TIBCO JasperReports® Server

Installation Guide

Release 6.0

http://www.jaspersoft.com

This is version 1114-JSP60-38 of the *JasperReports Server Installation Guide*.

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION

TIBCO™ JasperReports® Server builds on TIBCO™ JasperReports® Library as a comprehensive family of Business Intelligence (BI) products, providing robust static and interactive reporting, report server, and data analysis capabilities. These capabilities are available as either stand-alone products, or as part of an integrated end-to-end BI suite utilizing common metadata and provide shared services, such as security, a repository, and scheduling. The server exposes comprehensive public interfaces enabling seamless integration with other applications and the capability to easily add custom functionality.

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

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

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

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

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

You can use the following sources of information to extend your knowledge of JasperReports Server:

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

- Our Online Learning Portal lets you learn at your own pace, and covers topics for developers, system administrators, business users, and data integration users. The Portal is available online from Professional Services section of our website.
- Our free samples, which are installed with JasperReports, Jaspersoft Studio, and JasperReports Server, are documented online.

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

This chapter contains the following sections:

1.1 Conventions

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

Convention	Description
<js-install>	The root directory where JasperReports Server will be installed by the binary installer.
<js-install>	For manual installations, the directory where you unpack the WAR file distribution ZIP (jasperreports-server-<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 (Linux only)	`jasperreports-server-6.0-linux-x86-installer.run`

64-bit installer	`jasperreports-server-6.0-windows-x64-installer.exe`
	`jasperreports-server-6.0-linux-x64-installer.run`
	`jasperreports-server-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.
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 22**.

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/or an application server other than Apache Tomcat, install JasperReports Server using the WAR file.

For a complete list of applications supported by the WAR file distribution, refer to the release notes 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 22**.

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.bat`
- `js-install.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.bat and js-install.sh.
JasperReports Server Database Scripts	SQL scripts for each supported database.
JasperReports Server Documentation	Guides for end users and administrators.

Content Item	Description
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 112**.

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-6.0-windows-x64-installer.exe` (64 bit only)

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

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

```
./jasperreports-server-6.0-linux-x86-installer.run    (32 bit)
./jasperreports-server-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-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

When you run the installer, you choose an “Install Type”. The first option installs all installer components and sample data resources. The second option, “Custom Install” lets you choose which components to install and whether to include sample data resources.

Install All Components and Samples Option:

This option copies 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 have the following choices: install a Bundled Tomcat or use an Existing Tomcat, install a Bundled PostgreSQL or use an Existing PostgreSQL, choose ports for Tomcat and PostgreSQL, and choose whether 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-6.0
Linux:	<USER_HOME>/jasperreports-server-6.0
Linux (as root)	/opt/jasperreports-server-6.0
Mac OSX	/Applications/jasperreports-server-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-pro/META-INF/context.xml. (And delete, if it exists: <js-install>/apache-tomcat/conf/Catalina/localhost/jasperserver-pro.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 21**.

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.

We strongly recommend 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 30**.

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 27**.

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 27**.

- **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 **5.6.1, "JasperReports Server Heartbeat," on page 41**.

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 47**.
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 Installing a New License File

By default, JasperReports Server is installed with an evaluation license that expires a number of days after the software is installed. After the license expires, you can start the server, but you cannot log in.

To obtain a commercial license, contact Technical Support or your sales representative.

To upgrade the evaluation license to a commercial one, copy the commercial license file over the evaluation license file. Application servers have work directories where JSP files are compiled and cached and other objects are stored. These directories can cause errors when updating to a new license. To avoid errors, clear the work directory before upgrading your license. For instance, if you're using Tomcat:

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

After changing to a commercial license, make sure you stop the server before replacing the license file:

1. Stop the server

2. Replace the license named `jasperserver.license` in the deployed JasperReports Server root directory with the new license file

   The file name should be `jasperserver.license`
3. Restart the server

By default, the license is in the `<js-install>` directory, but can be located elsewhere. You need to define the `-Djs.license.directory` Java Environment Variable in the Tomcat startup scripts to point to the license location. The name of the license file is `jasperserver.license`. You may have to rename the new license file to this name.

Restart JasperReports Server and log in to see if the license grants access. For information about license errors, see the troubleshooting section **A.10, "License-related Errors," on page 128**.

For additional license configuration options, refer to **6.2, "Setting Up the JasperReports Server License," on page 52**.

2.10.4 License File for Existing Tomcat as Windows Service

If you are installing JasperReports Server into an existing Tomcat installation on a Windows system that is running as a Windows Service *and* the license file is not in the default location because you did not choose the default `<js-install>` installation directory (**2.4, "Choosing Installation Type," on page 18**), you will probably have to manually configure Tomcat to locate the license file.

Follow these steps to examine and update the license location:

1. Open the Tomcat configuration tool by right-clicking the Tomcat icon in your quick-launch bar (usually in the lower-right corner of your desktop) or from the Windows 7 menu, expand **Start > All Programs > Apache Tomcat**. Right-click **Configure Tomcat** and select **Run as administrator**.
2. Select the **Java** tab.
3. At the bottom of the **Java Options** field, enter the following option:

   `-Djs.license.directory=<js-install>`

   For example:

   `-Djs.license.directory=C:\Jaspersoft\jasperreports-server-6.0`
4. Stop and restart the application server.

You should now be able to run JasperReports Server.

2.10.5 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) Linux	-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:

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-<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-<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-<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-pro
   - <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
`superuser`	`superuser`	System-wide administrator
`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 and superuser passwords (jasperadmin and superuser) 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-<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-pro/WEB-INF/logs/jasperserver.log

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

GlassFish: <glassfish>/domains/domain1/autodeploy/jasperserver-pro.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-pro/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:

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-<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 Jaspersoft technical support (http://support.jaspersoft.com) or contact your sales representative. The WAR file distribution comes in a file named jasperreports-server-6.0-bin.zip in compressed ZIP format.

This chapter contains the following sections:

5.1 Applications Supported by the WAR File Distribution

5.1.1 Database and Application Server Support

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

Database	Application Server	Instructions Located In
PostgreSQL MySQL DB2 Oracle SQL Server	Apache Tomcat JBoss GlassFish	This chapter.
	WebSphere	**Chapter 7, "Installing the WAR File for WebSphere," on page 75**
	WebLogic	**Chapter 8, "Installing the WAR File for WebLogic," on page 95**

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

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.bat
Linux	Bash shell	Bash shell	js-install.sh
Solaris	Bash shell	Korn shell (ksh)	js-install.sh
IBM AIX	Bash shell	Korn shell (ksh)	js-install.sh
HP UX	Bash shell	Posix shell (posix/sh)	js-install.sh
FreeBSD	Bash shell	C shell (tcsh)	js-install.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 (additional steps may be required for JBoss 7. Please see **5.4, "Additional Steps for Using JBoss AS 7 ," on page 40**)
   - Glassfish 2.1 or 3.0 using the default domain (domain1)
     If you use GlassFish 3.1.0 or a custom domain, see **A.9.7, "GlassFish Modifications," on page 122**.
4. Locate or install the PostgreSQL, MySQL, Oracle, SQL Server, or DB2 database. If you use DB2, follow the steps in **5.3, "Additional Steps for Using DB2 and js-install Scripts," on page 39**.

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.bat test` or a similar command. For more information about how to perform a validation test in your environment, see **5.7.3.1, "js-install Script Test Mode," on page 43**.

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-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-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`

Oracle	`appServerType=tomcat6 [tomcat7, tomcat5, jboss, jboss-eap-6, jboss-as-7, glassfish2, glassfish3, skipAppServerCheck]`
	`appServerDir=c:\\Program Files\\Apache Software Foundation\\Tomcat 7`
	`dbUsername=jasperserver`
	`dbPassword=password`
	`sysUsername=system`
	`sysPassword=password`
	`dbHost=hostname`

Database	Sample Property Values

DB2	`appServerType=tomcat6 [tomcat7, tomcat5, jboss, jboss-eap-6, jboss-as-7, glassfish2, glassfish3, skipAppServerCheck`
	`appServerDir=c:\\Program Files\\Apache Software Foundation\\Tomcat 7`
	`dbUsername=db2admin`
	`dbPassword=password`
	`dbHost=localhost`
	If you use DB2, follow the steps in **5.3, "Additional Steps for Using DB2 and js-install Scripts," on page 39**
	For DB2 8.x, change your deployed JDBC driver as described in **6.5, "Locating and Changing Buildomatic Configuration Files," on page 65**.

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

The dbUsername must be the same as the Oracle user name. In addition, buildomatic will not work with the "sys as sysdba" syntax.

If, under Linux, Tomcat is installed using apt-get, yum, or rpm, see **A.9.5, "Tomcat 6 Installed Using apt-get," on page 121**.

5. Password encryption

   The `default_master.properties` file 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 about 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.bat` (Windows) `./js-install.sh` (Linux and Mac OSX)	Installs JasperReports Server, sample data, and sample databases (foodmart and sugarcrm)
`js-install.bat minimal` (Windows) `./js-install.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 **5.7.3, "Error Running js-install Scripts (js-install.bat/sh)," on page 42**.

7. Set Java JVM Options, as described in **6.1, "Setting JVM Options for Application Servers," on page 47**. This step is required.
8. Set up the JasperReports Server License as described in **6.2, "Setting Up the JasperReports Server License," on page 52**. This step is required.

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

5.3 Additional Steps for Using DB2 and js-install Scripts

The buildomatic scripts cannot automatically connect to a remote DB2 database and carry out Admin operations, so you have to perform additional steps to create the databases.

The DB2 client software, db2 or db2cmd, can be used to interact with DB2.

1. Enter commands similar to the ones below in the DB2 command window to create and initialize the repository database, called `jsprsrvr` in DB2 to conform to the 8-character limitation:

```
db2 create database jsprsrvr using codeset utf-8 territory us pagesize 16384
```

2. (Optional) Run the following commands in the DB2 command window if you want to install sample databases:

```
db2 create database sugarcrm
db2 create database foodmart
```

3. Now, continue installing JasperReports Server as described in **5.2, "Installing the WAR File Using js-install Scripts," on page 36**.

Further considerations:

- If you are using DB2 8.1, set the `LOGFIL_SIZ` parameter to a minimum of 3000 to avoid possible log file errors while loading the `foodmart` database. Configure your `foodmart` database right after creating it by

using Control Center.

- If JasperReports Server is deployed on the same host as DB2, delete the following file to avoid conflicts: `<db2>/SQLLIB/java/db2jcc.jar`

5.4 Additional Steps for Using JBoss AS 7

If you are using JBoss AS 7 as your application server and Oracle, SQL Server, or DB2 as your database, there is an additional set of required steps to handle the JDBC driver. If you are using a different driver from the one supplied with JasperReports Server, you should have already downloaded a JDBC driver jar for your database type. (See **6.4, "Working With JDBC Drivers," on page 60**, if you have not yet done this.)

You will need to make an explicit reference to your JDBC driver file name so that JBoss 7 will know the exact file name.

1. First update your `default_master.properties` file to specify the exact name (`artifactId` and `version`) of your JDBC driver:

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

   Look for the section "Setup JDBC Driver"

   Uncomment and edit the two lines shown below:

   ```
   # maven.jdbc.artifactId=ojdbc5
   # maven.jdbc.version=11.2.0
   ```

   So that they look like this:

   ```
   maven.jdbc.artifactId=ojdbc5
   maven.jdbc.version=11.2.0
   ```

   (This will work for a driver with the filename: `ojdbc5-11.2.0.jar`)

2. Edit your `jboss-deployment-structure.xml` file so that the JDBC filename is specified:

   Edit: `<js-install>/buildomatic/install_resources/jboss/jboss-deployment-structure.xml`

   Look for the section "Setup JDBC Driver"

   Uncomment and edit the line for your database type (for instance):

   ```
   <!-- <resource-root path="WEB-INF/lib/ojdbc5-11.2.0.jar" use-physical-code-
   source="true"/> -->
   ```

   So that it looks like this:

   ```
   <resource-root path="WEB-INF/lib/ojdbc5-11.2.0.jar" use-physical-code-
   source="true"/>
   ```

   (This will work for a driver with the filename: `ojdbc5-11.2.0.jar`)

Note: If your JDBC driver filename does not have a version number, it maybe required to rename the file so that it has a version number.

For instance, if you have a file named: `sqljdbc4.jar`

You can rename it with a "dummy" version number: `sqljdbc4-1.0.jar`

Then the `artifactId` and `version` can look like this:

```
maven.jdbc.artifactId=sqljdbc4
maven.jdbc.version=1.0
```

5.5 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 31**.

5.6 Logging into JasperReports Server

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

http://<hostname>:8080/jasperserver-pro

Example:

http://localhost:8080/jasperserver-pro

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

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
`superuser`	`superuser`	System-wide administrator
`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 and superuser passwords (jasperadmin and superuser) 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.

5.6.1 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-pro/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*.

5.7 Troubleshooting Your JasperReports Server Configuration

This section describes the most common installation problems.

5.7.1 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 109**.

5.7.2 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 109**.

5.7.3 Error Running js-install Scripts (js-install.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.

5.7.3.1 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.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.sh test
   ```

5.7.4 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.

5.8 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.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 36**.

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.
`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-pro` `js-ant import-minimal-pro`	Initializes the `jasperserver` database, loads core application data. Running js-ant import-minimal-pro is mandatory. The server cannot function without this data.
`js-ant import-sample-data-pro`	(Optional) Loads the demos that use the sample data.
`js-ant deploy-webapp-pro`	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 112**.

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, MySQL, Oracle, SQL Server, or DB2, see **6.3, "Manually Creating the JasperReports Server Database," on page 54**.

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 5-3**:

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 47**. This step is required.
6. Set up the JasperReports Server License as described in **6.2, "Setting Up the JasperReports Server License," on page 52**. This step is required.

CHAPTER 6 JVM OPTIONS, LICENSE SETUP, MANUAL DATABASE CREATION

This chapter contains the following sections:

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.

For the Oracle database, setting the Oracle localization option, `defaultNChar`, can substantially impact the performance of JDBC queries. If you do not need to support UTF-8 for your Oracle database, you can omit this setting.

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 25**.

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
For Oracle	set JAVA_OPTS=%JAVA_OPTS% -Doracle.jdbc.defaultNChar=true
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"
For Oracle	export JAVA_OPTS="$JAVA_OPTS -Doracle.jdbc.defaultNChar=true"
Additional options for Java 1.6-1.7 and JBoss	export JAVA_OPTS="$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`

File	Add JVM Options After This Line on Windows
<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="$BASEDIR"/common/endorsed`
<tomcat>/bin/setenv.sh	`JAVA_OPTS` setting can go anywhere in this file.
<jboss>/bin/run.sh (JBoss 5.1)	`export JAVA_OPTS="$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;-
Dcatalina.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 25**.

```
-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 27**.

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:
       -Doracle.jdbc.defaultNChar=true
```

   If you are not using an Oracle database, you can omit the last option in example above.

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
        -Doracle.jdbc.defaultNChar=true
</jvm-options>
```

   If you are not using an Oracle database, you can omit the last option in example above.

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 Setting Up the JasperReports Server License

JasperReports Server requires a license to run and comes with an evaluation license that is valid for 30 days. Please contact Technical Support or your sales representative to get your commercial license.

The license file is in the following location:

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

The license file specifies the terms of your license, such as the following:

- Expiration date, number of users, and/or number of CPUs
- Also, features licensed separately from the basic commercial license, such as multi-tenancy

Jaspersoft receives information about your system periodically. The information is used only to monitor compliance with your license. No personal information is collected or transmitted.

6.2.1 Default License Configuration for All Application Servers

At startup time, JasperReports Server automatically looks for the jasperserver.license file in the home directory of the system user who is running the application server. **Table 6-1** lists the application server user home directories for supported operating systems.

To configure the license:

1. Stop the application server.
2. Copy the `jasperserver.license` file in `<js-install>` to the directory for your operating system:

Table 6-1 License Locations

Operating System	
Linux	/home/<user>/
Mac OSX	/Users/<user>/
Windows 7 installed from WAR file	C:\Users\<user>\
Windows 7 installed from the binary installer	C:\Users\
Windows 7 using an existing Tomcat Windows service	C:\
Windows 2003	C:\Documents and Settings\<user>\
Windows 2008	C:\Documents and Settings\<user>\

6.2.2 User-Defined License Location

If you prefer to put your license in another directory, modify your application server startup script to set a JAVA_OPT value to explicitly point to that directory.

6.2.2.1 Alternate License Setup for Tomcat

If your license is not located in the home directory of the application server user, you can set a `JAVA_OPT` value to explicitly point to your license.

On Windows:

1. In the file `<tomcat>/bin/setclasspath.bat`, locate the following line:

   ```
   set JAVA_ENDORSED_DIRS=%BASEDIR%\common\endorsed
   ```

   Alternatively, create an empty file called `<tomcat>/bin/setenv.bat`.
2. Below that line or in the new file, insert the following line:

   ```
   set JAVA_OPTS=%JAVA_OPTS% -Djs.license.directory="<js-install>"
   ```

   For example:

   ```
   set JAVA_OPTS=%JAVA_OPTS% -Djs.license.directory="C:\jasperserver-bin"
   ```

On Linux and Mac OSX:

1. In the file `<tomcat>/bin/setclasspath.sh`, locate the following line:

   ```
   JAVA_ENDORSED_DIRS="$BASEDIR"/common/endorsed
   ```

   Alternatively, create an empty file called <tomcat>/bin/setenv.sh.
2. Below that line or in the new file, insert the following line:

   ```
   export JAVA_OPTS="$JAVA_OPTS -Djs.license.directory=<js-install>"
   ```

   For example:

   ```
   export JAVA_OPTS="$JAVA_OPTS -Djs.license.directory=/home/user/jasperserver-bin"
   ```

6.2.2.2 Alternate License Setup 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 specify a specific folder to hold the `jasperserver.license` are the following:

1. Open this file for editing:

   ```
   cd <js-install>/apache-tomcat/bin/service.bat
   ```
2. Look for the following line (second line of two that set JVM options):

```
"%EXECUTABLE%" //US//%SERVICE_NAME% ++JvmOptions
"-Djs.license.directory=C:\Jaspersoft\jasperreports-server-5.1;
-XX:PermSize=32m;-XX:MaxPermSize=512m;-XX:+UseConcMarkSweepGC;
-XX:+CMSClassUnloadingEnabled" "-Djava.io.tmpdir=%CATALINA_BASE%\temp;
-Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager;
-Djava.util.logging.config.file=%CATALINA_BASE%\conf\logging.properties"
```

Update the line above to point to this example license location:

```
-Djs.license.directory=C:\MyLicenses
```

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 these commands, the service is running.

6.2.2.3 Alternate License Location for Existing Tomcat as a Windows Service

Windows 7:

If you have an existing Tomcat as a Windows Service under Windows 7, copy your license to the root of the C: drive. This is the home folder for the SYSTEM user. The location is:

```
C:\jasperserver.license
```

6.2.2.4 Alternate License Setup for JBoss

If your license will not be located in the home directory of the application server user, you can set a `JAVA_OPT` value to explicitly point to your license.

On Windows:

1. In the file `<jboss>/bin/run.bat`, locate the following line:

   ```
   set JAVA_OPTS=%JAVA_OPTS% -Dprogram.name=%PROGNAME%
   ```

2. Below that line, insert the following line:

   ```
   set JAVA_OPTS=%JAVA_OPTS% -Djs.license.directory="<js-install>"
   ```

   For example:

   ```
   set JAVA_OPTS=%JAVA_OPTS% -Djs.license.directory="C:\jasperserver-bin"
   ```

On Linux and Mac OSX:

1. In the file `<jboss>/bin/run.sh`, locate the following line:

   ```
   export JAVA_OPTS="$JAVA_OPTS -Dprogram.name=$PROGNAME"
   ```

2. Below that line, insert the following line:

   ```
   export JAVA_OPTS="$JAVA_OPTS -Djs.license.directory=<js-install>"
   ```

   For example:

   ```
   export JAVA_OPTS="$JAVA_OPTS -Djs.license.directory=/home/user/jasperserver-bin"
   ```

6.3 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:

- **6.3.1, "PostgreSQL," on page 55**
- **6.3.2, "MySQL," on page 55**
- **6.3.3, "Oracle," on page 56**
- **6.3.4, "DB2," on page 58**
- **6.3.5, "SQL Server," on page 59**

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

6.3.1 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-pro-create.ddl
postgres=#\i quartz.ddl
postgres=#\q
```

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

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

3. If you did not install the optional sample databases, complete the installation with these commands:

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

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

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

   For more information about executing the Ant scripts, see **5.8, "Installing the WAR File Manually," on page 43**.
4. Set Java JVM Options, as described in **6.1, "Setting JVM Options for Application Servers," on page 47**. This step is required.
5. Set up the JasperReports Server License as described in **6.2, "Setting Up the JasperReports Server License," on page 52**. This step is required.

6.3.2 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-pro-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-pro
js-ant deploy-webapp-pro
```

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

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

   For more information about executing the Ant scripts, see **5.8, "Installing the WAR File Manually," on page 43**.
4. Set Java JVM Options, as described in **6.1, "Setting JVM Options for Application Servers," on page 47**. This step is required.
5. Set up the JasperReports Server License as described in **6.2, "Setting Up the JasperReports Server License," on page 52**. This step is required.

6.3.3 Oracle

To manually create the JasperReports Server database in Oracle:

The Oracle client software, `sqlplus.exe` or `sqlplus`, can be used to interact with Oracle.

For specific details on connecting to the Oracle 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/oracle
sqlplus /nolog (start sqlplus client)
SQL> connect system/password (use your sysUsername and password)
(or SQL>connect sys/password as sysdba
SQL> create user jasperserver identified by password;
SQL> grant connect, resource to jasperserver;
SQL> connect jasperserver/password@ORCL (use your password, your SID)
SQL> @js-pro-create.ddl
SQL> @quartz.ddl
SQL> exit
```

2. (Optional) Special edit to the `sugarcrm.sql` script which creates the `sugarcrm` sample database. The `sqlplus` command line tool interprets SQL statements differently than a JDBC call (that is, the way buildomatic executes SQL scripts). Because of this, the `sugarcrm.sql` file must be edited in order to execute using `sqlplus`. To make these edits do the following:
   - Unzip the `sugarcrm.zip` file to get the `sugarcrm.sql` file. Open `sugarcrm.sql` for editing:
   - Uncomment the "-- set define off" line to look like this "set define off" (Line 7)
   - Uncomment the "--/" line that follows the CREATE TRIGGER statements (there are 12 of these toward the very end of the file on line 71,282. Just before the CREATE INDEX statements). Change to be just "/". (This terminates the trigger procedure definition in `sqlplus`.)
   - Save the file.

If you build and load the sample databases using buildomatic, the NLS_LANG setting is automatically handled via a JDBC driver setting.

If you load the sample databases using buildomatic, you will not need to set any variables or make any script edits.

3. (Optional) Set the `NLS_LANG` variable. The `sugarcrm` database has test data that requires a specific NLS_LANG setting in order to load into Oracle correctly. You will need to set this in your shell environment if you are manually loading the `sugarcrm` database.

Windows: `set NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P1`

Linux: `export NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P1`

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

```
cd <js-install>/buildomatic/install_resources/sql/oracle
sqlplus /nolog (start sqlplus client)
SQL> connect system/password (use your sysUsername and password)
(or SQL>connect sys/password as sysdba
SQL> create user sugarcrm identified by password;
SQL> create user foodmart identified by password;
SQL> grant connect, resource to sugarcrm;
SQL> grant connect, resource to foodmart;
SQL> connect sugarcrm/password@ORCL
SQL> @sugarcrm.sql (First, make sure file is unzipped)
```

```
SQL> connect foodmart/password@ORCL
SQL> @foodmart-oracle.sql (First, make sure file is unzipped)
SQL> @supermart-update.sql
SQL> exit
```

5. If you did not install the optional sample databases, complete the installation with these commands:

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

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

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

   For more information about executing the Ant scripts, see **5.8, "Installing the WAR File Manually," on page 43**.
6. Set Java JVM Options, as described in **6.1, "Setting JVM Options for Application Servers," on page 47**. This step is required.
7. Set up the JasperReports Server License as described in **6.2, "Setting Up the JasperReports Server License," on page 52**. This step is required.

6.3.4 DB2

To manually create the JasperReports Server database in DB2:

The DB2 client software, `db2` or `db2cmd`, can be used to interact with DB2.

For specific details on connecting to the DB2 database and setting privileges for databases and db users, please refer to the documentation provided with your database.

1. Change to the following directory:

   ```
   cd <js-install>/buildomatic/install_resources/sql/db2
   ```
2. Enter these commands in the DB2 command window to create and initialize the repository database, called `jsprsrvr` in DB2 to conform to the 8-character limitation:

   ```
   db2 create database jsprsrvr using codeset utf-8 territory us pagesize 16384
   db2 -tf js-pro-create.ddl
   db2 -tf quartz.ddl
   ```
3. (Optional) Run the following commands in the DB2 command window if you want to install sample databases:

   ```
   db2 create database sugarcrm
   db2 -tf sugarcrm.sql (first make sure file is unzipped)
   db2 create database foodmart
   db2 -tf foodmart-db2.sql (first make sure file is unzipped)
   db2 -tf supermart-update.sql (if script is available)
   ```

4. If you did not install the optional sample databases, complete the installation with these commands:

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

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

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

   For more information about executing the Ant scripts, see **5.8, "Installing the WAR File Manually," on page 43**.
5. Set Java JVM Options, as described in **6.1, "Setting JVM Options for Application Servers," on page 47**. This step is required.
6. Set up the JasperReports Server License as described in **6.2, "Setting Up the JasperReports Server License," on page 52**. This step is required.

Further considerations:

- If you are using DB2 8.1, set the `LOGFIL_SIZ` parameter to at least 3000 to avoid possible log file errors while loading the foodmart database. Configure your foodmart database right after creating it by using Control Center.
- If JasperReports Server is deployed on the same host as DB2, delete the following file to avoid conflicts: `<db2>/SQLLIB/java/db2jcc.jar`

6.3.5 SQL Server

Using the `sqlcmd` utility to manually build the `jasperserver` database.

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

To manually create the JasperReports Server database in SQL Server:

1. Open a Command Prompt and enter the following commands using the administrator (sa) user name and password.

```
cd <js-install>\buildomatic\install_resources\sql\sqlserver
sqlcmd -S ServerName -Usa -Psa
1> CREATE DATABASE [jasperserver]
2> GO
1> USE [jasperserver]
2> GO
1> :r js-pro-create.ddl
2> GO
1> :r quartz.ddl
2> GO
```

2. From the Windows Start Menu, select **Microsoft SQL Server > SQL Server Management Studio**.
3. Connect to SQL Server as the administrative database user, and check that the `jasperserver` database appears in the Object Explorer.
4. Expand Tables in the `jasperserver` database, and check that the tables have been added.

To manually create the optional sample databases in SQL Server:

5. Extract the files in the `sugarcrm.zip` file to the level above your current directory, placing the `sugarcrm.sql` file in this directory:
   `<js-install>\jasperserver\buildomatic\install_resources\sql\sqlserver`
6. Enter these commands to create and initialize the `sugarcrm` database:

```
1> CREATE DATABASE [sugarcrm]
2> GO
1> USE [sugarcrm]
2> GO
1> :r sugarcrm.sql
2> GO
```

7. Extract the files in the `FoodMartCreateData.zip` into your current directory.
8. Enter these commands to create and initialize the foodmart database:

```
1> CREATE DATABASE [foodmart]
2> GO
1> USE [foodmart]
2> GO
1> :r FoodMartCreateData.sql
2> GO
```

To complete the manual installation of databases in SQL Server:

9. If you did not install the optional sample databases, complete the installation with these commands:

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

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

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

   For more information about executing the Ant scripts, see **5.8, "Installing the WAR File Manually," on page 43**.
10. Set Java JVM Options, as described in **6.1, "Setting JVM Options for Application Servers," on page 47**. This step is required.
11. Set up the JasperReports Server License as described in **6.2, "Setting Up the JasperReports Server License," on page 52**. This step is required.

6.4 Working With JDBC Drivers

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

6.4.1 Open Source JDBC Drivers

For open source JDBC drivers, buildomatic is set up 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 set up 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.

6.4.1.1 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-pro, the jdbc3 driver will be copied to your application server.

6.4.1.2 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
```

6.4.2 Commercial JDBC Drivers

As of version 5.6.1, JasperReports Server includes the TIBCO JDBC drivers for the following commercial databases. You can connect to these databases using the TIBCO JDBC driver without additional steps. The driver name is included for reference:

- Oracle — ji-oracle-driver-1.0.2.jar
- SQL Server — ji-sqlserver-driver-1.0.2.jar,
- DB2 — ji-db2-driver-1.0.2.jar

These drivers require a valid JasperReports Server license. The driver is for use by JasperReports Server only and the driver jar must be located under the jasperserver-pro directory, for example, <tomcat_home>/tomcat/jasperserverpro/web-inf/lib.

If you are using the default settings for the driver, you do not need to edit default_master.properties.

You can also choose to use the driver supplied by the database vendor as described below. To do this, you must first obtain and install the driver you want, then modify your default_master.properties to use your driver.

6.4.2.1 Download an Optional JDBC Driver Jar

To use the driver supplied by the database vendor, you can optionally download and install it.

Download Driver Jar from Vendor Website

To download a commercial JDBC driver for a particular vendor, you will need to go to the download website of that vendor. It is typical that these vendors will require you to register and login to their website before you can download driver packages. Additionally, it is standard for the vendor's website to present a terms of service agreement that will need to be agreed to. Here are some common URL addresses where you can find download packages for supported databases:

- http://www.microsoft.com/en-us/download/details.aspx?id=11774 (SQL Server)
- http://www.oracle.com/technetwork/indexes/downloads (Oracle)
- http://www-01.ibm.com/software/data/db2/linux-unix-windows/downloads.html (DB2)

Collect Driver Jar from Existing Application

It is very often the case that a JDBC driver is already present in an application that is running within your network. If this is the case, then it can be very convenient to collect a copy of that driver jar and copy it to the JasperReports Server install location.

Once you have downloaded your driver, copy it to the correct location and configure your files as described in the sections below.

6.4.2.2 Oracle Example

Once you have downloaded an Oracle driver or found an existing Oracle driver on your system, copy it to the correct location and modify the installation files as follows:

1. Copy the driver you want to use to the following directory:

   `<js-install>/buildomatic/conf_source/db/oracle/jdbc`
2. Change to the <js_install>/buildomatic directory and open default_master.properties in a text editor.
3. Go to the Additional Settings section in this file.
4. Go to the first setup item, Setup Standard Oracle JDBC Driver.
5. Follow the instructions to uncomment the required properties and enable your driver. The following example shows how you would set up default_master.properties to point to a driver named ojdbc5-11.2.0.jar using SID:

```
# 1) Setup Standard Oracle JDBC Driver
#
# Uncomment and modify the value in order to change the default
# 1a) Driver will be found here: <path>/buildomatic/conf_source/db/oracle/jdbc
#
maven.jdbc.groupId=oracle
maven.jdbc.artifactId=ojdbc5
maven.jdbc.version=11.2.0.2

jdbcDriverClass=oracle.jdbc.OracleDriver
jdbcDataSourceClass=oracle.jdbc.pool.OracleConnectionPoolDataSource
#
# 1b) Service Name
# If you are using an oracle service name instead of an SID, your JDBC URL will
# typically have a form similar to:  jdbc:oracle:thin:@host:1521/servicename
# To enable this you should uncomment the following line and add your service name
# serviceName=
```

   If you are using an Oracle service name instead of an SID, uncomment the line `serviceName=` and add your service name.
6. Save the default_master.properties file.
7. Open the <js_install>/conf_source/db/oracle/db.template.properties file in a text editor.
8. Locate the values for the TIBCO JDBC drivers and comment them out:

```
# admin.jdbcUrl=jdbc:tibcosoftware:oracle://${dbHost}:${dbPort};${dbSidOrServiceNameProp}
                                                              # ${AdditionalAdminProperties}
# js.jdbcUrl=jdbc:tibcosoftware:oracle://${dbHost}:${dbPort};${dbSidOrServiceNameProp}
# sugarcrm.jdbcUrl=jdbc:tibcosoftware:oracle://${dbHost}:${dbPort};${dbSidOrServiceNameProp}
# foodmart.jdbcUrl=jdbc:tibcosoftware:oracle://${dbHost}:${dbPort};${dbSidOrServiceNameProp}
```

9. Uncomment out the values for the vendor's Oracle driver and make sure they are as follows:

```
admin.jdbcUrl=jdbc:oracle:thin:@${dbHost}:${dbPort}${dbSidOrServiceName}
js.jdbcUrl=jdbc:oracle:thin:@${dbHost}:${dbPort}${dbSidOrServiceName}
sugarcrm.jdbcUrl=jdbc:oracle:thin:@${dbHost}:${dbPort}${dbSidOrServiceName}
foodmart.jdbcUrl=jdbc:oracle:thin:@${dbHost}:${dbPort}${dbSidOrServiceName}
```

10. Save the file.

6.4.2.3 SQL Server Example

Once you have downloaded SQL Server driver or found an existing SQL Server driver on your system, copy it to the correct location and modify the installation files as follows:

1. Copy the driver you want to use to the following directory:

   `<js-install>/buildomatic/conf_source/db/sqlserver/jdbc`
2. Change to the <js_install>/buildomatic directory and open default_master.properties in a text editor.
3. Go to the Additional Settings section in this file.
4. Go to the first setup item, Setup Standard SQL Server JDBC Driver.
5. Uncomment the required properties and enable your driver. The following example shows how you would set up default_master.properties to point to a driver named sqljdbc-1.6.jar:

```
# Default) Setup Tibco/Jaspersoft/Progress JDBC Driver
#
# uncomment the following line for JDBC 4
# jdbcDataSourceClass=tibcosoftware.jdbc.sqlserver.SQLServerDataSource40

# 1) Setup Standard SQLServer JDBC Driver
#
# Uncomment and modify the value in order to change the default
# Driver will be found here: <path>/buildomatic/conf_source/db/sqlserver/jdbc
#
maven.jdbc.groupId=sqlserver
maven.jdbc.artifactId=sqljdbc
maven.jdbc.version=1.6
# admin.jdbcUrl=jdbc:sqlserver://${dbHostOrInstance};SelectMethod=cursor
sqlserver.jdbcUrlProtocol=jdbc:sqlserver
jdbcDriverClass=com.microsoft.sqlserver.jdbc.SQLServerDriver
jdbcDataSourceClass=com.microsoft.sqlserver.jdbc.SQLServerConnectionPoolDataSource
```

6. Make sure the following line is **not** commented:

```
# admin.jdbcUrl=jdbc:sqlserver://${dbHostOrInstance};SelectMethod=cursor
```

7. Save the default_master.properties file.

6.4.2.4 DB2 Example

Once you have downloaded a DB2 driver or found an existing DB2 driver on your system, copy it to the correct location and modify the installation files as follows:

1. Copy the driver you want to use to the following directory:

   `<js-install>/buildomatic/conf_source/db/db2/jdbc`
2. Change to the <js_install>/buildomatic directory and open default_master.properties in a text editor.
3. Go to the Additional Settings section in this file.
4. Go to the first setup item, Setup Standard DB2 JDBC Driver.
5. Follow the instructions to uncomment the required properties and enable your driver.

6. Add the following additional properties, setting the correct values for your installation. For example:

```
db2.driverType=4
db2.fullyMaterializeLobData=true
db2.fullyMaterializeInputStreams=true
db2.progressiveStreaming=2
db2.progressiveLocators=2
dbPort=50000
js.dbName=JSPRSRVR
sugarcrm.dbName=SUGARCRM
foodmart.dbName=FOODMART
```

7. Save the default_master.properties file.

6.4.3 Application Server Copy-to Locations

When the `deploy-webapp-pro` 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

6.5 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.

6.5.1 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.

6.5.2 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.)

6.5.3 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-pro.war`

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

6.5.4 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-pro-create.ddl
<js-install>/buildomatic/install_resources/sql/postgresql/quartz.ddl
<js-install>/buildomatic/install_resources/sql/postgresql/upgrade-postgresql-5.6.0-6.0.0-pro.sql
<js-install>/buildomatic/install_resources/sql/postgresql/js-pro-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.

6.5.5 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
oracle/scripts.properties
(same pattern for additional databases)

6.5.6 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-pro`, for instance. These export files along with other important export files are located here:

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

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

js-catalog-postgresql-minimal-pro.zip
js-catalog-postgresql-pro.zip
js-catalog-mysql-minimal-pro.zip
js-catalog-mysql-pro.zip
js-catalog-oracle-minimal-pro.zip
js-catalog-oracle-pro.zip

6.5.7 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-pro` WAR file the hibernate.properties file is found at the following location:

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

The buildomatic scripts automatically create this configuration file. When you run the buildomatic target `deploy-webapp-pro` 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`
DB2:	`metadata.hibernate.dialect=com.jaspersoft.ji.hibernate.dialect.DB2JICustomDialect`
Oracle:	`metadata.hibernate.dialect=com.jaspersoft.ji.hibernate.dialect.OracleJICustomDialect`
SQL Server:	`metadata.hibernate.dialect=com.jaspersoft.ji.hibernate.dialect.SQLServerJICustomDialect`

6.5.8 Database Connection Configuration Files

6.5.8.1 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-pro` is run, the context.xml will be automatically copied into the jasperserver-pro 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-pro/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-pro 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-pro.xml

6.5.8.2 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-pro` 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-pro.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.9.8, "JBoss Modifications," on page 123**.

6.5.8.3 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-pro` 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 data source definition manually, you can run a command similar to the following:

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

6.6 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-pro/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-pro/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

6.6.1 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.

6.6.2 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
	DB2	org.quartz.impl.jdbcjobstore.DB2v8Delegate
	Oracle	org.quartz.impl.jdbcjobstore.StdJDBCDelegate
	SQL Server	org.quartz.impl.jdbcjobstore.StdJDBCDelegate

6.6.3 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-pro
	JBoss	http://localhost:8080/jasperserver-pro
	GlassFish	http://localhost:8080/jasperserver-pro
	WebLogic	http://localhost:7001/jasperserver-pro
	WebSphere	http://localhost:9080/jasperserver-pro

6.6.4 Settings for the Quartz Table Prefix

For databases that support schemas, such as Oracle, SQL Server, and DB2, you can set the Quartz table prefix so that it includes the schema if you use one. In the default configuration, only DB2 requires an explicit schema name.

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, for example `JSPRSRVR.QRTZ_` for DB2.

6.6.5 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/iePro/js.quartz.properties`

Change the following properties:

Property	Description
`quartz.delegateClass`	Set to the same value as described in **6.6.2, "Database Settings for the Quartz Driver Delegate Class," on page 70**.
`quartz.tablePrefix`	Set to the same value as described in **6.6.4, "Settings for the Quartz Table Prefix," on page 71**

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

6.6.6.1 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-pro
```

6.6.6.2 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.

6.7 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.

CHAPTER 7 INSTALLING THE WAR FILE FOR WEBSPHERE

JasperReports Server supports deployment on the IBM WebSphere Application Server, but requires its own database to store information such as users, organizations, and the repository. WebSphere users need the WAR file distribution to install JasperReports Server. Download the WAR file distribution from Jaspersoft technical support or contact your sales representative. The WAR file distribution comes in a file named jasperreports-server-6.0-bin.zip, the compressed ZIP format.

The WAR file distribution also includes two sample databases containing data for optional demos. For evaluation, Jaspersoft recommends you install the sample databases. In a production environment, you typically don't install the sample databases. You create and initialize the required repository database and the optional sample databases before JasperReports Server is deployed in WebSphere. Deployment is performed by the WebSphere administrator using the WebSphere Administrative Console.

This chapter contains the following sections:

7.1 Procedure for Installing and Deploying the WAR File in WebSphere

Perform the procedures in this section to install and deploy the JasperReports Server WAR file in WebSphere.

7.1.1 Installing WebSphere and a Database

Install WebSphere and a database:

1. (If you are not using WebSphere 7.0, you can skip this step.) Apply WebSphere Fix 7.0.0.11 to WebSphere 7.0 or install WebSphere 7.0.0.11.
2. Check that the WebSphere installation created a JAVA_HOME system environment variable. The variable needs to be set to the JAVA directory in the WebSphere installation.
3. Install the PostgreSQL, MySQL, Oracle, SQL Server, or DB2 database.

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

Next, prepare the JasperReports Server files.

7.1.2 Preparing JasperReports Server Files

To prepare JasperReports Server files:

1. Unpack jasperreports-server-6.0-bin.zip to a top-level directory.
   Unpacking the ZIP file creates the directory `jasperreports-server-6.0-bin`.
2. Manually create and load the JasperReports Server database. Required.
3. Manually create and load the sample databases. Optional.
   6.3, "Manually Creating the JasperReports Server Database," on page 54 contains instructions for manually creating the databases.
4. Manually import the default users and organization. Required.
   a. Copy the `<dbType>_master.properties` file for your database from `sample_conf` and paste it to `buildomatic`:
      - `cd <js-install>/buildomatic`
      - Copy from — `<js-install>/buildomatic/sample_conf`
      - Paste to — `<js-install>/buildomatic`

      For example, copy `sample_conf/postgresql_master.properties` to `buildomatic`.
   b. Rename the file you copied to `default_master.properties`.
   c. Edit the `default_master.properties` file:
      - Set `appServerType` to `skipAppServerCheck`.
      - Change `dbUsername`, `dbPassword`, and `dbHost` to appropriate settings for your database.
      - If you are using a different port from the default for your database, or if you have installed the database on a remote machine, change the `dbPort` field under Custom Properties to the appropriate settings.

      Each `sample_conf/<dbType>_master.properties` file contains appropriate sample values.
   d. Start your database server.
   e. Open a Command Prompt as Administrator and run these commands:

Table 7-1 Buildomatic Targets to Execute

Commands	Description
`cd <js-install>/buildomatic`	Go to the buildomatic directory
`js-ant create-js-db`	Create the jasperserver repository database
`js-ant init-js-db-pro` `js-ant import-minimal-pro`	Initializes database, loads core application data

Commands	Description
`js-ant create-sugarcrm-db` `js-ant create-foodmart-db`	(Optional) Creates sample databases
`js-ant load-sugarcrm-db` `js-ant load-foodmart-db`	(Optional) Loads sample data into the sample databases
`js-ant import-sample-data-pro`	(Optional) Loads the demos that use the sample data

5. Set up your license file. For information, refer to **6.2, "Setting Up the JasperReports Server License," on page 52**.

Next, configure Hibernate and Quartz information in the WAR file.

7.1.3 Configuring CsrfGuard, Hibernate and Quartz Settings

Before deploying the JasperReports Server WAR file, update the CsrfGuard, Hibernate, Quartz, and settings as described in this procedure.

Configure CrsfGuard, Hibernate and Quartz settings in the WAR file:

1. The WAR file is an archive format in a single file. We will start by extracting the `Owasp.CsrfGuard.properties` file using the command below:

```
cd <js-install>
"%JAVA_HOME%\bin\jar" xf jasperserver-pro.war WEB-INF/esapi/Owasp.CrsfGuard.properties
```

   The `jar` command creates the `WEB-INF/esapi` folder in the current location and places the extracted file there.

   Open the `WEB-INF/esapi/Owasp.CrsfGuard.properties` file for editing and make the following change:

   From:

   `org.owasp.csrfguard.PRNG=SHA1PRNG`

   To (change the first line and add the second line):

   `org.owasp.csrfguard.PRNG=IBMSecureRandom`

   `org.owasp.csrfguard.PRNG.Provider=IBMJCE`

2. Copy the already configured files for `hibernate.properties` and `js.quartz.properties` to the `WEB-INF` folder.

   (These files have been pre-configured by buildomatic for your database type in the steps above.)

   From:

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

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

   To:

   <js-install>/WEB-INF

3. Edit the scheduler URI port value for WebSphere in the `js.quartz.properties`:

   Edit `js.quartz.properties`:

   Set :

   ```
   report.scheduler.web.deployment.uri=http://localhost:8080/jasperserver-pro
   ```

   To:

   ```
   report.scheduler.web.deployment.uri=http://localhost:9080/jasperserver-pro
   ```

4. If you want to configure JasperReports Server to automatically schedule and email reports, enter your mail server information in the `js.quartz.properties` file. Modify all `report.scheduler.mail.sender.*` properties as necessary for your mail server.
5. Now that you have modified/updated the individual configuration files, you must replace them into the WAR file archive using the following commands.

```
cd <js-install>
"%JAVA_HOME%\bin\jar" uf jasperserver-pro.war WEB-INF/hibernate.properties
"%JAVA_HOME%\bin\jar" uf jasperserver-pro.war WEB-INF/js.quartz.properties
"%JAVA_HOME%\bin\jar" uf jasperserver-pro.war WEB-INF/esapi/Owasp.CsrfGuard.properties
```

6. (WebSphere 8 only) If you are using WebSphere 8, enter the following additional commands:

```
cd <js-install>
zip -d jasperserver-pro.war WEB-INF/lib/stax-api-1.0.1.jar
zip -d jasperserver-pro.war WEB-INF/lib/jta-1.1.jar
```

7. Delete the `WEB-INF` directory that you created, along with the edited files it contains.

Next, configure a JDBC Provider in WebSphere.

7.1.4 Configuring a JDBC Provider in WebSphere

To configure a JDBC Provider in WebSphere:

1. Launch the WebSphere Administrative Console and navigate to **Resources > JDBC > JDBC Providers**.
2. On the JDBC providers page, click the **Guided Activity** link at the top of the **JDBC Providers** page and follow Integrated Solutions Console instructions:
   a. Configure credentials for a secure database.

      Use the J2C authentication aliases panel to create a new authenticated user.

      In Global Security, click **New** and enter the following user alias, user ID, and password. These are the credentials that WebSphere uses to access the database:

Table 7-2 J2C Authentication Alias Settings

	Alias	Example User ID	Example Password
PostgreSQL	postgresql_jasperdb	postgres	postgres
MySQL	mysql_jasperdb	root	password

	Alias	Example User ID	Example Password
Oracle	jasperserver_user	jasperserver	password
DB2	db2admin_user	db2admin	password
SQL Server	jasperserver_user	sa	sa

a. Connect to a database panel. From the Scope drop-down, choose **Node:<node_name>,Server=<server_name>**

b. Click the **New** button to create a new JDBC Provider.

c. Select your database type as follows:

- If you are using PostgreSQL, MySQL, or a TIBCO JDBC driver — select User-defined.
- If you are using a vendor-supplied driver for DB2, Oracle, or SQL Server — select your database.

Different options appear depending on your database type.

d. Select or enter these options, depending on your database type:

Database type	Implementation class name or type	Name
User-defined (PostgreSQL)	org.postgresql.jdbc2.optional.ConnectionPool	PostgreSQL JDBC Provider
User-defined (MySQL)	com.mysql.jdbc.jdbc2.optional. MysqlConnectionPoolDataSource	MySQL JDBC Provider
User-defined (TIBCO JDBC Oracle)	tibcosoftware.jdbc.oracle.OracleDataSource	TIBCO JDBC Provider - Oracle
User-defined (TIBCO JDBC SQL Server)	tibcosoftware.jdbc.sqlserver.SQLServerDataSource	TIBCO JDBC Provider - SQL Server
User-defined (TIBCO JDBC DB2)	tibcosoftware.jdbc.db2.DB2DataSource	TIBCO JDBC Provider - DB2
DB2	Connection pool data source	DB2 Universal JDBC Driver Provider
Oracle	Connection pool data source	Oracle JDBC Driver
SQL Server	Connection pool data source	Microsoft SQL Server JDBC Driver

3. Click **Next** and enter the database classpath information for the JDBC provider.

a. For TIBCO JDBC drivers for Oracle, SQL Server, or DB2, locate the correct JAR in <js-install>\buildomatic\conf_source\db\<your_database>\jdbc:

- Oracle — <js-install>\buildomatic\conf_source\db\oracle\jdbc\ji-oracle-driver-1.0.2.jar
- SQL Server — <js-install>\buildomatic\conf_source\db\sqlserver\jdbc\ji-sqlserver-driver-1.0.2.jar
- DB2 — <js-install>\buildomatic\conf_source\db\db2\jdbc\ji-db2-driver-1.0.2.jar

Copy the JAR file to a location in your WebSphere deployment and specify that location for the JDBC driver path.

If JasperReports Server is deployed on the same host as DB2, delete the following file to avoid conflicts: <db2>/SQLLIB/java/db2jcc.jar

b. For PostgreSQL, MySQL, and vendor drivers for Oracle, SQL Server, and DB2, enter the following:

<js-install>\buildomatic\conf_source\db\<your_database>\jdbc\

For example, enter:

C:\jasperreports-server-6.0-bin\buildomatic\conf_source\db\postgresql\jdbc\postgresql-9.2-1002.jdbc4.jar

Alternatively, you can copy the jar to a location in your WebSphere deployment and specify that location for the JDBC driver path.

If JasperReports Server is deployed on the same host as DB2, delete the following file to avoid conflicts: <db2>/SQLLIB/java/db2jcc.jar

4. Click Next to proceed to the next step.
5. Review the JDBC provider information you have entered and click Finish.
6. Configure data sources and custom properties using one of these procedures:
   - **7.1.4.1, "Defining a JNDI Name and Sample Data Sources for PostgreSQL and MySQL," on page 80**
   - **7.1.4.2, "Defining a JNDI Name and Sample Data Sources for DB2," on page 82**
   - **7.1.4.3, "Defining a JNDI Name and Sample Data Sources for Oracle," on page 84**
   - **7.1.4.4, "Defining a JNDI Name and Sample Data Sources for SQL Server," on page 87**

7.1.4.1 Defining a JNDI Name and Sample Data Sources for PostgreSQL and MySQL

To define the JDBC data source and expose it through JNDI:

1. Click the name of the JDBC provider that you just created. For example, **PostgreSQL JDBC Provider**.
2. Click **Data sources** under the Additional Properties of the JDBC provider details panel.
3. To create a new data source, click **New** and the new data source wizard appears.
4. Enter the data source name: `jasperserver`
5. Enter the JNDI name: `jdbc/jasperserver`
6. Click **Next**, choose **Select an existing JDBC provider**, then select **PostgreSQL JDBC Provider** or **MySQL JDBC Provider** from the drop-down list.
7. Click **Next** and accept the default helper class (com.ibm.websphere.rsadapter.GenericDataStoreHelper). Select the check box to use this data source in container managed persistence (CMP).
8. Click **Next** and select the Setup security aliases:

Field Name	PostgreSQL Value	MySQL Value
Component-managed authentication alias	<node>/postgresql_jasperdb	<node>/mysql_jasperdb
Mapping configuration alias	DefaultPrincipalMapping	DefaultPrincipalMapping

Field Name	PostgreSQL Value	MySQL Value
Container-managed authentication alias	<node>/postgresql_jasperdb	<node>/mysql_jasperdb

9. Click **Next**, review the summary information, and click **Finish**.

Next, set the connection pool size.

To set the connection pool size:

1. In the list of JDBC data sources, click on the newly created **jasperserver** data source to edit it.
2. Click **Additional Properties > Connection Pool Properties**.
   You see that **Maximum Connections** is set to 10 by default.
3. Set **Maximum Connections** to 50. You may want to set it to a higher value if necessary.
4. Click **Save**.

Next, define custom properties.

To define custom properties:

1. In the list of JDBC data sources, select the check box for the newly created **jasperserver** data source, and click **Test Connection**.
   In the Messages area a success or failure message appears. The failure message gives you information about which custom properties you need to define. For example, in the case of PostgreSQL 9.0 and MySQL, a message indicates that the error is related to a null database name.
2. Navigate to the jasperserver data sources General Properties page.
3. In Additional Properties on the right side of the General Properties page of the jasperserver data source, click **Custom properties**.
4. Scroll down the list of properties and select **databaseName**. Set the value to `jasperserver`.
5. If you're using MySQL, create a new property called **url**. Enter the following value and save the change:
   `jdbc:mysql://localhost/jasperserver?useUnicode=true&characterEncoding=UTF-8`
6. Click **Save directly to the master configuration**.

Next, if you installed sample databases, sugarcrm and foodmart, create data sources to use them.

To create optional sugarcrm and foodmart data sources:

1. If you plan to run the sample reports, use the following values to create the foodmart and sugarcrm JNDI data sources:

Field Name	Value	
Data source name	foodmart	sugarcrm
JNDI name	jdbc/foodmart	jdbc/sugarcrm

2. Click **Save directly to the master configuration**.
3. Set the connection pool size as described in **"To set the connection pool size" on page 81**.

Next, deploy the WAR file in WebSphere as described in **7.1.5, "Deploying the JasperReports Server WAR File in WebSphere," on page 88**.

7.1.4.2 Defining a JNDI Name and Sample Data Sources for DB2

To define the JDBC data source and expose it through JNDI:

1. Click the name of the JDBC provider that you just created. For example, **TIBCO JDBC Provider - DB2** or **DB2 Universal JDBC Provider**.
2. Click Data sources under the Additional Properties of the JDBC provider details panel.
3. To create a new data source, click New and the new data source wizard appears.
4. Enter the data source name: `jasperserver`
5. Enter the JNDI name: `jdbc/jasperserver`
6. Click **Next.**
7. (If you are using the TIBCO JDBC Provider, skip to **step 9**.) If you are using the vendor's DB2 driver, choose **Select an existing JDBC provider**, then select **DB2 Universal JDBC Provider** from the drop-down list.
8. Click **Next** and enter these values:

Field Name	Value
Driver type	4
Database name	jsprsrvr
Server name	localhost
Port number	50000

9. Select **Use this data source in CMP** and click **Next**.
10. In the Setup security aliases page, enter the following value for Component-managed authentication alias:
    <node>/db2admin_user
11. Click **Next**, review the summary information, and click **Finish**.

Next, set the connection pool size.

To set the connection pool size:

1. In the list of JDBC data sources, click on the newly created **jasperserver** data source to edit it.
2. Click **Additional Properties > Connection Pool Properties**.
   You see that **Maximum Connections** is set to 10 by default.
3. Set **Maximum Connections** to 50. You may want to set it to a higher value if necessary.
4. Click **Save**.

Next, define custom properties.

To define custom properties:

1. In the list of JDBC data sources, select the check box for the newly created **jasperserver** data source, and click **Test Connection**.
2. In the Messages area a success or failure message appears. The failure message gives you information about which custom properties you need to define.

3. Edit the following properties and add the ones that don't exist, then save the changes:

Table 7-3 Properties for TIBCO JDBC Driver for DB2

Property Name	Value
alternateID	JSPRSRVR
batchPerformanceWorkaround	true
currentSchema	JSPRSRVR
databaseName	JSPRSRVR
serverName	localhost
portNumber	50000

To see a list of all properties available for the TIBCO JDBC driver for DB2, see the Progress DataDirect documentation at http://media.datadirect.com/download/docs/jdbc/alljdbc/help.html and navigate to **User's Guide > DataDirect Connect Drivers > DB2 Driver > Connection Properties**.

Table 7-4 Properties for Vendor JDBC Driver for DB2

Property Name	Value
currentSchema	JSPRSRVR
fullyMaterializeLobData	true
fullyMaterializeInputStreams	true
progressiveStreaming	2
progressiveLocators	2

4. Go back to the list of JDBC data sources, select the check box for the **jasperserver** data source, and click **Test Connection**.

Next, if you installed sample databases, sugarcrm and foodmart, create data sources to use them.

To create optional sugarcrm and foodmart data sources:

1. If you plan to run the sample reports, use the following values to create the foodmart and sugarcrm JNDI data sources:

Table 7-5 Field Values for Optional Data Sources for All DB2 Drivers with WebSphere

Field Name	Value	
Data source name	foodmart	sugarcrm

Field Name	Value	
JNDI name	jdbc/foodmart	jdbc/sugarcrm
Component-managed authentication alias	<node>/db2admin_user	
Database name	foodmart	sugarcrm
Driver type	4	
Server name	localhost	
Port number	50000	
Use this data source in CMP	selected	

Table 7-6 Custom Properties for TIBCO JDBC Driver for DB2 with WebSphere

Property Name	Value	
alternateID	FOODMART	SUGARCRM
batchPerformanceWorkaround	true	
currentSchema	FOODMART	SUGARCRM
databaseName	FOODMART	SUGARCRM
serverName	localhost	

Table 7-7 Custom Properties for Vendor's Driver for DB2 with WebSphere

Property Name	Value	
currentSchema	FOODMART	SUGARCRM
resultSetHoldability	1	

2. Click **Save directly to the master configuration**.
3. Set the connection pool size as described in **"To set the connection pool size" on page 82**.

Next, deploy the WAR file in WebSphere as described in **7.1.5, "Deploying the JasperReports Server WAR File in WebSphere," on page 88**.

7.1.4.3 Defining a JNDI Name and Sample Data Sources for Oracle

To define the JDBC data source and expose it through JNDI:

1. Click the name of the JDBC provider that you just created. For example, **TIBCO JDBC Provider - Oracle** or **Oracle JDBC Driver**.
2. Click Data sources under the Additional Properties of the JDBC provider details panel.

3. To create a new data source, click New and the new data source wizard appears.
4. Enter the data source name: `jasperserver`
5. Enter the JNDI name: `jdbc/jasperserver`
6. Click **Next**.
7. (If you are using the TIBCO JDBC Provider, skip to **step 9**.) If you are using the vendor's Oracle driver,choose **Select an existing JDBC provider**, then select **Oracle JDBC Driver** from the drop-down list.
8. Click **Next** and enter the following values:

Field Name	Value
URL	jdbc:oracle:thin:@localhost:1521:orcl
Data store helper class name	Oracle11g data store helper
Use this data source in CMP	selected

9. Click **Next** and in **Setup security alias**, set **Component-managed authentication alias** to the following value:

   `<node>/jasperserver_user`
10. Click **Next**, review the summary information and click **Finish**.

Next, set the connection pool size.

To set the connection pool size:

1. In the list of JDBC data sources, click on the newly created **jasperserver** data source to edit it.
2. Click **Additional Properties > Connection Pool Properties**.

   You see that **Maximum Connections** is set to 10 by default.
3. Set **Maximum Connections** to 50. You may want to set it to a higher value if necessary.
4. Click **Save**.

Next, define custom properties.

To define custom properties:

1. In the list of JDBC data sources, select the check box for the newly created **jasperserver** data source, and click **Test Connection**.

   In the **Messages** area a success or failure message appears. The failure message gives you information about which custom properties you need to define.
2. Navigate to the jasperserver data sources **General Properties** page.
3. In **Additional Properties** on the right side of the **General Properties** page of the jasperserver data source, click **Custom properties**.

4. Edit the following properties and add the ones that don't exist, then save the changes. The following table shows how to set the custom properties if you are using the TIBCO JDBC driver for Oracle:

Table 7-8 Custom Properties for TIBCO JDBC Driver for Oracle

Property Name	Value
serverName	localhost
SID	ORCL
portNumber	1521
batchPerformanceWorkaround	true

To see a list of all properties available for the TIBCO JDBC driver for Oracle, see the Progress DataDirect documentation at http://media.datadirect.com/download/docs/jdbc/alljdbc/help.html and navigate to **User's Guide > DataDirect Connect Drivers >Oracle Driver > Connection Properties**.

5. Go back to the list of JDBC data sources, select the check box for the **jasperserver** data source, and click **Test Connection**.

Next, if you installed sample databases, sugarcrm and foodmart, create data sources to use them.

To create optional sugarcrm and foodmart data sources:

1. If you plan to run the sample reports, use the following values to create the foodmart and sugarcrm JNDI data sources:

Field Name	Value	
Data source name	foodmart	sugarcrm
JNDI name	jdbc/foodmart	jdbc/sugarcrm
Component-managed authentication alias	<node>/foodmart_user	<node>/sugarcrm_user

Property Name	Value
portNumber	1521
serverName	localhost

2. Click **Save directly to the master configuration**.
3. Set the connection pool size as described in **"To set the connection pool size" on page 85**.

Next, deploy the WAR file in WebSphere as described in **7.1.5, "Deploying the JasperReports Server WAR File in WebSphere," on page 88**.

7.1.4.4 Defining a JNDI Name and Sample Data Sources for SQL Server

To define a JDBC provider:

1. Click the name of the JDBC provider that you just created. For example, **Microsoft SQL Server JDBC Driver**.
2. Click Data sources under the Additional Properties of the JDBC provider details panel.
3. To create a new data source, click New and the new data source wizard appears.
4. Enter the data source name: `jasperserver`
5. Enter the JNDI name: `jdbc/jasperserver`
6. Click **Next**.
7. (If you are using the TIBCO JDBC Provider, skip this step.) If you are using the vendor's SQL Server driver, choose **Select an existing JDBC provider**, then select **Microsoft SQL Server JDBC Driver** from the drop-down list.
8. Click **Next** and in Setup security alias, set Component-managed authentication alias to the following value:
   `jasperserver_user`
9. Click **Next**, review the summary information and click **Finish**.

Next, set the connection pool size.

To set the connection pool size:

1. In the list of JDBC data sources, click on the newly created **jasperserver** data source to edit it.
2. Click **Additional Properties > Connection Pool Properties**.
   You see that **Maximum Connections** is set to 10 by default.
3. Set **Maximum Connections** to 50. You may want to set it to a higher value if necessary.
4. Click **Save**.

Next, define custom properties.

To define custom properties:

1. In the list of JDBC data sources, select the check box for the newly created **jasperserver** data source, and click **Test Connection**.
   In the Messages area a success or failure message appears. The failure message gives you information about which custom properties you need to define.
2. Navigate to the jasperserver data sources **General Properties** page.
3. In **Additional Properties** on the right side of the **General Properties** page of the jasperserver data source, click **Custom properties**, and define properties. The following table shows how to set properties if you are using the TIBCO JDBC driver for SQL Server

Table 7-9 Custom Properties for TIBCO JDBC Driver for SQL Server

Property Name	Value
batchPerformanceWorkaround	true
databaseName	jasperserver
serverName	localhost
portNumber	1433

To see a list of all properties available for the TIBCO JDBC driver for SQL Server, see the Progress DataDirect documentation at http://media.datadirect.com/download/docs/jdbc/alljdbc/help.html and navigate to **User's Guide > DataDirect Connect Drivers >Microsoft SQL Server Driver > Connection Properties**.

Next, if you installed sample databases, sugarcrm and foodmart, create data sources to use them.

To create optional sugarcrm and foodmart data sources:

If you plan to run the sample reports, use the following values to create the foodmart and sugarcrm JNDI data sources:

1. In the list of JDBC data sources, click the link for the newly created **jasperserver** data source.
2. Click **Save directly to the master configuration**.
3. Set the connection pool size as described in **"To set the connection pool size" on page 87**.

Next, deploy the WAR file in WebSphere.

7.1.5 Deploying the JasperReports Server WAR File in WebSphere

To deploy the JasperReports Server WAR file in WebSphere:

1. In the Administrative Console, navigate to **Applications > New Application** and select **New Enterprise Application.**

   JasperReports Server is a modern application, based on Java Servlet version 2.4, so you do not select the older, WebSphere V4-compliant application type.
2. Browse to `<js-install>/jasperserver-pro.war` on the local file system. Keep the default setting (**Fast path** selected) and click **Next**.
3. On the Select installation options page, accept all the default settings, and select **Save**. Click **Next**.
4. On the Map modules to servers page, make sure the JasperReports Server module is mapped to the cell, node, and server that you want. Click **Next**.
5. On the Map modules to servers page, select **jasperserver**. Click **Next**.
6. On the Map virtual hosts page, choose the **JasperServer UI application module**. Click **Next.**
7. In the Map context roots for Web modules, enter `jasperserver-pro`.
8. Click **Next**, review the summary information and start the installation process. (The installation process may take a while.)
9. Click **Save directly to the master configuration**.

Next, set JVM Options. Required.

7.1.6 Setting JVM Options

To set the Java JVM Options:

For the JasperReports Server XML/A functionality to work, special Java JVM options need to be set to resolve class conflicts between the WebSphere and JasperReports Server web services implementation. JVM options also provide the optimal resources for running JasperReports Server.

To configure your Java JVM options:

1. Select **Enterprise Applications > jasperserver-pro_war > Target specific application status > (server name)**.
2. Expand **Java and Process Management** > **Process Definition** > **Java Virtual Machine** > **Generic JVM arguments.**
3. In the text box named **Generic JVM Options**, paste in the following JVM options that explicitly specify JasperReports Server classes for AXIS and Xalan, as well as optimize JVM resources:

Generic JVM Options on Windows	
Options for all databases	`-Dclient.encoding.override=UTF-8 -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 -Djavax.xml.transform.TransformerFactory=org.apache.xalan.processor.TransformerFactoryImpl`
For Oracle	`set JAVA_OPTS=%JAVA_OPTS% -Doracle.jdbc.defaultNChar=true`

Generic JVM Options on Linux	
Options for all databases	`-Dclient.encoding.override=UTF-8 -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 -Djavax.xml.transform.TransformerFactory=org.apache.xalan.processor.TransformerFactoryImpl`
For Oracle	`export JAVA_OPTS="$JAVA_OPTS -Doracle.jdbc.defaultNChar=true"`

Setting the Oracle localization option, defaultNChar, can substantially impact the performance of JDBC queries. If you do not need to support UTF-8 for your Oracle database, you can omit this setting.

4. Click **Save** on the console task bar.

To configure class loading:

1. Select **Enterprise Applications > jasperserver-pro_war > Class loading and update detection**.
2. In the section **Class loader order**, select **Classes loaded with local class loader first (parent last)**.
3. (WebSphere 8.x only; if you are using WebSphere 7.x, skip this step.) In the section **WAR class loader policy** select **Single class loader for application**.
4. Save directly to your master configuration.
5. Restart WebSphere.

Next, start the jasperserver-pro application in WebSphere.

7.1.7 Starting and Restarting JasperReports Server

To start the jasperserver-pro application:

1. Restart WebSphere.
2. In the Administrative Console, navigate to: **Applications > Application Types > WebSphere Enterprise Application**.
3. Select the check box next to the **jasperserver-pro** application, then click **Start** to start JasperReports Server.
   If you make configuration changes to your JasperReports Server instance, restart JasperReports Server.
4. Log into JasperReports Server.

7.2 Logging into JasperReports Server

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

1. Log in by going to the following URL:
   `http://<hostname>:9080/jasperserver-pro`
   Where <hostname> is localhost, a machine name, or an IP address. The login page should appear after some time to compile the necessary JSP files.
2. Use the following administrative identity to log into the system:

User ID	Password	Description
`superuser`	`superuser`	System-wide administrator
`jasperadmin`	`jasperadmin`	Administrator for the default organization

If you logged in successfully, the JasperReports Server home page appears. If you have trouble logging in and get the following error message, you may be running at a WebSphere patch level that needs a further configuration:

Page cannot be found, HTTP 404 error

Refer to troubleshooting section **A.9.9, "WebSphere Modifications," on page 126**.

The first time you log into JasperReports Server, you will be prompted to opt-in or opt-out of the JasperReports Server Heartbeat. For more information, refer to **5.6.1, "JasperReports Server Heartbeat," on page 41**.

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

7.3 Configuring Report Scheduling

The scheduled reporting feature of JasperReports Server allows reports to be run at pre-configured time intervals. Optionally, notification email can be sent to users to let them know that new reports are available.

7.3.1 Additional Fix for Scheduled Report with JNDI Data Source

Under the WebSphere application server, if you have a scheduled report that uses a JNDI data source you will need to make an additional configuration in order for the JNDI lookup to be resolved correctly.

There are special WebSphere specific configuration files included in the jasperserver-pro.war archive. These files are generated during the installation process. In order to enable this fix, you will need to rename the WebSphere specific files to remove the prefix `webSphere`. Two of the file rename operations will overwrite the existing configuration file names.

Rename:	WEB-INF/webSphere-applicationContext-report-scheduling-wm.xml
To:	WEB-INF/applicationContext-report-scheduling-wm.xml
Rename:	WEB-INF/webSphere-js.quartz.base.properties
To:	WEB-INF/js.quartz.base.properties (overwrite existing file)
Rename:	WEB-INF/webSphere-js.scheduling.properties
To:	WEB-INF/js.scheduling.properties (overwrite existing file)

Notes on this configuration change:

Since JasperServer 3.7 and 4.0, a new work manager class is used to run scheduled report jobs under WebSphere. The JNDI name of the work manager is configured in the WebSphere version of js.scheduling.properties. Also, for WebSphere, we need a different default value for the specified JNDI name (wm/default). This is also defined in the WebSphere version of js.scheduling.properties.

The number of threads that run report jobs is no longer configured in js.quartz.base.properties, but is instead provided by the work manager configuration.

7.3.2 Additional Change for Mail Server Authentication

If your mail server requires authentication, edit the applicationContext-report-scheduling.xml file after applying the changes above.

1. Extract the file from the WAR archive:
   ```
   "%JAVA_HOME%\bin\jar" xf jasperserver-pro.war WEB-INF/applicationContext-report-scheduling.xml
   ```
2. Open the file for editing and locate the `reportSchedulerMailSender` bean.
3. Set the `javaMailProperties key="mail.smtp.auth"` value to `true`.
4. Save the file and replace it in the archive:
   ```
   "%JAVA_HOME%\bin\jar" uf jasperserver-pro.war WEB-INF/applicationContext-report-scheduling.xml
   ```
5. Delete the WEB-INF directory that was created, along with the file it contains.

For more information about setting up report scheduling, refer to **6.6, "Configuring Report Scheduling," on page 69**.

7.4 Updating XML/A Connection Definitions (Optional)

If you have loaded the JasperReports Server sample data, and you would like to run the Analysis XML/A examples, you will need to update the XML/A connection resources to use the correct web port.

The typical port used by WebSphere is 9080. Follow the procedure in **6.7, "Updating XML/A Connection Definitions," on page 73**.

7.5 Troubleshooting your JasperReports Server Configuration

7.5.1 Startup Problems

The most common problems are errors in the database configuration. These problems are typically incorrect configurations within the database configuration files or in the application server configuration files. For information about resolving these types of errors, refer to troubleshooting section **Appendix A, "Troubleshooting," on page 109**.

7.5.2 Error Running Report

If you have trouble running reports in your new JasperReports Server instance, refer to troubleshooting section **A.8.10, "Error Running a Report," on page 118**. If you are having trouble running the MDX example Topic or SugarCRM OLAP view, you need to update the port for XML/A connections. See **7.4, "Updating XML/A Connection Definitions (Optional)," on page 91**.

7.5.3 Filter Error Using MySQL

The following error could be caused by an incorrect ampersand setting on your data source configuration:

```
Error 500: Filter [characterEncodingProxyFilter]: cold not be initialized
```

The data source line needs to have `&` and not `&` in order to be evaluated correctly. That is, the URL you enter in the procedure to define the JDBC data source and expose it through JNDI should look like the following:

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

Note the ampersand string is `&` and not `&`.

7.5.4 Error Creating Internationalized Name

If you encounter errors in JasperReports Server when creating resources with internationalized names, and you have an Oracle database, configure your Oracle JDBC driver. Set the Oracle-specific option listed in the tables of **7.1.6, "Setting JVM Options," on page 88**.

7.5.5 Xerces Error

In earlier releases of JasperReports Server it was possible to find the following error in the WebSphere log:

```
SRVE0068E: Uncaught exception thrown in one of the service methods of the servlet:
jasperserver. Exception thrown: org.springframework.web.util.NestedServletException:
javax.xml.validation.SchemaFactoryFinder$ConfigurationError: Provider
org.apache.xerces.jaxp.validation.XMLSchemaFactory could not be instantiated:
org.apache.xerces.impl.dv.DVFactoryException: DTD factory class
org.apache.xerces.impl.dv.dtd.DTDDVFactoryImpl does not extend from DTDDVFactory.
```

Since around release 4.0, the xercesImpl jar used is version 2.7.1 and more recently 2.10.0.

The error shown above is caused by a conflict between the IBM JDK which is used by WebSphere and the xercesImpl-2.6.2 library which is bundled in older versions of JasperReports Server. There are two solutions:

- Remove the xercesImpl library from the following location:

```
<websphere>\profiles\AppSrv<NN>\installedApps\<node>\jasperserver-pro_war.ear\
   jasperserver-pro.war\WEB-INF\lib
```

- Update the xercesImpl library to a new version if it is an old version.
- The newer xercesImpl jars do not cause this error.

7.5.6 OLAP View Fails With Exception

The following error could caused by AspectJ requiring class loaders to be tried out in a specific order:

```
java.lang.NoSuchMethodError: org/aspectj/runtime/reflect/Factory.makeMethodSig(
  java/lang/String;
  ...)
  org/aspectj/lang/reflect/MethodSignature;
```

Change the default class loader policy using the following procedure:

1. In the WebSphere Administrative Console, navigate to **Applications > (app-name) > Manage Modules > JasperServer UI application**.
2. Change the following setting:

Property Name	Value
Class loader order	Class loaded with local class loader first (parent last)

3. Click **OK**.
4. Save the master configuration.
5. Restart the WebSphere server.

CHAPTER 8 INSTALLING THE WAR FILE FOR WEBLOGIC

JasperReports Server supports deployment on the WebLogic Application Server, but requires its own database to store information such as users, organizations, and the repository. WebLogic users need the WAR file distribution to install JasperReports Server. Download the WAR file distribution from Jaspersoft technical support or contact your sales representative. The WAR file distribution comes in a file named jasperreports-server-6.0-bin.zip, the compressed ZIP format.

The WAR file distribution includes two sample databases containing data for optional demos. For evaluation, Jaspersoft recommends you install the sample databases. In a production environment, you typically don't install the sample databases. You create and initialize the required repository database and the optional sample databases before JasperReports Server is deployed in WebLogic. Deployment is performed by the WebLogic administrator using the WebLogic Administrative Console or domain config.xml.

This chapter contains the following sections:

8.1 Procedure for Installing the WAR File for WebLogic

To meet prerequisites for installing the WAR file for WebLogic:

1. Check that the Oracle/Sun Java JDK 1.6 or 1.7 is installed.
2. Check that the `JAVA_HOME` system environment variable points to the JDK.
3. Install the PostgreSQL, MySQL, Oracle, SQL Server, or DB2 database.

The target database can be on a remote server.

To install the WAR file for WebLogic:

1. Extract all files in jasperreports-server-6.0-bin.zip. Choose a top-level directory, such as `C:\Jaspersoft` on Windows or `/home/<user>` on Linux, for the extracted files.

   Unpacking the ZIP file creates the directory jasperreports-server-6.0-bin.
2. Check that WebLogic is installed in the default location on your local machine.

   If WebLogic is not installed in the default location, or if you encounter problems using the buildomatic scripts, set up the database manually, as described in **6.1, "Setting JVM Options for Application Servers," on page 47**. After setting up the database manually, skip **step 6** through **step 9**, and proceed to **step 10**.
3. (If you are using MySQL, you can skip this step.) Copy your JDBC driver to WebLogic.
   a. PostgreSQL example: Copy the JDBC jar from

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

      to

      `<weblogic_home>/server/lib`

   b. TIBCO JDBC Oracle driver example: Copy ji-oracle-1.0.0-driver.jar from

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

      to

      `<weblogic_home>/server/lib`

   Note that the MySQL JDBC driver is included in recent versions of WebLogic by default.
4. (If you are using PostgreSQL or MySQL, you can skip this step). If you are using the TIBCO JDBC driver for Oracle, SQL Server, or DB2, you need to add the driver to the WebLogic jdbcdrivers.xml file. To do this:
   a. Open the file <weblogic_home>/server/lib/jdbcdrivers.xml in a text editor.
   b. Add the correct settings for your database. The following example shows the settings for the TIBCO JDBC driver for Oracle. For DB2 or SQL Server, see **8.3.1, "Configuring TIBCO JDBC Driver Connections," on page 104**.

```
<Driver
    Database="Oracle"
    Vendor="Tibco"
    Type="Thin"
    DatabaseVersion="10g and later"
    ForXA="false"
    Cert="false"
    ClassName="tibcosoftware.jdbc.oracle.OracleDriver"

URLHelperClassname="weblogic.jdbc.utils.OracleJDBC4DriverURLHelper$SIDHelper"
    Description="tibco driver for Oracle">
        <Attribute Name="DbmsName" Required="true" InURL="true"/>
        <Attribute Name="DbmsHost" Required="true" InURL="true"/>
        <Attribute Name="DbmsPort" Required="true" InURL="true" DefaultValue="1521"/>
        <Attribute Name="DbmsUsername" Required="true" InURL="false"/>
        <Attribute Name="DbmsPassword" Required="true" InURL="false"/>
  </Driver>
```

   c. Save the file

5. (If you are using MySQL, you can skip this step.) Set your WebLogic classpath for your database:
   a. Add the following line to your <weblogic_home>/bin/setDomainEnv.cmd/sh file as follows:
      On Windows:

```
set CLASSPATH=<weblogic_home>\server\lib\<jdbc-driver-name>.jar;%CLASSPATH%
set CLASSPATH={pathToCommonsLangJar}\commons-lang-2.5.jar;%CLASSPATH%
```

      On Linux:

```
export CLASSPATH=<weblogic_home>/server/lib/<jdbc-driver-name>.jar:$CLASSPATH
commons-lang-2.5.jar can be found in jasperserver WAR (WEB-INF/lib)
```

   b. Save the file.
   c. Restart WebLogic using the `startWebLogic.cmd/sh`.
6. Copy the .properties file for your database:
   - From — <js-install>/buildomatic/sample_conf/
   - To — <js-install>/buildomatic
7. Rename the file you copied to `default_master.properties` file.
8. Edit the default_master.properties file to add the settings that are specific to your database and your application server. **Table 8-1** shown example property settings.

When appServerType = skipAppServerCheck, buildomatic skips the application servertype validation. Use this setting when installing JasperReports Server with WebLogic. Backslashes in appServerDir must be doubled, for example C:\\WL\\Application_Server. Make sure there are no spaces in the appServerDir path.

Table 8-1 Sample Values for the default_master.properties File

Database	Sample Property Values

PostgreSQL	`appServerType=skipAppServerCheck`
	`appServerDir=[path to WebLogic application server]`
	`dbUsername=postgres`
	`dbPassword=postgres`
	`dbHost=localhost`

DB2	`appServerType=skipAppServerCheck`
	`appServerDir=[path to WebLogic application server]`
	`dbUsername=db2admin`
	`dbPassword=password`
	`dbHost=localhost`
	For DB2 8.x, change your deployed JDBC driver as described in**6.5, "Locating and Changing Buildomatic Configuration Files," on page 65**.

MySQL	`appServerType=skipAppServerCheck`
	`appServerDir=[path to WebLogic application server]`
	`dbUsername=root`
	`dbPassword=password`
	`dbHost=localhost`

Database	Sample Property Values
Oracle	appServerType=skipAppServerCheck appServerDir=[path to WebLogic application server] sysUsername=system sysPassword=password dbUsername=jasperserver dbPassword=password dbHost=hostname Note that dbUsername must be the same as the Oracle user name.
SQL Server	appServerType=skipAppServerCheck appServerDir=[path to WebLogic application server] dbUsername=sa dbPassword=sa dbHost=localhost If your application server runs on Java 1.5, change your deployed JDBC driver as described in **6.5, "Locating and Changing Buildomatic Configuration Files," on page 65**.

9. Setup the database and optional sample databases using the buildomatic Ant scripts. Enter commands in the following table to call buildomatic Ant scripts:

Exception: For DB2, skip this step and perform **step 1** to **step 3** in **6.3.4, "DB2," on page 58**, then go to the next step (**step 10**) of this procedure.

You call buildomatic Ant scripts from the command line using the following syntax:

```
Windows — js-ant <target-name>
Linux — ./js-ant <target-name>
```

Commands	Description
cd <js-install>/buildomatic	Goes to the buildomatic directory.
js-ant create-js-db	Create the jasperserver repository database
js-ant init-js-db-pro js-ant import-minimal-pro	Initializes database, loads core application data
js-ant create-sugarcrm-db js-ant create-foodmart-db	(Optional) Creates sample databases
js-ant load-sugarcrm-db js-ant load-foodmart-db	(Optional) Loads sample data into the sample databases
js-ant import-sample-data-pro	(Optional) Loads the demos that use the sample data

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 112**.

10. Add the database driver to your classpath.
11. In WebLogic, open an Administrative Console window, and navigate to **Services > Data Sources** or **Domain Configurations > Services > Data Sources**.
12. Click **New** and then **Generic Data Source** for each of the data source columns in the following table, and enter the following values for a PostgreSQL database. You will need to click **Next** after entering the database driver and after **One-Phase Commit**.

To use a database other than PostgreSQL, configure the database connections using settings shown in **8.3, "Configuring Other Database Connections," on page 104**.

If you plan on using the sample databases (Foodmart and Sugar CRM), perform this step and the following step for each database.

Parameter Name	JasperReports Server	Foodmart Example	Sugar CRM Example
Name	JasperServerDataBase	FoodmartDataBase	SugarcrmDataBase
JNDI Name	JasperServerDataBase	FoodmartDataBase	SugarcrmDataBase
Database Type	PostgreSQL		
Database Driver	PostgreSQL Driver Versions: using org.postgresql.Driver		
Supports Global Transactions	Selected		
One-Phase Commit	Selected		

13. Set connection properties. Sample properties for a PostgreSQL database are:

Parameter Name	JasperReports Server	Foodmart Example	Sugar CRM Example
Database Name	jasperserver	foodmart	sugarcrm
Host Name	localhost		
Port	5432		
Database User Name	postgres		
Password	postgres		
Confirm Password	postgres		

14. Test the database connection:
    a. For SugarCRM and Foodmart, use the default connections:
    jdbc:postgresql://localhost:5432/sugarcrm

jdbc:postgresql://localhost:5432/foodmart

b. Change the URL for the `jasperserver` database to:

jdbc:postgresql://localhost:5432/jasperserver

15. Select targets, and ensure that **AdminServer** is set for all data sources.
16. Using the Java jar tool or an unzip tool, unpack the jasperserver-pro.war file. For example, enter these commands to use the Java jar tool:

```
cd <js-install-dir>
mkdir jasperserver-pro
cd jasperserver-pro
"%JAVA_HOME%/bin/jar" xvf ../jasperserver-pro.war
```

The jasperserver-pro.war file is unpacked to a folder.

17. Search for conflicting JARs and delete them from the WAR file if necessary. Some installations of WebLogic already contain the following JARs. If these JARs are present in your WebLogic installation, you need to delete them from your JasperReports Server installation to avoid conflicts. To do this:
    a. Search your WebLogic installation for the following files:

    jaxb-api-<ver>.jar

    jaxb-impl-<ver>.jar

    serializer-<ver>.jar

    stax-api-<ver>.jar

    xalan-<ver>.jar

    xercesImpl-<ver>.jar

    xml-apis-<ver>.jar

    b. Change to the JasperReports Server WEB-INF/lib directory:

    `cd <js-install>/jasperserver-pro/WEB-INF/lib`

    c. Delete any conflicting JARs, but save a copy of them to a temporary location.

    If you experience a problem with OLAP Views, you need these JARs for the workaround described in **8.10, "Troubleshooting Problems with OLAP Views," on page 108**.

18. If you are using WebLogic 10.3.6, you need to replace xml-apis-1.3.04.jar with xml-apis-2.10.0.jar. If you are using WebLogic 10.3.5 or WebLogic 12c, skip this step. To update the JAR:
    a. Go to the WEB-INF/lib directory.
    b. Copy xml-apis-2.10.0.jar from <js-install>/buildomatic/install_resources/extra-jars
    c. Remove xml-apis-1.3.04.jar.
19. Update your Hibernate, Quartz, and Mail Server configuration:
    a. The `hibernate.properties` and `js.quartz.properties` files have been already configured for your database type by the buildomatic logic. So, these pre-configured files can be copied to the `jasperserver-pro` file:

    Copy from:

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

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

    To:

    jasperserver-pro/WEB-INF

b. Edit the scheduler URI port value for WebLogic in the `js.quartz.properties`:

   Edit `js.quartz.properties`:

   Set :

   ```
   report.scheduler.web.deployment.uri=http://localhost:8080/jasperserver-pro
   ```

   To:

   ```
   report.scheduler.web.deployment.uri=http://localhost:7001/jasperserver-pro
   ```

c. If you want to configure JasperReports Server to automatically schedule and email reports, enter your mail server information in the `js.quartz.properties` file. Modify all `report.scheduler.mail.sender.*` properties as necessary for your mail server.

20. If your mail server requires authentication, edit the `applicationContext-report-scheduling.xml` file in the same manner:
    a. Open the `jasperserver-pro/WEB-INF/applicationContext-report-scheduling.xml` file for editing and locate the `reportSchedulerMailSender` bean.
    b. Set the `javaMailProperties key="mail.smtp.auth"` value to `true`.
21. Now you can change to the `jasperserver-pro` folder and rearchive the `jasperserver-pro.war` file, using commands such as the following:

Commands	Description
`cd ../..`	Changes to the jasperserver-pro folder
`mv ../jasperserver-pro.war ../BAK-jasperserver-pro.war`	Renames the original jasperserver-pro.war file.
`"%JAVA_HOME%/bin/jar" cvf ../jasperserver-pro.war *`	Rearchives the jasperserver-pro.war file.
`cd ..` `mv jasperserver-pro BAK-jasperserver-pro`	Renames the unneeded working folder to a backup location.

You now have a `jasperserver-pro.war` file that can be used for deploying to WebLogic.

22. Edit your WebLogic domain configuration file `<wl-domain>/config/config.xml`:

<wl-domain> is the path of the domain within WebLogic that contains your JasperReports Server deployment. For example `<weblogic>/samples/domains/wl_server`.

a. Locate the `server` and `security-configuration` elements, and insert the following parameters:

```
<server>
...
       <stuck-thread-max-time>1200</stuck-thread-max-time>
       <listen-address></listen-address>
</server>
<security-configuration>
```

```
        ...
        <enforce-valid-basic-auth-credentials>false</enforce-valid-basic-auth-credentials>
</security-configuration>
```

b. Check that the `stuck-thread-max-time` element appears above the `listen-address` element before the closing `</server>` tag.

In some cases, setting the `stuck-thread-max-time` may cause a schema validation error. In this case, you can try removing this line from the configuration file.

23. Copy the file `<js-install>/buildomatic/conf_source/iePro/lib/xercesImpl-2.10.0.jar` to the folder `<weblogic_home>/server/lib`.
24. Edit the `setdomainEnv` file to force WebLogic to use the correct version of the Joda time library, which is shipped with JasperReports Server. By default, the Joda time library is in:
    `<js-install>/buildomatic/conf_source/iePro/lib`.
    - Windows: Edit the `setDomainEnv.bat` file and add the following line:
      `set PATCH_CLASSPATH=${PATH TO NEW JODA LIBRARY}\joda-time-2.3.jar`
    - Linux: Edit the `setDomainEnv.sh` file and add the following line:
      `export PATCH_CLASSPATH=${PATH TO NEW JODA LIBRARY}/joda-time-2.3.jar`
25. Set JVM options as described in **8.2, "Setting Java Properties," on page 103**.

Deploy JasperReports Server to WebLogic:

1. Enable the **Lock & Edit** button:
   a. Select the **Preferences** link at the top of the Admin console
   b. On the **User Preferences** screen, scroll to the bottom of the screen and deselect **Automatically Acquire Lock** and **Activate Changes**.
   c. Save.
2. In the Administrative Console, click the **Lock & Edit** button and navigate to **Deployments**.
3. On the Deployments page click the **Install** button.
4. Select the path to `<js-install>`. Click **Next**.
5. Leave the radio button selected for **Install this deployment as an application**. Click **Next**.
6. When prompted, enter the following parameter values:

Parameter Name	Example Value
Name	jasperserver-pro
Security	Custom Roles and Policies
Source accessibility	Use the defaults defined by the deployment's targets

7. Review your choices and click **Finish**.
8. Click **Save**.

8.2 Setting Java Properties

Edit your WebLogic startup script to include the settings described in the following tables, according to your platform. Substitute the location of your JasperReports Server license file where necessary:

WebLogic Startup Settings on Windows	
Filename	<wl-domain>\bin\startWebLogic.cmd
Settings	set JAVA_OPTIONS=%JAVA_OPTIONS% -Djs.license.directory=C:\<js-install>\ -Dfile.encoding=UTF-8 -Dcom.sun.xml.namespace.QName.useCompatibleSerialVersionUID=1.0 -Djavax.xml.soap.SOAPConnectionFactory=org.apache.axis.soap.SOAPConnectionFactoryImpl -Djavax.xml.soap.MessageFactory=org.apache.axis.soap.MessageFactoryImpl -Djavax.xml.soap.SOAPFactory=org.apache.axis.soap.SOAPFactoryImpl -Xms1024m -Xmx2048m -XX:PermSize=128m -XX:MaxPermSize=512m -Xss2m
For Oracle	set JAVA_OPTIONS=%JAVA_OPTIONS% -Doracle.jdbc.defaultNChar=true

Setting the Oracle localization option, defaultNChar, can substantially impact the performance of JDBC queries. If you do not need to support UTF-8 for your Oracle database, you can omit this setting.

WebLogic Startup Settings on Linux	
Filename	<wl-domain>/bin/startWebLogic.sh
Settings	export JAVA_OPTIONS="$JAVA_OPTIONS -Djs.license.directory=/home/<user>/weblogic/jasperlicense/ -Dfile.encoding=UTF-8 -Dcom.sun.xml.namespace.QName.useCompatibleSerialVersionUID=1.0 -Djavax.xml.soap.SOAPConnectionFactory=org.apache.axis.soap.SOAPConnectionFactoryImpl -Djavax.xml.soap.MessageFactory=org.apache.axis.soap.MessageFactoryImpl -Djavax.xml.soap.SOAPFactory=org.apache.axis.soap.SOAPFactoryImpl -Xms1024m -Xmx2048m -XX:PermSize=128m -XX:MaxPermSize=512m -Xss2m"
For Oracle	export JAVA_OPTIONS="$JAVA_OPTIONS -Doracle.jdbc.defaultNChar=true"

Setting the Oracle localization option, defaultNChar, can substantially impact the performance of JDBC queries. If you do not need to support UTF-8 for your Oracle database, you can omit this setting.

8.3 Configuring Other Database Connections

8.3.1 Configuring TIBCO JDBC Driver Connections

8.3.1.1 Configuring a TIBCO JDBC Oracle Connection

If you are using the TIBCO JDBC driver for Oracle, add the driver to the WebLogic jdbcdrivers.xml file, as described in **step 4**in **8.1, "Procedure for Installing the WAR File for WebLogic," on page 95**

To use the driver, select the TIBCO driver for Oracle from the menu after **step 11** in **8.1, "Procedure for Installing the WAR File for WebLogic," on page 95** and enter the following properties:

Parameter Name	JasperReports Server	Foodmart Example	Sugar CRM Example
Name	JasperServerDataBase	FoodmartDataBase	SugarcrmDataBase
JNDI Name	JasperServerDataBase	FoodmartDataBase	SugarcrmDataBase
URL	jdbc:tibcosoftware:oracle://localhost:1521;SID=ORCL		
Driver Class Name	tibcosoftware.jdbc.oracle.OracleDriver		
Properties	SID=ORCL user=jasperserver	SID=ORCL user=foodmart	SID-ORCL user=sugarcrm

Make sure that **Test Table Name** is blank. If it says `SQL Null`, delete this.

8.3.1.2 Configuring a TIBCO JDBC DB2 Connection

If you are using the TIBCO JDBC driver for DB2, you need to add the following information to the WebLogic jdbcdrivers.xml file, as described in **step 4**in **8.1, "Procedure for Installing the WAR File for WebLogic," on page 95**

```
<Driver
    Database="DB2"
    Vendor="Tibco"
    Type="Type 4"
    Cert="true"
    DatabaseVersion="7.X and later"
    ForXA="false"
    ClassName="tibcosoftware.jdbc.db2.DB2Driver"
    URLHelperClassname="weblogic.jdbc.utils.WLDB2JDBC4DriverURLHelper">
        <Attribute Name="DbmsName" Required="true" InURL="false"/>
        <Attribute Name="DbmsHost" Required="true" InURL="true"/>
        <Attribute Name="DbmsPort" Required="true" InURL="true" DefaultValue="50000"/>
        <Attribute Name="DbmsUsername" Required="true" InURL="false"/>
        <Attribute Name="DbmsPassword" Required="true" InURL="false"/>
  </Driver>
```

To use the driver, select the TIBCO driver for DB2 from the menu after **step 11** in **8.1, "Procedure for Installing the WAR File for WebLogic," on page 95** and enter the following properties. For the URL, enter the

value of the `databaseName` lower in the table. For example, for the `jasperserver` database, the URL is `jdbc:tibcosoftware:db2://localhost:50000;databaseName=jsprsrvr`

Parameter Name	JasperReports Server	Foodmart Example	Sugar CRM Example
Name	JasperServerDataBase	FoodmartDataBase	SugarcrmDataBase
JNDI Name	JasperServerDataBase	FoodmartDataBase	SugarcrmDataBase
URL	jdbc:tibcosoftware:db2://localhost;databaseName=<databaseName>		
Driver Class Name	tibcosoftware.jdbc.db2.DB2Driver		
Properties	user=db2admin		
portNumber	5000		
databaseName	foodmart	jsprsrvr	sugarcrm
serverName	localhost		
batchPerformanceWorkaround	true		

Make sure that **Test Table Name** is blank. If it says `SQL Null`, delete this.

8.3.1.3 Configuring a TIBCO JDBC SQL Server Connection

If you are using the TIBCO JDBC driver for SQL Server, you need to add the following information to the WebLogic jdbcdrivers.xml file, as described in **step 4**in **8.1, "Procedure for Installing the WAR File for WebLogic," on page 95**

```
<Driver
    Database="SQL Server"
    Vendor="TIBCO"
    Type="Type 4"
    DatabaseVersion="7.0 and later"
    ForXA="false"
    Cert="true"
    ClassName="tibcosoftware.jdbc.sqlserver.SQLServerDriver"
    URLHelperClassname="weblogic.jdbc.utils.MSSQL2005JDBC4DriverURLHelper">
        <Attribute Name="DbmsName" Required="false" InURL="false"/>
        <Attribute Name="DbmsHost" Required="true" InURL="true"/>
        <Attribute Name="DbmsPort" Required="true" InURL="true"
DefaultValue="1433"/>
        <Attribute Name="DbmsUsername" Required="true" InURL="false"/>
        <Attribute Name="DbmsPassword" Required="true" InURL="false"/>
  </Driver>
```

To use the driver, select the TIBCO driver for SQL Server from the menu after **step 11** in **8.1, "Procedure for Installing the WAR File for WebLogic," on page 95** and enter the following properties. For the URL, enter the value of the `databaseName` lower in the table. For example, for the `jasperserver` database, the URL is `jdbc:tibcosoftware:sqlserver://localhost:1433;databaseName=jasperserver`

Parameter Name	JasperReports Server	Foodmart Example	Sugar CRM Example
Name	JasperServerDataBase	FoodmartDataBase	SugarcrmDataBase
JNDI Name	JasperServerDataBase	FoodmartDataBase	SugarcrmDataBase
URL	jdbc:tibcosoftware:sqlserver://localhost:1433;databaseName=<databaseName>		
Driver Class Name	tibcosoftware.jdbc.sqlserver.SQLServerDriver		
Properties	user=sa		
databaseName	foodmart	jasperserver	sugarcrm

Make sure that **Test Table Name** is blank. If it says `SQL Null`, delete this.

8.3.2 Configuring Databases Using the Vendor's Driver

Use these settings to connect to a database other than PostgreSQL using the database vendor's driver.

Database Setting	MySQL	Oracle	DB2	SQL Server
Host	localhost	localhost	localhost	localhost
Name or SID	jasperserver	Orcl	jsprsrvr	jasperserver
User	root	jasperserver	db2admin	sa
Password	password	password	password	sa
Port	3306	1521	50000	1433
Hibernate Dialect	MySQLInnoDB Dialect	OracleJICustom Dialect	DB2JICustomDialect	SQLServerJI CustomDialect
Quartz Driver Delegate	StdJDBCDelegate	StdJDBCDelegate	DB2v8Delegate	StdJDBCDelegate

8.4 Starting JasperReports Server

1. In the Administrative Console, navigate to **Deployments**.
2. Select the **jasperserver-pro** application with a checkmark, and click **Start**.
3. In the Start Application Assistant page, click **Yes**.

8.5 Logging into JasperReports Server

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

1. Login by going to the following URL:

   `http://<hostname>:7001/jasperserver-pro`

   Where `<hostname>` could be localhost, a machine name, or an IP address. The login page should appear after some time to compile the necessary JSP files.
2. Use the following credentials to log into the system:

User ID	Password	Description
`superuser`	`superuser`	System-wide administrator
`jasperadmin`	`jasperadmin`	Administrator for the default organization

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

The first time you log into JasperReports Server, you will be prompted to opt-in or opt-out of the JasperReports Server Heartbeat. For more information, refer to **5.6.1, "JasperReports Server Heartbeat," on page 41**.

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

8.6 Configuring Report Scheduling

The scheduled reporting feature of JasperReports Server allows reports to be run at pre-configured time intervals. Optionally, notification emails can be set to users to let them know that new reports are available.

For more information about setting up report scheduling, refer to **6.6, "Configuring Report Scheduling," on page 69**.

8.7 Restarting JasperReports Server

If you made configuration changes to your JasperReports Server instance, restart JasperReports Server.

8.8 Updating XML/A Connection Definitions (Optional)

If you loaded the JasperReports Server sample data and would like to run the XML/A examples, update the XML/A connection resources to use the correct web port.

The typical port used by WebLogic is 7001. Follow the procedure in **6.7, "Updating XML/A Connection Definitions," on page 73**.

8.9 Troubleshooting Your JasperReports Server Configuration

If you have problems running buildomatic scripts that set up the database, set up the database manually. For more information, see **6.1, "Setting JVM Options for Application Servers," on page 47**.

8.9.1 Startup Problems

When trying to run a new JasperReports Server instance, most typical problems encountered are errors in the database configuration.

These problems are typically related to having incorrect configurations within the database configuration files or in the application server configuration files. For more information about resolving these types of errors, refer to troubleshooting section **Appendix A, "Troubleshooting," on page 109**.

8.9.2 Error Running Report

If you have trouble running reports in your new JasperReports Server instance, refer to troubleshooting section **A.8.10, "Error Running a Report," on page 118**.

8.10 Troubleshooting Problems with OLAP Views

If you are using WebLogic 10.3.2 or lower, and you encounter a 404 Error when attempting to change a data cube in an OLAP View, follow these steps to work around the problem:

1. Add the following java option to `startWebLogic.cmd` or `startWebLogic.sh`:

Windows:
```
set JAVA_OPTS=%JAVA_OPTS% -Djavax.xml.transform.TransformerFactory=
org.apache.xalan.processor.TransformerFactoryImpl
```

Linux:
```
export JAVA_OPTS="$JAVA_OPTS -Djavax.xml.transform.TransformerFactory=
org.apache.xalan.processor.TransformerFactoryImpl"
```

2. Set EXT_PRE_CLASSPATH in setDomainEnv.cmd:

Windows:
```
set
EXT_PRE_CLASSPATH=%DOMAIN_HOME%/lib/serializer-2.7.1.jar;
%DOMAIN_HOME%/lib/xalan-2.7.1.jar;
%DOMAIN_HOME%/lib/xercesImpl-2.10.0.jar;
%DOMAIN_HOME%/lib/dom4j-1.6.1.jar;
%DOMAIN_HOME%/lib/jdom-1.1.jar;
%EXT_PRE_CLASSPATH%
```

Linux:
```
export
EXT_PRE_CLASSPATH="$DOMAIN_HOME/lib/serializer-2.7.1.jar:
$DOMAIN_HOME/lib/xalan-2.7.1.jar:
$DOMAIN_HOME/lib/xercesImpl-2.10.0.jar:
$DOMAIN_HOME/lib/dom4j-1.6.1.jar:
$DOMAIN_HOME/lib/jdom-1.1.jar:
$EXT_PRE_CLASSPATH"
```

3. Copy the JAR files listed in the previous step from the `jasperserver-pro.war/WEB-INF/lib` folder to %DOMAIN_HOME%/lib or $DOMAIN_HOME/lib.

APPENDIX A TROUBLESHOOTING

This appendix contains the following sections:

A.1 Binary Installer Freezes

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

A.1.1 Installer Log Files

If you 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-<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, “pasword$” or “pas$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.sh
js-upgrade-newdb.sh
js-upgrade-samedb.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 42**.

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 111**.

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.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.8.1.4 Logging into Oracle

Start SQL*Plus and try to log into Oracle directly. Three users are created during installation:

- `jasperserver` — schema user for the JasperReports Server metadata.
- `sugarcrm` — schema user for the SugarCRM sample data.
- `foodmart` — schema user for the foodmart sample data.

To log in as each of these users, supply the password specified during installation.

A.8.1.5 Logging into Microsoft SQL Server

Run the `sqlcmd` and try to log into MSSQL Server directly. For example:

```
sqlcmd -S localhost\jasperserver -d jasperserver -U jasperadmin -P password
```

A.8.1.6 Connectivity Errors with Vendor's Driver for SQL Server

If you are using the vendor's driver for SQL Server and have configured default_master.properties as described in **6.4.2.3, "SQL Server Example," on page 64**, you will see connection errors if you uncommented the following line:

```
# admin.jdbcUrl=jdbc:sqlserver://${dbHostOrInstance};SelectMethod=cursor
```

Make sure this line is commented.

A.8.2 Case Sensitive Collation in SQL Server

In Microsoft SQL Server, setting the collation to be case-sensitive is not supported. When collation is case-sensitive in SQL Server, column and table names are also treated as case-sensitive. This can happen when setting a locale that includes case-sensitive collation and will cause an error:

```
[sql] Failed to execute:
INSERT INTO JIUserRole (userId,roleId) select u.id, r.id
from JIUser u, JIRole r
where u.username = \'anonymousUser\' and r.roleName = \'ROLE_ANONYMOUS\'
  [sql] com.microsoft.sqlserver.jdbc.SQLServerException: Invalid column name \'roleName\'
```

Use a different locale or remove the case-sensitivity setting.

A.8.3 Maximum Packet Size in MySQL

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

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

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

```
max_allowed_packet = 16M
```

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

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

A.8.3.1 Connection reset by peer MySQL Error

If you are using the MariaDB JDBC driver in order to connect to the MySQL database, and you get an error similar to the following:

Could not send query:

Connection reset by peer: socket write error

This is the maximum packet size error seen in the previous section, with a different error message. 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 Oracle and 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-pro/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 Invalid SQL statement Error with TIBCO JDBC Driver Under WebLogic

When you set up a TIBCO JDBC driver in the WebLogic console, you need to make sure the Test Table Name property is empty. WebLogic inserts SQL null in this property by default.

If you do not delete this when setting up the driver, you will get errors such as the following:

```
Test "null" set up for pool "JasperserverDataBase" failed with exception:
"java.sql.SQLSyntaxErrorException: [TIBCO][Oracle JDBC Driver][Oracle]ORA-00900: invalid SQL statement
```

If you see this error, you need to edit or recreate the driver for each database that failed using the WebLogic console. See **step 11** in **8.1, “Procedure for Installing the WAR File for WebLogic,” on page 95** as well as **8.3, “Configuring Other Database Connections,” on page 104** for more information.

A.8.7 Performance Issues with Oracle JDBC Queries

Setting the Oracle database localization option defaultNChar to true can substantially impact the performance of JDBC queries. When defaultNChar is set to true, the database will implicitly convert all CHAR data into

NCHAR when you access CHAR columns. If you do not need to support UTF-8 for your Oracle database, you can omit this setting.

The option you need and how to set it depends on your version of Java, your application server, and how it is deployed. For information about changing a JVM option setting for your particular environment, see your application server documentation.

To change this setting on Windows, enter a command such as the following at the command line:

```
set JAVA_OPTS=%JAVA_OPTS% -Doracle.jdbc.defaultNChar=false
```

To change this setting on Linux, enter a command such as the following at the command line:

```
export JAVA_OPTS="$JAVA_OPTS -Doracle.jdbc.defaultNChar=false"
```

A.8.8 Using an Oracle Service Name

If your Oracle database is configured to use a service name instead of an Oracle system identifier (SID), set up the service name by updating your default_master.properties file before using buildomatic:

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

In default_master.properties, uncomment the `serviceName` property and enter your Oracle service name, for example:

```
serviceName=ORCL
```

When you are using an Oracle service name, make sure not to set the SID or dbPort in the default_master.properties file.

A.8.9 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.8.10 Error Running a Report

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

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

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

5. Click the **Test Connection** button in order to validate the connection.

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

A.8.11 Save Error with DB2 Database

When using the DB2 database as your repository database, it is possible to get errors when saving longer strings (such as over 50 characters) to data entry fields in the UI. An example would be saving a resource that has a name over 50 characters long. It is possible to get an error similar to the following:

Expected status code is 200, but was 400. Response body contained:

An unexpected exception has occurred

The problem, in this case, is that DB2 handles UTF-8 characters differently than other Jaspersoft certified databases. The database columns holding these strings need to be made larger. When DB2 is used as the repository database, there is a limitation on the number of characters that can be entered into UI fields.

A.8.12 BeanDefinitionStoreException with DB2 with Vendor's Driver

If you are using the vendor's driver for DB2, you must manually add some properties to default_master.properties. If you do not add these properties, you will get an error such as:

```
[java] Resource name: applicationContext-virtual-data-source.xml
     [java] org.springframework.beans.factory.BeanDefinitionStoreException:
Invalid bean definition with name 'dataSource' defined in file
[/opt/JasperReports-Server-6.0-src/jasperserver/buildomatic/conf_source/iePro/applicationContext-
export-config.xml]:
Could not resolve placeholder 'dbPort' in string value
```

Make sure to add the following properties to your default_master.properties, setting the correct values for your installation:

```
db2.driverType=4
db2.fullyMaterializeLobData=true
db2.fullyMaterializeInputStreams=true
db2.progressiveStreaming=2
db2.progressiveLocators=2
dbPort=50000
js.dbName=JSPRSRVR
sugarcrm.dbName=SUGARCRM
foodmart.dbName=FOODMART
```

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

Now, if you make changes to your <tomcat>/webapps/jasperserver-pro/META-INF/context.xml, Tomcat will not "see" them. Instead, the jasperserver-pro.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-pro/META-INF/context.xml

Remember to delete the jasperserver-pro.xml file:

<tomcat>/conf/Catalina/localhost/jasperserver-pro.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 Installing on tc Server

If you are installing JasperReports Server to an instance of Pivotal tc Server, you can follow the directions for Tomcat installation in **5.2, "Installing the WAR File Using js-install Scripts," on page 36**, with the following modifications:

1. Make the correct application server settings in default_master.properties for tc Server. tc Server is For example:

```
appServerType = tomcat6

...

# Tomcat app server root dir
appServerDir = <tc-home>/<js-instance>
```

In this example, your tc Server instance is installed to <tc-home> and <js-instance> is the subdirectory where you want to install JasperReports Server.

2. Edit the other settings and set up your database.
3. Run the install as described in **"To install the WAR file using js-install scripts" on page 37**
4. After installation, set up your JVM options. For example, for tc Server 2.5 with an Oracle database, you might set the following options:

   On Windows:

   ```
   set JVM_OPTS="-Xms1024m -Xmx2048m -XX:PermSize=32m -XX:MaxPermSize=512m -Xss2m"
   set JVM_OPTS="-XX:+UseConcMarkSweepGC -XX:+CMSClassUnloadingEnabled"
   set JVM_OPTS="-Doracle.jdbc.defaultNChar=true"
   ```

   On Linux:

   ```
   export JVM_OPTS="-Xms1024m -Xmx2048m -XX:PermSize=32m -XX:MaxPermSize=512m -Xss2m"
   export JVM_OPTS="-XX:+UseConcMarkSweepGC -XX:+CMSClassUnloadingEnabled"
   export JVM_OPTS="-Doracle.jdbc.defaultNChar=true"
   ```

   See **6.1, "Setting JVM Options for Application Servers," on page 47** for more information.
5. Start tc Server:

   ```
   <tc-home>/tcruntime-ctl.sh <instance> start
   ```

A.9.7 GlassFish Modifications

A.9.7.1 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.9.7.2 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.9.7.3 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.9.7.4 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.9.7.5 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.9.8 JBoss Modifications

A.9.8.1 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.9.8.2 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.9.8.2.1 Linux

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

A.9.8.2.2 Windows

set JAVA_OPTS=%JAVA_OPTS% -DReservedCodeCacheSize=128m

A.9.8.3 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.9.8.4 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-pro.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.9.8.5 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.9.8.6 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 36**:

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.9.8.7 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.9.8.8 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 47**.

A.9.9 WebSphere Modifications

A.9.9.1 Page Not Found Error on Login

This error is seen during a WebSphere installation when the user attempts to log into JasperReports Server. After typing in a correct user ID and password, the user sees an error page: `Page cannot be found, HTTP 404`

Some WebSphere versions or fix packs have modified code that processes web server filters incorrectly. Components with the `/*` URL pattern get affected by this. JasperReports Server uses the Spring framework for authentication and it is mapped using a filter chain with the `/*` URL pattern. You need to set a special property that WebSphere provides to solve this problem.

To solve the Page Not Found Error on Login:

1. Login into WebSphere Administrative Console.
2. Navigate to **Application Servers > <server> > Web Container Settings > Web Container > Custom Properties**.
3. Create a new property with the following attributes:

   `name: com.ibm.ws.webcontainer.invokefilterscompatibility`

   `value: true`
4. Save the master configuration.
5. **Restart** the WebSphere server.

A.9.10 WebLogic Modifications

A.9.10.1 Schema Validation Error

An error such as the following may be seen on some WebLogic installations:

```
<Critical> <WebLogicServer> <BEA-000362> <Server failed. Reason: [Management:141245]
Schema Validation Error in /u01/app/oracle/WLS/user_projects/domains/x2o_uat_
01/config/config.xml
```

This may be due to the configuration of the `<stuck-thread-max-time>`element in the designated configuration file. In this case, removing `stuck-thread-max-time` may resolve the error.

A.9.11 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.9.12 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.9.12.1 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.9.12.2 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-pro\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 License-related Errors

A.10.1 License Not Found Errors

Normally, the JasperReports Server installer includes an evaluation license file that you replace with a commercial license file, as described in **2.10.3, "Installing a New License File," on page 24**. If JasperReports Server returns an error after you replace the license file, the most likely causes are:

- You did not clear your application server's work directory, as explained in **2.10.3**. Delete the work directory, restart the application server, and try logging into JasperReports Server again.
- The `Djs.license.directory` property in your application server startup environment is incorrectly set: For example, in Windows the correct setting looks like this:

  ```
  -Djs.license.directory=<js-install>
  ```

  In Linux, the correct setting looks like this:

  ```
  -Djs.license.directory=/opt/jasperreports-server-6.0
  ```

  The specified directory must contain the license file, named jasperserver.license. The property is typically set for your application server in the environment startup script. It must contain the location of your license file, which is typically in the <js-install> directory:

Tomcat: <tomcat>/bin/setclasspath.bat/.sh or bin/setenv.bat/.sh

JBoss: <jboss>/bin/run.bat or .sh

A.10.2 Failure to Unlock TIBCO JDBC Driver Error

When you use one of the TIBCO JDBC drivers, the server checks for a valid JasperReports Server license in the correct location. If no such license is found, you will see an error such as the following:

```
Failed to unlock TIBCO JDBC driver. Please check the license file.
```

This is essentially a "License not found" error. Make sure that the jar file for the TIBCO JDBC driver jar is located under the jasperserver-pro/web-inf/lib directory, for example, <tomcat_home>/tomcat/jasperserver-pro/web-inf/lib. The driver jar cannot be placed outside the jasperserver-pro directory, for example, it cannot be in the <tomcat_home>/lib directory. The TIBCO driver jar files are Jaspersoft-specific and will not work with other applications.

If moving the jar file does not resolve this error, use the methods in **A.10.1, "License Not Found Errors," on page 128**.

A.10.3 License Not Found or License Corrupt Error with Tomcat as a Service

If you have an existing Tomcat running as a service under Windows, the installer attempts to make the proper updates so that the JasperReports Server license file is found at application server startup time. If the installer is unsuccessful, be sure that you took the steps described in **2.10.4, "License File for Existing Tomcat as Windows Service," on page 25**.

A.11 Problems Importing and Exporting Data from the Repository

A.11.1 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.11.1.1 Error During Export

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

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

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

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

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

A.11.1.2 Error During Export from Repository on Oracle

Oracle requires a specific JVM property to handle UTF-8 characters properly. When this error happens, the export is empty and an error occurs when attempting to compress the result:

```
ERROR ExporterImpl:129 - java.util.zip.ZipException: ZIP file must have at least one
entry
```

If you have stored your repository database on an Oracle RDBMS, modify the last line of both `<js-install>/buildomatic/js-export.*` files as follows:

From: `java -classpath ...`

To: `java` **`-Doracle.jdbc.defaultNChar=true`** `-classpath ...`

A.12 Webapp Name for Security Property 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-pro/WEB-INF/esapi/Owasp.CsrfGuard.properties`

If you change the name of your webapp from the default of "jasperserver-pro", 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-pro
webAppNamePro = jasperserver-inst2
```

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

Change:

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

To:

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

A.13 Problems with Upgrade

A.13.1 Oracle Error on Upgrade when PL/SQL Not Enabled

If you are upgrading to JasperReports Server version 6.0 or later using the `js-upgrade-samedb.sh/bat` script, you can encounter an error if Oracle's Procedural Language (PL/SQL) is not enabled.

The upgrade script used to upgrade to a 4.7 database from a 4.5 database requires the PL/SQL language to be enabled. The script is located here:

buildomatic/install_resources/sql/oracle/upgrade-oracle-4.5.0-4.7.0-pro.plsql

The error you encounter might look something like the following:

[advanced-sql] PLS-00103: Encountered the symbol "end-of-file"

If your PL/SQL language is not enabled please consult the documentation for your Oracle database to enable PL/SQL.

A.13.2 DB2 Script Error on Upgrade

If you are upgrading from 4.7 to 6.0, and you use the js-upgrade-samedb.bat/sh script, it is possible to get an error related to an inadequate PAGESIZE setting. The recommended minimal PAGESIZE setting for both 4.7 and 5.0 is 16384. If the PAGESIZE setting is less than this you can get an error such as the following:

```
com.ibm.db2.jcc.am.SqlException:
DB2 SQL Error: SQLCODE=-670,SQLSTATE=54010, SQLERRMC=4005;
```

or

```
com.ibm.db2.jcc.am.SqlSyntaxErrorException:
DB2 SQL Error: SQLCODE=-286, SQLSTATE=42727, SQLERRMC=8192;DB2ADMIN,
DRIVER=4.11.77
```

If you get this error, recreate your 4.5 database using a command such as the following:

```
CREATE DATABASE JSPRSRVR USING CODESET UTF-8 TERRITORY US PAGESIZE 16384
```

This may require reloading your database from your database backup and rerunning the upgrade procedure after PAGESIZE has been changed.

A.13.3 Include Audit Events on Upgrade

If you have auditing enabled and you run upgrade using `js-upgrade-newdb.bat/sh`, audit events are not imported by default. To import audit events, you need to run an additional command after the after the `js-upgrade-newdb` script completes. To do this, change to the buildomatic directory:

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

Then run one of the following commands (for example):

`js-import.bat --input-zip=js-my-export-all.zip --include-audit-events` (Windows)

`js-import.sh --input-zip=js-my-export-all.zip --include-audit-events` (Linux)

or

```
ant import -DimportFile=js-my-export-all.zip -DimportArgs="--include-audit-events"
```

These commands reimport all resources from the specified export file. Existing resources will not be overwritten and the audit event will be added.

When using either import utility, the server must be stopped to avoid issues with caches, configuration, and security.

A.13.4 Overlay Upgrade Permissions Error with Bundled Installation

If you are using the overlay upgrade procedure with PostgreSQL as your database and you installed an earlier version of JasperReports Server, you might see an HTTP Status 404 error. The jasperserver-pro.log file will show something like:

```
ERROR ContextLoader,Thread-1:307 - Context initialization failed
org.springframework.beans.factory.BeanCreationException: Error creating bean with
name 'proMaintenanceSchedulerTriggers' defined in ServletContext resource [/WEB-
INF/applicationContext-audit.xml]: Cannot resolve reference to bean
'heartbeatTrigger' while setting constructor argument with key [1];
```

The occurs because if you use the Installer, in the default master properties file we have the user `postgres` and in context.xml we have the user `jasperdb`.

If this occurs, perform the following procedure:

1. Back up your existing master properties.
2. Change the DB user in the master properties to `jasperdb`.
3. Run the overlay upgrade.

A.13.5 Overlay Upgrade Domain Issue with MySQL and MariaDB JDBC Driver

When working with Domains using the MySQL database and when using the 1.1.2 version of the MariaDB JDBC driver (mariadb-java-client-1.1.2.jar), there can be an issue handling boolean values correctly. The fix is to upgrade to a higher version of the MariaDB JDBC driver such as 1.1.6.

Because JasperReports Server 5.5, 5.6, and 6.0 and also the Overlay packages are distributed with the MariaDB 1.1.2 JDBC driver, this error can potentially be seen even without an upgrade operation.

The issue is potentially seen when a boolean filter is created on a domain. The filter ends up not being evaluated as a boolean type , but instead ends up as a numeric type. This is because of a bug in older versions of the MariaDB driver (https://mariadb.atlassian.net/browse/CONJ-72) where a TINYINT(1) is not evaluated to BIT.