Finder:

1. Navigate to the `<js-install>/tomcat/temp` folder
   For instance: `/Applications/jasperreports-server-cp-/tomcat/temp`
2. Delete `catalina.pid`

Delete the catalina.pid file using Terminal shell:

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

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

1. Open a Terminal shell (Finder > Go > Utilities > Terminal Icon).
2. Navigate to the `<js-install>` folder.
   For instance: `/Applications/jasperreports-server-cp-`
3. To Start:
   `./ctlscript start postgresql`
   `./ctlscript start tomcat`
4. To shutdown:
   `./ctlscript stop`
   or
   `./ctlscript stop tomcat`
   `./ctlscript stop` postgresql

A.8 Database-related Problems

A.8.1 Database Privileges Required By JasperReports Server

Install/upgrade process permissions:

The JasperReports Server installation/upgrade processes and the repository database user need the following privileges to install and initialize the `jasperserver` repository database.

Database Component	Permissions Required
databases	`CREATE` `DROP`

tables
indexes
constraints
	`CREATE`
	`ALTER`
	`DROP`

data records	`INSERT`
	`UPDATE`
	`DELETE`

If you are upgrading in a restricted environment, your database administrator may need to give you temporary admin permissions for the upgrade. For example, if you are using PostgreSQL for your database, the database admin may use one of the following workarounds:

- Add administrator credentials in the default_master.properties file prior to upgrade and then replace them with jasperadmin credentials after upgrade.
- Prior to upgrade, grant `CREATE` and `DROP` permissions at the database server level for the jasperadmin user , then revoke those permissions after successful upgrade.

A.8.2 Database Connectivity Errors

The most common problems encountered with a new JasperReports Server instance are database configuration problems. The connection may fail because the application server can't find the driver for the data source. For example, in a default installation of JasperReports Server, Tomcat looks for data source drivers in `<js-install>/apache-tomcat/lib`. If the driver's in a different location, put a copy of the driver in this directory and restart Tomcat.

A.8.2.1 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 below.

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

If you're having trouble logging into JasperReports Server on the login page, check the existing users and passwords in the `jasperserver.JIUser` table.

A.8.2.2 Logging into PostgreSQL

Run the PostgreSQL client from the command line and try to connect to the database. For example:

```
psql -U postgres jasperserver
```

A.8.2.3 Logging into MySQL

Run the MySQL client from the command line and try to log in directly as the `root` user, for example:

```
<mysql>/bin/mysql -u root -p
```

You're prompted for the password of the user you specified on the command line.

A.8.3 Maximum Packet Size in MySQL

If you're upgrading or importing into a MySQL database and your repository contains large objects like images, you may see an error like this:

```
ERROR 1153 (08S01): Got a packet bigger than 'max_allowed_packet' bytes
```

The default `max_allowed_packet` on the MySQL server is 1M (one Megabyte = 1,048,576 bytes). The most effective fix is to change this value in the server configuration to accommodate the largest resource stored in your repository. The server configuration file is typically named my.cnf (or my.ini) and located in the MySQL root directory, but this may vary. Change the configuration setting to a larger value, for example:

```
max_allowed_packet = 16M
```

For more information, see http://dev.mysql.com/doc/refman/5.0/en/packet-too-large.html.

After changing this value, restart the MySQL server. Then perform the upgrade or import step again.

A.8.3.1 Connection reset by peer MySQL Error

If you are using the MariaDB JDBC driver to connect to the MySQL database and get an error like:

```
Could not send query:
Connection reset by peer: socket write error
```

This message refers to the maximum packet size error described above. Follow those instructions.

A.8.4 Case Sensitivity for Table and Column Names

In some databases, table names are case-sensitive and “customer” and “Customer” are two different tables.

If you're using a case-sensitive database for JasperReports Server, table names specified in query strings in the JRXML file of a saved report must match the table names used in the database. A mismatch when transferring data from one database to another may cause the capitalization of table names to change.

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

In Linux MySQL, table and column names are case-sensitive. You can configure Linux MySQL 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.8.5 PostgreSQL: Job Scheduling Error

If the Quartz settings in the PostgreSQL database aren't updated to specify the driver delegate class specific to PostgreSQL you'll get errors when you try and run a scheduled report.

The errors will look like this:

```
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, check your Quartz properties file in the following location:

```
<tomcat>/webapps/jasperserver-ce/WEB-INF/js.quartz.properties
```

Make sure the following property does not have the standard driver delegate, but instead has the PostgreSQL-specific driver delegate. It should look like the following for PostgreSQL:

```
quartz.delegateClass=org.quartz.impl.jdbcjobstore.PostgreSQLDelegate
```

A.8.6 Error Running a Scheduled Report

If you run a scheduled report and save it as HTML or RTF, the resulting report may be quite large. If you are running MySQL and get the error shown here, the problem may be the default size of the MySQL blob datatype.

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

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.8.7 Error Running a Report

If you can log into JasperReports Server but encounter an error when running a report, browse the repository to identify and resolve the problem.

One common problem with an individual report is the data source. 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 and identify the data source on the fourth edit page.
3. Edit the data source in the repository and check its settings.
4. Click the **Test Connection** button.

   If the connection fails, perhaps the application server can't find the driver for the data source. For example, in a default installation of JasperReports Server, Tomcat looks for data source drivers in <js-install>/apache-tomcat/lib.
5. Test your report. If it still returns errors, edit the data source again and try checking other values, like the port used by the database.

A.9 Application Server-related Problems

A.9.1 Memory Issues Running Under Tomcat

These steps might solve problems related to the release of memory or to container tag pooling:

1. Set the following parameter in the global $CATALINA_BASE/conf/web.xml:

   ```
   enablepooling = false
   ```
2. Restart Tomcat.

A.9.2 Java Out of Memory Error

If you encounter a Java out of memory error, try increasing your Java heap size setting. See **4.1, "Setting JVM Options for Application Servers," on page 37**. As a minimum, add `-Xms1024m -Xmx2048m` to your `JAVA_OPTS` setting.

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

A.9.3 Configuration File Locations

You'll find JasperReports Server configuration properties specific to your application server in the following files.

Tomcat:	<tomcat>/webapps/jasperserver/META-INF/context.xml	
	<tomcat>/webapps/jasperserver/WEB-INF/hibernate.properties	
	<tomcat>/apache-tomcat/webapps/jasperserver/WEB-INF/web.xml	(JNDI config)
	<tomcat>/apache-tomcat/config/Catalina/localhost/jasperserver.xml	(delete: see below)

JBoss: <jboss>/standalone/deployments//jasperserver.war/WEB-INF/js-jboss7-ds.xml
<jboss>/standalone/deploymentsjasperserver.war/WEB-INF/hibernate.properties
<jboss>/standalone/deployments/jasperserver.war/WEB-INF/web.xml
<jboss>/standalone/deployments/jasperserver.war/WEB-INF/jboss-web.xml

GlassFish: <glassfish>/domains/domain1/autodeploy/jasperserver.war/WEB-INF/hibernate.properties
<glassfish>/domains/domain1/autodeploy/jasperserver.war/WEB-INF/js.quartz.properties
<glassfish>/domains/domain1/config/domain.xml

A.9.4 Context.xml on Tomcat: Special Case

If you deploy multiple instances of JasperServer to Tomcat, the context.xml (database connection configuration) can be superseded by a file in this location: `<tomcat>/conf/Catalina/localhost/jasperserver.xml`. This is the case with some Tomcat versions before Tomcat 7.

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 it's the way Tomcat operates.

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

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

Remember to delete the jasperserver.xml file:

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

A.9.5 Tomcat Installed Using apt-get/yum

A.9.5.1 Setting CATALINA_HOME

If you're installing JasperReports Server to an instance of Tomcat that was installed using a package manager like apt-get, yum, or rpm, you can use the `CATALINA_HOME` and `CATALINA_BASE` properties found in your default_master.properties file.

Go to the section of the `default_master.properties` that looks like this:

```
# Tomcat app server root dir
appServerDir = C:\\Program Files\\Apache Software Foundation\\Tomcat 7.0
# appServerDir = /home/devuser/apache-tomcat-7.0.26
# if linux package managed tomcat instance, set two properties below
# CATALINA_HOME = /usr/share/tomcat6
# CATALINA_BASE = /var/lib/tomcat6
```

and change it to this:

```
# Tomcat app server root dir
# appServerDir = C:\\Program Files\\Apache Software Foundation\\Tomcat 7.0
# appServerDir = /home/devuser/apache-tomcat-7.0.26
# if linux package managed tomcat instance, set two properties below
CATALINA_HOME = /usr/share/tomcat6
CATALINA_BASE = /var/lib/tomcat6
```

Note that you must set both `CATALINA_HOME` and `CATALINA_BASE`.

A.9.5.2 Database Driver Location

After installing JasperReports Server, make sure there's a copy of the database driver file in the /usr/share/tomcat7/webapps/jasperserver/WEB-INF/lib directory. If it's not there, copy the driver to this location. For example, for PostgreSQL, you can copy the driver from the <js-install>/buildomatic/conf_source/db/postgresql/jdbc directory.

A.9.6 GlassFish Modifications

A.9.6.1 Using a Custom Domain

If GlassFish is your application server and you're using a custom domain, set up the following authentication information in the default_master.properties:

```
# Glassfish domain name (default is domain1)
glassfishDomain=domain1

# Glassfish domain port (default is 4848), user (default is admin) and password.
# Uncomment and set up next parameters if you install JasperServer to the custom Glassfish domain (not
default)
#glassfishPort=4848
#glassfishUser=admin
#AS_ADMIN_PASSWORD=adminadmin
```

A.9.6.2 Requests to Single Permissions REST2 Service fail on GlassFish

Requests to Single Permissions REST2 service are failing on GlassFish with the following error:

```
400 Invalid URI: Encoded slashes are not allowed by default. To enable
encodedslashes, set the property com.sun.grizzly.util.buf.UDecoder.ALLOW_ENCODED_
SLASH to true
```

To fix this issue, run this command:

```
./bin/asadmin create-jvm-options -Dcom.sun.grizzly.util.buf.UDecoder.ALLOW_ENCODED_
SLASH=true
```

A.9.6.3 BufferOverflowException When Working With Input Controls

In some cases, adding a large number of values to an input control causes an overflow error like this:

```
Request URI is too large.
java.nio.BufferOverflowException
```

To fix this, increase the allowed URI size in the GlassFish admin console. Go to **Configurations > cluster-config > Network Config > Transports > tcp > Buffer Size** and increase the value to 131072 or more.

A.9.7 JBoss Modifications

A.9.7.1 JBoss 7 Startup Error

JBoss 7 has a default startup time period. If your JBoss 7 takes longer than 60 seconds to start or deploy, you may receive the following error:

```
"(DeploymentScanner-threads - 1) Did not receive a response to the deployment
operation within the allowed timeout period [60 seconds]. Check the server
configuration file and the server logs to find more about the status of the
deployment".
```

To fix this, you need to increase your `deployment-timeout` setting as follows:

1. Change to the JBoss standalone configuration directory.

   ```
   cd <jboss>/standalone/configuration
   ```

2. Open the standalone.xml file.
3. Look for the `<subsystem xmlns="urn:jboss:domain:deployment-scanner:1.1">` element, for example:

   ```
   <subsystem xmlns="urn:jboss:domain:deployment-scanner:1.1">
       <deployment-scanner path="deployments" relative-to="jboss.server.base.dir" scan-
       interval="5000"/>
   </subsystem>
   ```

4. Edit this to add or set the attribute `deployment-timeout` to the preferred time in seconds, for example:

   ```
   <subsystem xmlns="urn:jboss:domain:deployment-scanner:1.1">
       <deployment-scanner path="deployments" relative-to="jboss.server.base.dir" scan-
       interval="5000" deployment-timeout="600"/>
   </subsystem>
   ```

5. Save the file.

   On server restart, your system will have the specified time to start up.

A.9.7.2 JBoss 7 ReservedCodeCacheSize Error

If you get a fatal error like this:

"out of space in CodeCache for adapters"

It may be the result of a too-low a memory setting for the `ReservedCodeCacheSize` flag. This error has been observed when running the Oracle JDK, version 1.6 (no longer supported).

You can set a higher value as shown in the example below:

A.9.7.2.1 Linux

export JAVA_OPTS="$JAVA_OPTS -DReservedCodeCacheSize=128m"

A.9.7.2.2 Windows

set JAVA_OPTS=%JAVA_OPTS% -DReservedCodeCacheSize=128m

A.9.7.3 JBoss Large INFO Log Message on Drill-through

JBoss has an internal mechanism to track and log information on unclosed JDBC connections. Jaspersoft OLAP Views leaves a connection open for performance reasons when doing a drill-through. In this case, JBoss puts a

large INFO level message in 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 this value:

```
<category name="org.jboss.resource.connectionmanager.CachedConnectionManager">
<priority value="WARN"/>
</category>
```

A.9.7.4 Using a Non-default JBoss Profile

If JBoss is your application server, and you're using a profile other than the default, you need to set the jboss.profile property before running the js-install script in **3.2, "Installing the WAR File Using js-install Scripts," on page 30**:

1. Open this buildomatic property file:
   `<js-install>/buildomatic/build_conf/default/app.srv.properties`
2. Uncomment the jboss.profile property and change the profile name as follows:
   from
   ```
   # jboss.profile = default
   ```
   to
   ```
   jboss.profile = <your_profile>
   ```

A.9.7.5 Using JBoss with Non-Latin Characters

If JBoss is your application server, and your organization is created with non-Latin characters, you will need to edit the standalone.xml configuration file.

1. Edit `<jboss-home>/standalone/configuration/standalone.xml`
2. Add a new `<system-properties>` tag after the `<extensions>` tag:

```
<extensions>

......
</extensions>

<system-properties>
  <property name="org.apache.catalina.connector.URI_ENCODING" value="UTF-8"/>
  <property name="org.apache.catalina.connector.USE_BODY_ENCODING_FOR_QUERY_STRING" value="true"/>
</system-properties>
```

A.9.7.6 Maximum Post Size in Wildfly

If you're upgrading or importing on some versions of Wildfly and your repository or other import file is large, the import may fail and the connection may be reset. In this case, you may need to set `max-post-size`. To do this, open the file <wildfly-home>/wildfly/standalone/configuration/standalone.xml and add or change the `max-post-size` attribute of the `http-listener` property, for example:

```
<http-listener name="default" socket-binding="http" max-header-size="974247881"
     max-post-size="974247881"/>
```

A.9.7.7 Performance Issues with JBoss EAP 6.1

A known defect in JBoss EAP 6.1 on Windows causes significant performance issues with JasperReports Server. The workaround is to remove the `-XX:+TieredCompilation` option from the <jboss-install>\bin\standalone.bat file. This defect has been fixed in later versions of JBoss EAP.

A.9.8 Disabling User Session Persistence in Application Servers

JasperReports Server stores non-serializable data in its user sessions, which can cause errors after restarting your application server:

```
Exception loading sessions from persistent storage
Cause: java.io.NotSerializableException ...
```

The errors appear in the JasperReports Server log when users log in after the application server has been restarted. Users don't see the errors, 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, 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 activate the change. For other application servers, refer to the product documentation.

A.9.9 Session Error Using JasperReports Server and Tomcat 7

On some versions of Tomcat 7, a session error might occur while running reports, with the log error “A request has been denied as a potential CSRF attack.” This is due to a known conflict between security settings in Direct Web Remote library (DWR) 2.x and some versions of Tomcat 7.0.x:

- Tomcat 7 sets httpOnly on session ID cookies to safeguard against cross-site scripting (XSS) attacks.
- DWR 2.x uses session ID cookies to safeguard against cross-site request forgery (CSRF).

To work around this problem, you must modify these safeguards by doing one of the following:

- Disabling httpOnly for cookies in Tomcat
  OR
- Allowing requests from other domains in DWR

For more information on the security impact and relative risks of these two choices, see, for example, the Cross-site Scripting and Cross-site Request Forgery pages at the Open Web Application Security Project (OWASP).

A.9.9.1 Disabling httpOnly for Cookies in Tomcat

The application server hosting JasperReports Server handles the session cookie. To prevent malicious scripts on a client from accessing the session cookie and the user connection, Tomcat 7 is set to use httpOnly cookies. This tells the browser that only the server may access the cookie, not scripts running on the client. When enabled, this setting safeguards against XSS attacks.

You can disable this by setting httpOnly in the file <tomcat>/conf/context.xml:

```
<Context useHttpOnly="false">
...
</Context>
```

A.9.9.2 Allowing Requests from Other Domains in DWR

DWR is a server-side component used for Input Controls. By default, DWR uses session ID cookies to prevent cross-site request forgery. You can disable the protection in DWR by setting the `crossDomainSessionSecurity` parameter for the `dwr` servlet in the file <tomcat>\webapps\jasperserver\WEB-INF\web.xml:

```
<servlet>
    <servlet-name>dwr</servlet-name>
<servlet-class>org.directwebremoting.spring.DwrSpringServlet</servlet-class>
    ...
    <init-param>
      <param-name>crossDomainSessionSecurity</param-name>
      <param-value>false</param-value>
    </init-param>
  </servlet>
```

A.10 Problems Importing and Exporting Data from the Repository

A.10.1 Exporting a Repository That Contains UTF-8

You may see the following errors when you have international characters in repository objects, for example, in user IDs.

A.10.1.1 Error During Export

An Upgrade usually requires exporting your database. If you're using MySQL and getting this null pointer exception, it may be caused by an incorrect character in the `js.jdbc.properties` file:

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

Check the URL in this file in <js-install>buildomatic/build_conf/default/; it should look like this:

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

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

APPENDIX B MANUALLY CREATING THE JASPERREPORTS SERVER DATABASE

If you can't use the `js-install` scripts to create the JasperReports Server database and the sample databases, you can create them manually. Follow the instructions for your database to create the repository database and optional sample databases:

- **PostgreSQL**
- **MySQL**

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

B.1 PostgreSQL

To manually create the JasperReports Server database in PostgreSQL:

1. On the Windows, Linux, or Mac command line, enter these commands:

```
cd <js-install>/buildomatic/install_resources/sql/postgresql
psql -U postgres -W
postgres=#create database jasperserver encoding='utf8';
postgres=#\c jasperserver;
postgres=#\i js-create.ddl
postgres=#\i quartz.ddl
postgres=#\q
```

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

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

3. If you didn't install the optional sample databases, complete the installation with these commands:

```
cd <js-install>/buildomatic
js-ant import-minimal-ce
js-ant deploy-webapp-ce
```

   If you installed the optional sample databases, complete the installation with these commands:

```
cd <js-install>/buildomatic
js-ant import-sample-data-ce
js-ant deploy-webapp-ce
```

   For more information about executing the Ant scripts, see **3.6, "Installing the WAR File Manually," on page 35**.
4. Set Java JVM Options (required), as described in **4.1, "Setting JVM Options for Application Servers," on page 37**.

B.2 MySQL

To manually create the JasperReports Server database in MySQL:

You can use the MySQL client software, `mysql.exe` or `mysql`, to interact with the MySQL database.

For specific details on connecting to the MySQL database and setting privileges for databases and db users, please refer to the documentation provided with your database.

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

```
cd <js-install>/buildomatic/install_resources/sql/mysql
mysql -u root -p
mysql>create database jasperserver character set utf8;
mysql>use jasperserver;
mysql>source js-create.ddl
mysql>source quartz.ddl
mysql>exit
```

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

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

3. If you didn't install the optional sample databases, complete the installation with these commands:

```
cd <js-install>/buildomatic
js-ant import-minimal-ce
js-ant deploy-webapp-ce
```

   If you installed the optional sample databases, complete the installation with these commands:

```
cd <js-install>/buildomatic
js-ant import-sample-data-ce
js-ant deploy-webapp-ce
```

   For more information about executing the Ant scripts, see **3.6, "Installing the WAR File Manually," on page 35**.
4. Set Java JVM Options (required), as described in **4.1, "Setting JVM Options for Application Servers," on page 37**.