JasperReports Server Community Project Installation Guide

Release 5.6.0

http://www.jaspersoft.com

This is version 0614-JSO56-31 of the *JasperReports Server Community Project Installation Guide*.

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION

JasperReports Server builds on 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 providing shared services, such as security, a repository, and scheduling.

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

- Easily view and explore your data in the web-based drag-and-drop Ad Hoc Editor interface.
- Efficiently and securely manage many reports.
- Interact with reports, including sorting, filtering, formatting, entering parameters and drilling on data.
- Arrange reports and web content to create appealing, data-rich Jaspersoft Dashboards that quickly convey business trends.

Jaspersoft OLAP is an optional component of JasperReports Server, controlled by licence and described in its own user guide.

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

- Our Ultimate Guides document advanced features, best practices, and numerous examples. Customers can download them freely from our community website.
- Our free Business Intelligence Tutorials let you learn at your own pace, and cover topics for developers, administrators, business users, and data integrators. The tutorials are available online in the Professional Services section of our website.

Our free samples, which are installed with JasperReports Library,Jaspersoft iReport Designer, and JasperReports Server, are documented online. The samples documentation can be found on our community website.

This chapter contains the following sections:

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

1.1 Conventions

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

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

1.2 Java Version Supported

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

JasperReports Server is tested and certified using Oracle/Sun Java. OpenJDK 1.6 has also been certified to run with JasperReports Server.

1.3 JasperReports Server Distributions

There are two main distribution packages for JasperReports Server.

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

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

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

1.3.1 Installer Support

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

Platform	Versions supported
Linux	Red Hat Enterprise Linux 5, 6 Novell SUSE Linux Enterprise 10, 11 Debian 6 Ubuntu 10
Mac OSX	10.6 (Snow Leopard) 10.7 (Lion)
Windows	Windows 2008 Windows 7 Windows 8

1.3.1.1 Installer Naming for 32-bit and 64-bit

Native 32- and 64-bit installers are supported. The 64-bit installer will put 64-bit versions of Java 7 and PostgreSQL 9 onto your system for increased speed and performance.

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

Installer Type	Naming
32-bit installer	`jasperreports-server-cp-5.6.0-windows-x86-installer.exe` `jasperreports-server-cp-5.6.0-linux-x86-installer.run`

64-bit installer	`jasperreports-server-cp-5.6.0-windows-x64-installer.exe`
	`jasperreports-server-cp-5.6.0-linux-x64-installer.run`
	`jasperreports-server-cp-5.6.0-osx-x64-installer.app.zip`

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

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

1.3.1.2 Installer Distribution Components

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

Component	Description
JasperReports Server Application	WAR file and configuration support scripts.

Component	Description
JasperReports Server Documentation	Found in the <js-install>/docs directory.
Apache Tomcat 7	Web application container. You can use the bundled version or an existing version.
Java 1.7 Runtime	Runs the web application container.
PostgreSQL 9 Database	Database server. You can use the bundled version or an existing version.

1.3.1.3 Installing with Existing Components

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

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

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

If you use an existing PostgreSQL, it can be on a local or remote machine. If it's on a remote Linux machine, configure PostgreSQL to allow remote connections as described in **2.7.4, "Enabling Connections to a Remote Host," on page 18**.

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

1.3.1.4 Running Components as Windows Services

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

Control Panel > System and Security > Administrative Tools > Services

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

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

- `jasperreportsPostgreSQL`
- `jasperreportsTomcat`

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

1.3.2 WAR File Binary Distribution Support

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

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

The target database can be on a remote server. Using a remote PostgreSQL database on some Linux platforms requires a change to its configuration file, as described in **2.7.4, "Enabling Connections to a Remote Host," on page 18**.

The application server should reside on the local machine.

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

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

The main contents of the WAR file binary distribution are:

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

1.3.2.1 About Bundled Apache Ant

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

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

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

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

On Linux and Solaris, the js-ant commands may not be compatible with all shells. If you have errors, use the `bash` shell explicitly. For more information, see **A.4, “Bash Shell for Solaris, IBM AIX, HP UX and FreeBSD,” on page 59**.

1.4 Release Notes

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

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

1.5 System Requirements

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

Resource	Footprint	Minimum	Recommended
Disk	~1.3 Gigabytes	10GB free	40GB +
RAM		4GB	8GB +
Processor		2 core minimum	2.5GHz + multi-core Pentium for Windows, Mac, and Linux

1.6 Support for Internationalization

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

CHAPTER 2 INSTALLING JASPERREPORTS SERVER

This chapter contains the following sections:

2.1 Pre-Installation Steps

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

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

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

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

2.2 Starting the Installer

In Windows, the installer is an executable file that you can double-click to run. The installer under Windows will need a user account with Administrative privileges. The installer should also be started and run with “Run as administrator”. For example, right click on the binary installer file to bring up the context menu. Choose “Run as administrator”:

`jasperreports-server-cp-5.6.0-windows-x86-installer.exe` (32 bit)

`jasperreports-server-cp-5.6.0-windows-x64-installer.exe` (64 bit)

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:

`./jasperreports-server-cp-5.6.0-linux-x86-installer.run` (32 bit)

`./jasperreports-server-cp-5.6.0-linux-x64-installer.run` (64 bit)

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

`jasperreports-server-cp-5.6.0-osx-x64-installer.app` (64 bit only)

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

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

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

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”.

2.3 Accepting the License Agreement

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

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

2.4 Choosing Installation Type

As of the 5.6.0 Release, the installer has been updated to offer a choice of “Install Type”. The first option will install all installer components and sample data resources. The second option is the “Custom Install”. With the custom install, you can choose which components to install and whether to include sample data resources.

Install All Components and Samples Option:

This option will copy a Bundled version of the Apache Tomcat package and a Bundled version of the PostgreSQL database to your file system. Additionally, all sample data resources (Reports, Data Sources, OLAP Views, etc) are added to your JasperReports Server and additional sample databases are created. With this install option, the installer will attempt to find open Tomcat ports in the 8080 and higher range. And for the PostgreSQL port, the installer will start with port 5432 and then try values higher than this if 5432 is already being used.

After you choose this first option, you can next choose the installation directory for JasperReports Server. Then, next, all files and components can be installed without requiring any further information.

Custom Install:

With the custom install, you will have the same choices that the installer has had in the past: 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 to install sample data resources or not.

2.5 Choosing an Installation Directory

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

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

Windows:	C:\Jaspersoft\jasperreports-server-cp-5.6.0
Linux:	<USER_HOME>/jasperreports-server-cp-5.6.0
Linux (as root)	/opt/jasperreports-server-cp-5.6.0
Mac OSX	/Applications/jasperreports-server-cp-5.6.0

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

- **I want to use the bundled Tomcat**

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

- **I want to use an existing Tomcat**

  If you already have an instance of Tomcat on your system, you can choose this option. Later, after choosing a bundled or existing database, you are prompted for the location of Tomcat. You can browse to the folder where you installed Tomcat, such as C:\Apache Software Foundation\Tomcat 7.

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

2.7 Selecting a PostgreSQL Configuration

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

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

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

Parameter	Default Value and Description
Binary Directory	The directory where the postgres and pgAdmin3 binaries are located.
Port	The port number that PostgreSQL uses (default is 5432). User must choose an alternate port if 5432 is in use.
IP or Host Name	The IP address or name of the machine where PostgreSQL is installed. The default value is 127.0.0.1.
PostgreSQL Administrative Password	Password of the database administrative user: postgres. The installer cannot handle special characters at the end of a password string. Incompatible characters include: & ; $
Database User Name	Hard coded default: jasperdb - The installer creates this user which is used to connect to the JasperReports Server database
Database User Password	Hard coded default: password - The installer uses this password for the jasperdb user.
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 the option to use an existing PostgreSQL database, you are eventually prompted for the location of PostgreSQL and the port to use. If you have an instance of PostgreSQL installed locally, accept the default, which is 127.0.0.1, the localhost. Accept the default location for the PostgreSQL \bin directory, or click **Browse** to locate and select another location. You are also prompted for the default administrative account password of the PostgreSQL administrative user. The database administrative user account name postgres is used by default. Enter the database administrative user password and click **Enter**.

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

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

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

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

2.7.3 Using an Existing PostgreSQL on a Remote Host

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

```
psql --version
```

or

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

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

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

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

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

or

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

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

2.7.4 Enabling Connections to a Remote Host

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

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

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

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

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

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

   ```
   host all 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 17**.

2.8 Installing Sample Data

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

- 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.
- Jaspersoft strongly recommends that you install this data, unless you are not interested in testing or evaluating with the default sample data.

During installation, the following prompt appears:

Would you like to install sample databases and sample reports?

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

2.9 Completing the Installation

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

- **View Release Notes** - If you choose to view the release notes, you must exit the release notes text viewer before JasperReports Server will launch and open a browser (if you have chosen that option below).
- **Launch JasperReports Server Now** - If you choose to launch JasperReports Server from the installer, the installer exits and the application server starts if you chose the bundled Tomcat and PostgreSQL. A pause that lasts approximately 25 seconds occurs as the server starts up, then the login page appears in your system default browser. If you're installing under Linux, do not close the terminal window running the start script. For information about logging in, see **3.4, "Logging into JasperReports Server," on page 26**.

> The **Launch JasperReports Server Now** check box option will only be displayed if you have chosen to install a bundled Tomcat and a bundled PostgreSQL. The menu based start/stop scripts only control the bundled applications that you chose to be installed. For more information, see **Chapter 3, "Starting and Stopping JasperReports Server," on page 23**.
>
> Additionally, if you do not choose to Launch JasperReports Server Now the bundled components will not be started. If you only have one bundled component this component will not be started unless you use the Start/Stop menus or scripts. To Start and Stop JasperReports Server see **Chapter 3, "Starting and Stopping JasperReports Server," on page 23**.

- **Opt-in for JasperServer Heartbeat** - When the heartbeat is enabled, the server sends anonymous system and version information to Jaspersoft using HTTPS. JasperReports Server heartbeat information helps Jaspersoft create better products by improving our understanding of customer installation environments. For more information, see **5.4.1, "JasperReports Server Heartbeat," on page 35**.

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

You should now be ready to log into the server.

2.10 Post-Installation Steps

2.10.1 Updates Made by the Installer During Installation

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

Updates made to the application server

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

File or Directory	Updates
Windows: bin/setclasspath.bat Linux and Mac OSX: bin/setclasspath.sh	Modifies JAVA_OPTS to add -Djs.license.directory. (Commercial installer only)

File or Directory	Updates
Windows: bin/setenv.bat Linux and Mac OSX: bin/setevn.sh	Creates this file. Sets increased Java memory allocation values to JAVA_OPTS. For additional settings, refer to **6.1, "Setting JVM Options for Application Servers," on page 39**.
Tomcat 5: common/lib Tomcat 6 and 7: lib	Adds PostgreSQL JDBC driver to this directory. As of 5.1, add additional JDBC drivers for other databases.

Updates made to the PostgreSQL database

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

PostgreSQL Updates	Description
Database `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 Checking your Java JVM Options

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

Installer Type	Setting	File Location
32 bit (x86) Windows	-Xms512m -Xmx1024m -XX:MaxPermSize=512m	<js-install>/apache-tomcat/bin/service.bat
32 bit (x86) Linux and Mac OSX	-Xms512m -Xmx1024m -XX:MaxPermSize=512m	<js-install>/apache-tomcat/scripts/ctl.sh

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

CHAPTER 3 STARTING AND STOPPING JASPERREPORTS SERVER

This chapter contains the following sections:

- **Start/Stop Menu — Windows**
- **Start/Stop Scripts — Linux**
- **Start/Stop Apps — Mac OSX**
- **Logging into JasperReports Server**
- **JasperReports Server Log Files**

3.1 Start/Stop Menu — Windows

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

3.1.1 Start/Stop Menus — Bundled Tomcat and PostgreSQL

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

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

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

3.1.2 Additional Information about the Bundled Tomcat and PostgreSQL

JasperReports Server Windows Service Names:

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

- jasperreportsPostgreSQL
- jasperreportsTomcat

Preventing JasperReports Server from starting up automatically:

By default, the bundled services are started automatically on a reboot. Consequently, the JasperReports Server also automatically starts. You can 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:

JasperReports Server can be manually started from a Windows Command Shell:

- Open a Windows CMD Shell
- Navigate to the root of the <js-install> folder (C:\Jaspersoft\jasperreports-server-cp-<ver>)
- 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

3.1.3 Start/Stop Scripts — No Bundled Applications

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

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

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

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

3.2 Start/Stop Scripts — Linux

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

3.2.1 Manual Start/Stop

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

Start JasperReports Server:

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

```
./ctlscript.sh start
```

Stop JasperReports Server:

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

```
./ctlscript.sh stop
```

To start and stop individual components:

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

3.2.2 Auto Start/Stop with Bundled Tomcat and PostgreSQL

To have JasperReports Server automatically start when you reboot your Linux server, you need to install the 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, an example jasperserver service script can be found in the following location:

<js-install>/scripts/linux/jasperserver

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

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

3.3 Start/Stop Apps — Mac OSX

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

/Applications/jasperreports-server-cp-<ver>

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

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

jasperServerStart.app

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

jasperServerStop.app

The Mac lists the following information in the Activity Monitor:

- java

  or

  org.apache.catalina.startup.Bootstrap
- postgres

3.3.1 Start/Stop Apps — Mac Dock

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

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

3.3.2 Start/Stop JasperReports Server — Mac Terminal Shell

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

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

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

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

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

3.4 Logging into JasperReports Server

To log into JasperReports Server on any operating system:

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

   http://<hostname>:8080/jasperserver

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

   The login page appears.

4. Log in using the following credentials:

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

If you installed the sample data, these additional sample end-users are also created. These users are non-administrative users who have fewer system privileges than an administrative user.

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-<ver>/jasperServerLogin

To use the Dock to log into JasperReports Server:

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

3.5 JasperReports Server Log Files

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

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

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

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

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

   The Log Settings page appears.

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

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

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

CHAPTER 4 UNINSTALLING JASPERREPORTS SERVER

This chapter contains the following sections:

- **Windows**
- **Linux**
- **Mac OSX**
- **Uninstall Survey**

4.1 Windows

To uninstall JasperReports Server on Windows 7:

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

4.2 Linux

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

To uninstall JasperReports Server:

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

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

   ```
   ./uninstall
   ```

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

4.3 Mac OSX

To use Finder to uninstall JasperReports Server:

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

   For example: /Applications/jasperreports-server-cp-<ver>

2. Click the uninstall.app to launch the uninstaller.

4.4 Uninstall Survey

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

Chapter 5 Installing the WAR File Distribution

In addition to the installer binaries, you can install the JasperReports Server application using the stand-alone WAR file distribution. For production environments, use the WAR file distribution. Download the WAR file distribution from http://community.jaspersoft.com. The WAR file distribution comes in a file named jasperreports-server-cp-5.6.0-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 JasperReports Server**
- **Logging into JasperReports Server**
- **Troubleshooting Your JasperReports Server Configuration**
- **Installing the WAR File Manually**

5.1 Applications Supported by the WAR File Distribution

5.1.1 Database and Application Server Support

The instructions support the following configurations:

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

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

5.1.2 Operating System Support for Bash Shell

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

Operating System	Required Shell for 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

5.2 Installing the WAR File Using js-install Scripts

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

Prerequisites for installing the WAR file:

1. Install the Oracle/Sun Java JDK 1.6 or 1.7. OpenJDK 1.6 has also been certified.
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 5.5, 6, or 7
   - JBoss 5.1 or 7.1
   - Glassfish 2.1 or 3.0 using the default domain (domain1)

     If you use GlassFish 3.1.0 or a custom domain, see **A.9.6, "GlassFish Modifications," on page 67**.
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.

If you would like to run a pre-install validation test, you can run `js-install-ce.bat test` or a similar command. For more information about how to perform a validation test in your environment, see **5.5.3.1, "js-install Script Test Mode," on page 36**.

To install the WAR file using js-install scripts:

The scripts are intended for the bash shell.

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

1. Extract all files from `jasperreports-server-cp-5.6.0-bin.zip`. Choose a destination, such as `C:\Jaspersoft` on Windows, `/home/<user>` on Linux, or `/Users/<user>` on Mac.

   The directory, `jasperreports-server-cp-5.6.0-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 5-1** lists sample property values for each supported database.

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

Database	Sample Property Values

PostgreSQL	`appServerType=tomcat6 [tomcat7, tomcat5, jboss, jboss-eap-6, jboss-as-7, glassfish2, glassfish3, skipAppServerCheck]`
	`appServerDir=c:\\Program Files\\Apache Software Foundation\\Tomcat 7`
	`dbHost=localhost`
	`dbUsername=postgres`
	`dbPassword=postgres`

MySQL	`appServerType=tomcat6 [tomcat7, tomcat5, jboss, jboss-eap-6, jboss-as-7, glassfish2, glassfish3, skipAppServerCheck]`
	`appServerDir=c:\\Program Files\\Apache Software Foundation\\Tomcat 7`
	`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.

If, under Linux, Tomcat is installed using apt-get, yum, or rpm, see **A.9.5, “Tomcat 6 Installed Using apt-get,” on page 67**.

5. Password Encryption

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

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

```
encrypt=true
```

For more information of the Encryption functionality, refer to the *JasperReports Server Administrator 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 does not install sample data and sample databases

   If you encounter errors during the `js-install` script execution, see **5.5.3, "Error Running js-install Scripts (js-install-ce.bat/sh)," on page 36**.

7. Set Java JVM Options, as described in **6.1, "Setting JVM Options for Application Servers," on page 39**. This step is required.

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

5.3 Starting JasperReports 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/run.bat
	Linux and Mac OSX	<jboss>/bin/run.sh
GlassFish:	Windows, Linux, and Mac OSX	asadmin start-domain domain1

To view the JasperReports Server application logs, see **3.5, "JasperReports Server Log Files," on page 27**.

5.4 Logging into JasperReports Server

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

http://<hostname>:8080/jasperserver

Example:

http://localhost:8080/jasperserver

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

The login page appears after compiling the necessary JSP 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.

5.4.1 JasperReports Server Heartbeat

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

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

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

- Operating System 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 Administrator Guide*.

5.5 Troubleshooting Your JasperReports Server Configuration

This section describes the most common installation problems.

5.5.1 JasperReports Server Startup Problems

If you encounter a problem trying to run a new JasperReports Server, an incorrect database configuration is the likely culprit. Another common cause is a mistake in the application server configuration files. For information about resolving these types of errors, see **Appendix A, "Troubleshooting," on page 57**.

5.5.2 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 57**.

5.5.3 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.

5.5.3.1 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
   ```

5.5.4 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.

5.6 Installing the WAR File Manually

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

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

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 **5.2, "Installing the WAR File Using js-install Scripts," on page 32**.
4. Open a Command Prompt as Administrator on Windows or open a terminal window on Linux or Mac. Run the following commands:

Table 5-2 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.

Commands	Description

`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 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.

`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 59**.

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 **6.2, "Manually Creating the JasperReports Server Database," on page 43**.

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

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

Commands	Description

`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, as described in **6.1, "Setting JVM Options for Application Servers," on page 39**. This step is required.

CHAPTER 6 JVM OPTIONS AND MANUAL DATABASE CREATION

This chapter contains the following sections:

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

6.1 Setting JVM Options for Application Servers

JasperReports Server is supported on Java 1.6 and 1.7. Java Virtual Machine (JVM) runtime parameters normally need to be explicitly set so that the memory settings have values that are larger than the default settings. The options that you should set and the values they are set to depend on your version of Java and the application server that you use.

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

6.1.1 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 **"Checking your Java JVM Options" on page 20**.

JVM Options on Windows (64 bit)	
Options for Java 1.6 and 1.7	set JAVA_OPTS=%JAVA_OPTS% -Xms1024m -Xmx2048m -XX:PermSize=32m set JAVA_OPTS=%JAVA_OPTS% -XX:MaxPermSize=512m -Xss2m -XX:+UseConcMarkSweepGC set JAVA_OPTS=%JAVA_OPTS% -XX:+CMSClassUnloadingEnabled

JVM Options on Windows (64 bit)	
Additional options for Java 1.6-1.7 and 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 5.1	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 Java 1.6 and 1.7	export JAVA_OPTS="$JAVA_OPTS -Xms1024m -Xmx2048m -XX:PermSize=32m" export JAVA_OPTS="$JAVA_OPTS -XX:MaxPermSize=512m -Xss2m" export JAVA_OPTS="$JAVA_OPTS -XX:+UseConcMarkSweepGC" export JAVA_OPTS="$JAVA_OPTS -XX:+CMSClassUnloadingEnabled"
Additional options for Java 1.6-1.7 and 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="$JAVA_OPTS -Djavax.xml.soap.SOAPFactory=org.apache.axis.soap.SOAPFactoryImpl"
Additional option for JBoss 5.1	export JAVA_OPTS="$JAVA_OPTS -Djavax.xml.transform.TransformerFactory=org.apache.xalan.processor.TransformerFactoryImpl"

There are a number of ways to set JVM options. Sections 6.1.2 - 6.1.5 present step-by-step instructions for performing this task. Alternatively, you can add your `JAVA_OPTS` settings to any one of these files:

File	Add JVM Options After This Line on Windows
<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/run.bat (JBoss 5.1)	`set JAVA_OPTS=%JAVA_OPTS% -Dprogram.name=%PROGNAME%`
<jboss>/bin/standalone.bat (JBoss 7)	`rem Setup JBoss specific properties`

File	Add JVM Options After This Line on Linux
<tomcat>/bin/setclasspath.sh	`JAVA_ENDORSED_DIRS="$BASEDIR"/common/endorsed`

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

6.1.2 Changing JVM Options for Bundled Tomcat as a Windows Service

The Windows binary installer installs the bundled Tomcat component as a Windows Service by default. The steps to change JVM options are:

1. Open this file for editing:

   ```
   cd <js-install>/apache-tomcat/bin/service.bat
   ```

2. Look for the following line to change the JVM heap size, for example:

```
"%EXECUTABLE%" //US//%SERVICE_NAME% --Startup auto --JvmOptions "-Xms1024M;-Xmx2048M;-Xss2M;-Dca-
talina.base=%CATALINA_BASE%;-Dcatalina.home=%CATALINA_HOME%;-Djava.endorsed.dirs=%CATALINA_HOME%\-
endorsed" --StartMode jvm --StopMode jvm
```

3. Update this line to increase the maximum heap size from 2048M to 3072M, for example:

   ```
   -Xmx3072M
   ```

4. Because Tomcat is installed as a service, you need to re-install the service. From a Windows Command shell, enter these commands (Note: the cmd shell will disappear when these commands are run. You need to open a new cmd shell for each command.). To open a cmd shell: `Start Menu > Run... > cmd`

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

The Tomcat service is removed and then installed. After execution of the commands, the service is running.

6.1.3 Changing JVM Options for Existing Tomcat as a Windows Service

If you installed JasperReports Server to use an existing Tomcat (not the bundled component) that is 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 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, add options as follows:

These example settings are for 64-bit systems. For 32-bit systems, see **"Checking your Java JVM Options" on page 20**.

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

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

6.1.4 Changing JVM Options for Bundled Tomcat on Linux

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

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:

```
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 to 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 **Chapter 3, "Starting and Stopping JasperReports Server," on page 23**.

6.1.5 Changing GlassFish JVM Options

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

6.1.5.1 Setting GlassFish JVM Options with asadmin Command

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

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

2. Restart the application server using the following commands:

   ```
   asadmin stop-domain domain1
   ```

   ```
   asadmin start-domain domain1
   ```

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

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

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

6.1.5.2 Setting GlassFish JVM Options by Editing domain.xml

1. Open the `<glassfish>/domains/domain1/config/domain.xml` configuration file for editing.
2. Add the following lines to the section entitled `java-config`:

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

</jvm-options>
```

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

   ```
   asadmin stop-domain domain1
   ```

   ```
   asadmin start-domain domain1
   ```

6.2 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:

- **6.2.1, "PostgreSQL," on page 43**
- **6.2.2, "MySQL," on page 44**

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

6.2.1 PostgreSQL

To manually create the JasperReports Server database in PostgreSQL:

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

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

2. (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-postqresql.sql; (first make sure the file is unzipped)
postgres=#\i supermart-update.sql;
postgres=#\q
```

3. If you did not 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 **5.6, “Installing the WAR File Manually,” on page 37**.

4. Set Java JVM Options, as described in **6.1, “Setting JVM Options for Application Servers,” on page 39**. This step is required.

6.2.2 MySQL

To manually create the JasperReports Server database in MySQL:

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

> 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 these commands to create and initialize the JasperReports Server database:

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

```
mysql -u root -p
mysql>create database jasperserver character set utf8;
mysql>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 did not 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 **5.6, “Installing the WAR File Manually,” on page 37**.

4. Set Java JVM Options, as described in **6.1, “Setting JVM Options for Application Servers,” on page 39**. This step is required.

6.3 Working With JDBC Drivers

6.3.1 Changing the JDBC Driver Deployed by Buildomatic

For open source JDBC drivers, buildomatic is setup so that there will be a single default driver used. If you would like to use a driver other than the default driver, you can modify the buildomatic property files that control which JDBC driver is used as the default.

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

If you would like 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.

6.3.1.1 PostgreSQL Example

In the buildomatic/conf_source/db/postgresql/jdbc folder there are the following driver files:

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

If, for instance, you would like to change the default driver used by PostgreSQL from type jdbc4 to jdbc3, you can make the following changes:

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.

6.3.1.2 MySQL Example

In the `buildomatic/conf_source/db/mysql/jdbc` folder there is the following driver file:

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

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

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

Next, you would 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
```

6.3.1.3 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 6 and 7:	<tomcat>/lib
JBoss 5	<jboss>/server/default/lib
GlassFish:	<glassfish>/domains/domain1/lib/ext

6.4 Locating and Changing Buildomatic Configuration Files

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

6.4.1 Regenerating Buildomatic Settings

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

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

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

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

```
cd <js-install>/buildomatic

js-ant clean-config

js-ant gen-config
```

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

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

6.4.2 Locating Buildomatic-Generated Property Files

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

The generated property files are in this location:

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

Some of the key configuration files are:

```
js.jdbc.properties
```

`js.quartz.properties`

`js-glassfish-ds.xml`

`js-jboss-ds.xml`

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

More generated property files are:

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

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

`META-INF/context.xml`

`WEB-INF/hibernate.properties`

`WEB-INF/js.quartz.properties`

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

6.4.3 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 takes the war archive and unpacks it into your application server. Next, the database configuration files needed by the application server are copied to the appropriate locations. For instance, in the case of Tomcat:

- <js-install>/jasperserver.war

  Unpacked and copied to <tomcat>/webapps/jasperserver/*
- <js-install>/buildomatic/build_conf/default/webapp/META-INF/context.xml

  Copied to <tomcat>/webapps/jasperserver/META-INF/context.xml
- <js-install>/buildomatic/build_conf/default/webapp/WEB-INF/hibernate.properties

  Copied to <tomcat>/webapps/jasperserver/WEB-INF/hibernate.properties
- <js-install>/buildomatic/build_conf/default/webapp/WEB-INF/js.quartz.properties

  Copied to <tomcat>/webapps/jasperserver/WEB-INF/js.quartz.properties
- <js-install>/buildomatic/build_conf/db/postgres/jdbc/postgresql-9.2-1002.jdbc4.jar

  Copied to <tomcat>/lib

6.4.4 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-5.5.0-5.6.0-ce.sql

<js-install>/buildomatic/install_resources/sql/postgresql/js-drop.ddl

<js-install>/buildomatic/install_resources/sql/postgresql/drop-quartz.ddl

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

6.4.5 Buildomatic Location for Database Creation Scripts

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

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

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

<js-install>/buildomatic/conf_source/db/

postgresql/scripts.properties

mysql/scripts.properties

6.4.6 Buildomatic Location for Sample Data Catalog ZIP Files

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

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

Here are some key files:

js-catalog-postgresql-minimal-ce.zip

js-catalog-postgresql-ce.zip

js-catalog-mysql-minimal-ce.zip

js-catalog-mysql-ce.zip

6.4.7 Hibernate Properties Settings

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

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

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

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

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

Hibernate property values are:

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

6.4.8 Database Connection Configuration Files

6.4.8.1 Tomcat

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

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

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

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

The final location of the context.xml is:

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

In older versions of Tomcat, 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 is often a source of confusion for Tomcat users who attempt change their database settings. If you change your settings, delete the file in this location:

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

6.4.8.2 JBoss

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

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

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

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

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

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

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

<jboss>/standalone/deployments/jasperserver.war/WEB-INF/js-jboss7-ds.xml

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

For more information, refer to troubleshooting section **A.9.7, "JBoss Modifications," on page 68**.

6.4.8.3 Glassfish

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

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

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

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

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

6.5 Configuring Report Scheduling

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

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

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

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

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

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

6.5.1 Mail Server Configuration Settings

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

Configuration File	
<app-server>/<deployment>/WEB-INF/js.quartz.properties	
Property	**Description**
`report.scheduler.mail.sender.host`	The name of the computer hosting the mail server
`report.scheduler.mail.sender.username`	The name of the user in the mail server that JasperReports Server can use
`report.scheduler.mail.sender.password`	The password of the mail server user
`report.scheduler.mail.sender.from`	The address that appears in the **From** field on email notifications

`report.scheduler.mail.sender.protocol`	The protocol that the mail server uses. JasperReports Server only supports SMTP. Note: Your entry must be lower case. For example: `smtp`
`report.scheduler.mail.sender.port`	The port number that the mail server uses. For SMTP, the default is typically 25 (values other than 25 may not work in earlier 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`.

6.5.2 Database Settings for the Quartz Driver Delegate Class

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

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

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

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

Property	**Database**	**Value**
`quartz.delegateClass`	MySQL	`org.quartz.impl.jdbcjobstore.StdJDBCDelegate`
	PostgreSQL	`org.quartz.impl.jdbcjobstore.PostgreSQLDelegate`

6.5.3 Settings for the Report Scheduler Web URI

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

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

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

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

6.5.4 Settings for the Quartz Table Prefix

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

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

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

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

6.5.5 Settings for Import-Export

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

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

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

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

Change the following properties:

Property	Description
`quartz.delegateClass`	Set to the same value as described in **6.5.2, “Database Settings for the Quartz Driver Delegate Class,” on page 52**.
`quartz.tablePrefix`	Set to the same value as described in **6.5.4, “Settings for the Quartz Table Prefix,” on page 53**

6.5.6 Setting Properties in the default_master.properties File

At installation time, there is a group of properties that can be set in order to configure JasperReports Server functionality. These properties are found in the `default_master.properties` file. They can be un-commented to set them to the non-default values that will take effect upon installation. Look in the `default_master.properties` file for additional details.

Your `default_master.properties` file should be found here:

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

A sample master.properties can be found here (in the case of PostgreSQL):

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

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

6.5.6.1 Report Scheduler Email Properties

As of Release 5.5, properties have been added that will set configuration for the Report Scheduler Email properties.

The properties that can be set are the following (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
```

6.5.6.2 Diagnostic Properties

Properties which will set the configuration of 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 Administrator Guide* for more information on these settings.

6.6 Updating XML/A Connection Definitions

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

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

There are two sample connections:

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

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

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

APPENDIX A TROUBLESHOOTING

This appendix contains the following sections:

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

A.1 Binary Installer Freezes

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

A.1.1 Installer Log Files

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

You can find the installer log in the following locations:

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

If you have tried multiple installs, make sure you view the most recent install log file. Then, you can submit the installation.log to Jaspersoft Technical Support.

A.1.2 Installer DebugTrace Mode

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

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

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

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

A.2 Error Running Buildomatic Scripts

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

A.2.1 Missing Java JDK

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

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

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

If you are upgrading JasperReports Server, you can also use the Java 1.6 JDK bundled in the previous version, as described in the *JasperReports Server Upgrade Guide*.

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

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

```
BUILD FAILED
```

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

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

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

Ant is unable to accept more than one consecutive '$' character in passwords in buildomatic scripts.

This issue only occurs when two dollar signs are used in a row. For example, “$pas$word$” or “pas$word$” will not fail. If you have two consecutive dollar signs, escape each with 4 dollar signs. For example, if you use “pa$$word” you would need to set it as “pa$$$$$$$$word” in the configuration file. Once you do this, JasperReports Server will have all data connections set to “pa$$word”.

A.2.4 Older Apache Ant Version

As of the release of JasperReports Server 4.0, Apache Ant version 1.8.1 or higher is required. There are improvements to error handling routines in the buildomatic js-install scripts which required the higher level of Ant. If you are using an older version of Ant, you will get an error similar to the following:

```
BUILD FAILED

c:\js-builds\jasperserver\buildomatic\install.xml:37:

Problem: failed to create task or type componentdef
```

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

```
ant -version
```

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

```
echo $CLASSPATH
```

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

A.3 Unable to Edit Files on Windows 7

In some cases, you may want to manually edit files under your C:/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 such as the following:

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

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

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

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

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

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

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

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

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

Solaris

IBM AIX

HP UX

FreeBSD

Alternatively, you can manually run the same “buildomatic” Ant targets that are run by the `js-install` script. These Ant targets are listed in **“Troubleshooting Your JasperReports Server Configuration” on page 31**.

Also, check that you have updated your local Ant to include the `ant-contrib.jar` which supports conditional logic in Ant. The `ant-contrib.jar` is found in the location below and it should be copied to your `<ant_home>/lib` folder:

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

For updating your local Ant instance with the `ant-contrib.jar` see **A.2.2, “Forgot to Copy the File ant-contrib.jar,” on page 58**.

If you try and use the bundled ant that is 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 a Linux server does not have proper hostname entries in the `/etc/hosts` file, it is possible to get installer errors.

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

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

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

An improperly configured hosts file typically causes the log, or error messages displayed on the console, to contain error messages such as these:

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

To fix the /etc/hosts file:

1. Include entries that look similar to these:

   ```
   127.0.0.1      localhost.localdomain        localhost
   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. Also, you can 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 length to the path to the war archive is longer than the maximum allowed by Windows, you will get an error message similar to:

```
BUILD FAILED
c:\jaspers\war_file_installations\war_mysql_500\jasperreports-server-5.0-bin\buildomatic\bin\db-common.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\jdk1.6.0_38\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 will 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

Jaspersoft has seen some issues caused by the improper shutdown of the Tomcat included with the JasperReports Server. This could be related to the machine being shutdown while Tomcat is running.

When the Tomcat scripts start Tomcat, they write a .pid (Process ID) file to the Tomcat folder. Tomcat uses this to determine whether the Tomcat instance is already running. When Tomcat is shutdown, this pid file is removed. However, if the pid file is not removed on shutdown, Tomcat will fail to start up.

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

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

Delete catalina.pid using Finder:

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

   For instance: `/Applications/jasperreports-server-<ver>/tomcat/temp`
2. Delete `catalina.pid`

Delete the catalina.pid file using Terminal shell:

1. Open a Terminal shell (Finder > Go > Utilities > Terminal Icon)
2. Navigate to the `<js-install>/tomcat/temp` folder

   For instance: `/Applications/jasperreports-server-<ver>/tomcat/temp`
3. Enter the following command:

   ```
   rm catalina.pid
   ```

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

1. Open a Terminal shell (Finder > Go > Utilities > Terminal Icon).
2. Navigate to the `<js-install>` folder.

   For instance: `/Applications/jasperreports-server-<ver>`
3. To Start:

   ```
   ./ctlscript start postgresql
   ```

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

   ```
   ./ctlscript stop
   ```

   or

   ```
   ./ctlscript stop tomcat
   ```

   ```
   ./ctlscript stop postgresql
   ```

A.7.2 Installation Error on Mac OSX 10.8 (Lion)

When trying to install Jaspersoft on Mac OSX 10.8 (Lion) you might see the pop-up message “jasperreports-server-5.2-osx-x86-installer.app can't be opened because it is from an unidentified developer. Your security preferences allow installation of only apps from the Mac App Store and identified developers.”

If you see that message, perform the following procedure:

1. Right-click or Control-click the application and choose **Open**.
2. Click the **Open** button at the next dialog warning to launch the installer.

A.8 Database-related Problems

A.8.1 Database Connectivity Errors

The most common problems encountered with a new JasperReports Server instance are database configuration problems. If the connection fails, perhaps the application server cannot find the driver for the data source. For example, in a default installation of JasperReports Server, Tomcat looks for data source drivers in `<js-install>/apache-tomcat/lib`. If the driver is not there, put a copy of the driver in this directory and restart Tomcat.

This section contains information that may help resolve other connectivity issues.

A.8.1.1 Testing the Database Connection

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

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

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

A.8.1.2 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.8.1.3 Logging into MySQL

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

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

You are prompted for a password for the user you specified on the command line. Enter the appropriate password to login. The default password used in the sample configuration scripts is `password` (`jasperadmin` in 2.1 and earlier).

A.8.2 Maximum Packet Size in MySQL

If you are upgrading or importing into a MySQL database and your repository contains large objects such as images, you may see an error such as:

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

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

```
max_allowed_packet = 16M
```

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

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

A.8.3 Connection reset by peer MySQL Error

If you are using the MariaDB JDBC driver in order to connect to the MySQL database, you can still get the same error as the maximum packet size error seen in the section above. However, the error message displayed by the MariaDB driver is different than that displayed by the MySQL JDBC driver. If you get an error similar to the following error:

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

Then you should follow the instructions in the section above.

A.8.4 Case Sensitivity for Table and Column Names

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

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

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

Table and column names in PostgreSQL are case-sensitive.

A.8.5 PostgreSQL: Job Scheduling Error

If the Quartz settings under the PostgreSQL database have not been updated to specify the driver delegate class specific to PostgreSQL you will get errors when you try and run a scheduled report. The errors would look similar to the following:

```
Error while fetching Quartz runtime information
org.quartz.JobPersistenceException: Couldn't obtain triggers: Bad value for type int
org.postgresql.util.PSQLException: Bad value for type int
```

If you see this error you will need to check your Quartz properties file found at the following location:

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

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

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

A.8.6 Error Running Scheduled Report

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

```
JDBC exception on Hibernate data access
```

```
org.hibernate.exception.GenericJDBCException: could not insert
```

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

```
max_allowed_packet=32M
```

A.8.7 Error Running a Report

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

One common problem with an individual report is the data source being used. To validate a data source connection:

1. Log into JasperReports Server as a user with administrative permissions and locate the report unit that returns errors.
2. Select the report and click the **Edit** button in the toolbar to identify the data source the report uses. The data source name is found on the fourth edit page.
3. Select this data source in the repository and click the **Edit** button in the toolbar.
4. Review the information specified for this data source.
5. Click the **Test Connection** button in order to validate the connection.

   If the connection fails, perhaps the application server cannot find the driver for the data source. For example, in a default installation of JasperReports Server, Tomcat looks for data source drivers in <js-install>/apache-tomcat/lib.
6. Click **Save** or **Cancel** when you are done.
7. Test your report. If it still returns errors, edit the data source again and try checking other values, such as the port used by the database.

A.9 Application Server-related Problems

A.9.1 Memory Issues Running Under Tomcat

If you experience problems related to the release of memory or to container tag pooling, the following steps might solve the problem:

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

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

A.9.2 Java Out of Memory Error

If you encounter a Java out of memory error, try increasing your Java heap size setting. See **6.1, "Setting JVM Options for Application Servers," on page 39**. As a minimum, add `-Xms1024m -Xmx2048m` to your `JAVA_OPTS` setting.

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

A.9.3 Configuration File Locations

JasperReports Server configuration properties are found in the following files, depending on your application server.

The following list shows the location of the properties for supported application servers:

Tomcat:	<tomcat>/webapps/jasperserver/META-INF/context.xml	
	<tomcat>/webapps/jasperserver/WEB-INF/hibernate.properties	
	<tomcat>/apache-tomcat/webapps/jasperserver/WEB-INF/web.xml	(JNDI config)
	<tomcat>/apache-tomcat/config/Catalina/localhost/jasperserver.xml	(delete: see below)
JBoss 5:	<jboss>/server/default/deploy/js-postgresql-ds.xml *or* js-<database name>-ds.xml	
	<jboss>/server/default/deploy/jasperserver.war/WEB-INF/hibernate.properties	
	<jboss>/server/default/deploy/jasperserver.war/WEB-INF/web.xml	
	<jboss>/server/default/deploy/jasperserver.war/WEB-INF/jboss-web.xml	
GlassFish:	<glassfish>/domains/domain1/autodeploy/jasperserver.war/WEB-INF/hibernate.properties	
	<glassfish>/domains/domain1/autodeploy/jasperserver.war/WEB-INF/js.quartz.properties	
	<glassfish>/domains/domain1/config/domain.xml	

A.9.4 Context.xml under Tomcat: Special Case

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

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

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

Remember to delete the jasperserver.xml file:

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

A.9.5 Tomcat 6 Installed Using apt-get

If you are installing JasperReports Server to an instance of Tomcat that has been installed using a package manager such as apt-get, yum, or rpm then you can use the `CATALINA_HOME` and `CATALINA_BASE` properties found in your default_master.properties file.

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

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

and change it to the following:

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

Note that you must set both CATALINA_HOME and CATALINA_BASE.

A.9.6 GlassFish Modifications

A.9.6.1 Using a Custom Domain

If your application server is GlassFish and you're using a custom domain, set up the following authentication information in the 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 Using GlassFish 3.1.0

There is a known issue with GlassFish 3.1.0 where Java JVM options are not properly set. This issue is fixed in GlassFish 3.1.1 and later.

To set the JVM options in GlassFish 3.1.0:

1. Open this buildomatic property file:

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

2. Add the `glassfishPort` property as follows:

   `glassfishPort=4848`

A.9.6.3 Can’t Upload Files on GlassFish 3.1.2

There is a known issue with file upload on GlassFish 3.1.2 which prevents you from uploading files to the JasperReports Server (GLASSFISH 18446). This issue is resolved in the 3.1.2.2 release, which is a microrelease that resolves this and other critical issues.

A.9.6.4 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, perform the following command:

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

A.9.6.5 BufferOverflowException When Working With Input Controls

In some cases, when you adding a large number of values to an input control, you may see an overflow error such as the following:

```
Request URI is too large.

java.nio.BufferOverflowException
```

To fix this issue, 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 startup or deploy, you may receive the following error:

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

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

1. Change to the JBoss standalone configuration directory.

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

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

```
<subsystem xmlns="urn:jboss:domain:deployment-scanner:1.1">
```

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

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

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

5. Save the file.

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

A.9.7.2 JBoss 7 ReservedCodeCacheSize Error

If you get a fatal error similar to the following:

"out of space in CodeCache for adapters"

This might be the result of too low a memory setting for the ReservedCodeCacheSize flag. This error has been observed when running the Oracle JDK, version 1.6.

You can increase the value of this flag by setting a higher value such 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 into the server.log.

To silence this INFO message, perform these steps:

1. Open the JBoss log4j configuration file for editing:

   `<jboss>/server/default/conf/jboss-log4j.xml`

2. Set the logging level for the `CachedConnectionManager` class to the following value:

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

A.9.7.4 JBoss 5.0.1 and 5.1.x Error

With JBoss 5.0.1 and 5.1.x, you might see the following error:

```
org.jboss.xb.binding.JBossXBRuntimeException: Failed to create a new SAX parser
```

```
Caused by: java.lang.ClassCastException
```

This error might have occurred with older releases of JasperReports Server. Newer releases use a jar newer than 2.7.1 so this problem should not occur. Additionally, the xercesImpl-*.jar is automatically deleted when installing using the buildomatic scripts.

This is a class conflict with the xercesImpl-2.7.1.jar in JasperReports Server. To correct it, delete the following file:

```
<jboss>/server/default/deploy/jasperserver.war/WEB-INF/lib/xercesImpl-*.jar
```

When running the buildomatic scripts to deploy to JBoss, the xercesImpl-*.jar file is automatically deleted in order to fix this problem.

A.9.7.5 AttachmentStore Error in JBoss 5.1

With JBoss 5.1 you might see the following error:

```
DEPLOYMENTS IN ERROR:
```

```
 Deployment “AttachmentStore” is in error due to: java.lang.IllegalArgumentException:
Wrong arguments. new for target java.lang.reflect.Constructor expected=[java.net.URI]
actual=[java.io.File]
```

This is a known JBoss issue. The resolution is to edit the file jboss-5.1.0.GA\server\default\conf\bootstrap\profile.xml to include `java.io.File`:

```
<bean name="AttachmentStore"
      class="org.jboss.system.server.profileservice.repository.AbstractAttachmentStore">
  <constructor>
    <parameter class="java.io.File">
      <inject bean="BootstrapProfileFactory" property="attachmentStoreRoot"/>
    </parameter>
  </constructor>
  <property name="mainDeployer">
    <inject bean="MainDeployer" />
  </property>
  <property name="serializer">
    <inject bean="AttachmentsSerializer" />
  </property>
  <property name="persistenceFactory">
    <inject bean="PersistenceFactory"/>
  </property>
</bean>
```

A.9.7.6 Using a Non-default JBoss Profile

If your application server is JBoss and you’re using a profile other than the default, you need to set the jboss.profile property before running the js-install script in **5.2, “Installing the WAR File Using js-install Scripts,” on page 32**:

1. Open this buildomatic property file:

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

2. Uncomment the jboss.profile property and change the profile name as follows:

   from

   ```
       # jboss.profile = default
   ```

to

```
jboss.profile = <your_profile>
```

A.9.7.7 Using JBoss with Non-Latin Characters

If your application server is JBoss, 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.9.7.8 JBoss 4.2 XML/A Connection Fix

JBoss 4.2 includes the JBossWS service as a standard, default feature. JasperReports Server has web services support for XML/A connections.

The web services classes in JasperReports Server and JBoss can conflict and cause the following error when attempting to utilize a JasperReports Server XML/A connection:

```
javax.xml.soap.SOAPException: Unable to create message factory for

SOAP: org.jboss.ws.core.soap.MessageFactoryImpl
```

To prevent the web services class conflict, set the special Java JVM options for JBoss 4.2, as described in **6.1.1, "Tomcat and JBoss JVM Options," on page 39**.

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. The errors do not appear to users, and they have no impact on JasperReports Server operations.

Because JasperReports Server user sessions are not persistent, you can configure your application server to disable persistence and avoid the error. For example, in Apache-Tomcat 5.5, 6, and 7 edit the file <tomcat>/conf/context.xml and locate the following lines:

```
<!-- Uncomment this to disable session persistence across Tomcat restarts -->
    <!--
    <Manager pathname="" />
    -->
```

Remove the comment markers from lines 2 and 4 above, then restart Apache-Tomcat for the change to take effect. For other application servers, refer to the product documentation.

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

You can disable this by 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 against cross-site request forgery. You can disable the protection in DWR by setting the `crossDomainSessionSecurity` parameter for the `dwr` servlet in the file <tomcat>\webapps\jasperserver\WEB-INF\web.xml:

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

A.10 Problems Importing and Exporting Data from the Repository

A.10.1 Exporting a Repository That Contains UTF-8

The following errors may happen when you have international characters in repository objects, for example, in user IDs.

A.10.1.1 Error During Export

Upgrading typically requires doing an export operation on your database. If you are using MySQL and getting this null pointer exception, it may be due to an incorrect character in the file `js.jdbc.properties`:

```
java.lang.NullPointerException
```

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

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

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

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

A.11 Property File Updates

A.11.1 Owasp.CsrfGuard.properties - Security File

There is a set of property files that help control the settings for the JasperReports Server functionality that prevents Cross-Site Request Forgery (CSRF). The key files are described in the *JasperReports Server Administrator Guide*.

One of these properties files refers to the JasperReports Server “webapp name”, but it is not currently automatically updated by the installation procedure. This file is at the following location:

```
<appserver-path>/jasperserver/WEB-INF/esapi/Owasp.CsrfGuard.properties
```

If you change the name of your webapp from the default of “jasperserver”, then you will also need to manually update the `Owasp.CsrfGuard.properties` file.

So, if you modify your default.master.properties like so:

```
# webAppNamePro = jasperserver
```

```
webAppNamePro = jasperserver-inst2
```

And then do an installation (using the `js-install-ce.sh` scripts), you will need to edit `Owasp.CsrfGuard.properties` like so:

Change:

```
org.owasp.csrfguard.NewTokenLandingPage=/jasperserver/login.html
```

```
org.owasp.csrfguard.action.Redirect.Page=/jasperserver/login.html
```

To:

```
org.owasp.csrfguard.NewTokenLandingPage=/jasperserver-inst2/login.html
org.owasp.csrfguard.action.Redirect.Page=/jasperserver-inst2/login.html
```