JASPERSOFT

JASPERREPORTS SERVER INSTALLATION GUIDE

RELEASE 4.0

http://www.jaspersoft.com

This is version 0111-JSP40-24 of the *JasperReports Server Installation Guide*.

TABLE OF CONTENTS

Chapter 1 Introduction

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

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

- Easily create new reports using an intuitive web-based drag and drop Ad Hoc reporting interface.
- Efficiently and securely manage many reports.
- Interact with reports, including entering parameters and drilling on data.
- Arrange reports and web content to create appealing, data-rich Jaspersoft Dashboards that quickly convey business trends.

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

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

- Our Ultimate Guides document advanced features and configuration. They also include best practice recommendations and numerous examples. The guides are available as downloadable PDFs. Community project users can purchase individual guides or bundled documentation packs from the Jaspersoft online store. Commercial customers can download them freely from the support portal.
- Our free Business Intelligence Tutorials let you learn at your own pace, and cover topics for developers, system administrators, business users, and data integration users. The tutorials are available online from the Professional Services section of our website.
- Our free samples, which are installed with JasperReports, iReport, and JasperReports Server, are documented online. The samples documentation can be found on our community website.

This chapter contains the following sections:

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

1.1 Conventions

For clarity, this document uses the following conventions when referring to file locations, user names, passwords, and other values that are specific to your environment:

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

1.2 Java Version Supported

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

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.
WAR File Binary Distribution Zip	Used for manual installation on Windows, Linux, Mac, and other platforms.

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

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

1.4 Installer Distribution Support

The installers support the following operating systems:

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

1.4.1 Installer Distribution Components

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

Component	Description
JasperReports Server Application	WAR file and configuration support scripts.
Jaspersoft iReport Designer	Latest version of Jaspersoft iReport Designer for Netbeans (optional).
Java 1.6 Runtime	Runs web application container (optional).
MySQL Database	Database server. Can use the bundled version or an existing version.
Apache Tomcat	Web application container: Can use the bundled version or an existing version.
JasperReports Server Documentation	Found in the <js-install>/docs directory.

1.4.2 Installing with Existing Components

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

1.4.3 Running Components as Windows Services

Starting with release 4.0, the Windows installer now installer MySQL and Tomcat as Windows Services. This makes it more convenient for users to manage JasperReports Server under the Windows operating system. JasperReports Server can still be started and stopped from the Windows Startup menu. But, additionally, it can be managed from the Windows Services panel.

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

You can do this in the following location:

Control Panel > System and Security > Administrative Tools > Services

Then, change the Startup Type from "automatic" to "manual"

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

jasperreportsMySQL

jasperreportsTomcat

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

1.5 WAR File Binary Distribution Support

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

Now in version 4.0, there are "auto-install" scripts that will handle the installation tasks by putting your local settings in a single property file. These scripts are named:

js-install.bat

js-install.sh

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

Contents of the WAR file binary distribution:

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

1.6 Release Notes

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

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

1.7 Prerequisites for Installation

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

1.8 System Requirements

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

Operating System	Resource	Footprint	Minimum	Recommended
Windows	Disk	~600MB	10 GB free	40 GB +
	RAM		2 GB	3 GB +
	Processor		1 GHz (single Pentium)	1.5 GHz + (multi-core Pentium)
MAC (OSX)	Disk	~600MB	10 GB free	40 GB +
	RAM		2 GB	3 GB +
	Processor		1 GHz (single Pentium)	1.5 GHz + (multi-core Pentium)
Linux	Disk	~600MB	10 GB free	40 GB +
	RAM		2 GB	3 GB +
	Processor		1 GHz (single Pentium)	1.5 GHz + (multi-core Pentium)
Solaris	Disk	~600 MB	10 GB free	40 GB +
	RAM		2 GB	3 GB +
	Processor		UltraSparc II	
AIX	Disk	~600 MB	10 GB free	40 GB +
	RAM		2 GB	3 GB +
HP-UX	Disk	~600 MB	10 GB free	40 GB +
	RAM		2 GB	3 GB +

1.9 Support for Internationalization

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

CHAPTER 2 INSTALLING JASPERREPORTS SERVER

This chapter contains the following sections:

- **Installation Steps**
- **Post-Installation Steps**

2.1 Installation Steps

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

When you run the installer against an existing database instance, the database must be running at install time.

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

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

```
jasperreports-server-4.0-windows-installer.exe
```

In Linux, the installer is a .bin file; you can run it from the command line or from a graphical environment. To start the installer from the command line, login with an account that has administrative privileges and open a bash shell. At the command line, enter the name of the installer file. For example:

```
./jasperreports-server-4.0-linux-installer.bin
```

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

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

```
jasperreports-server-4.0-osx-installer.app
```

2.1.1 Welcome

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

2.1.2 Accepting the License Agreement

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

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

2.1.3 Choosing an Installation Directory

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

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

Windows:	C:\Program Files\jasperreports-server-4.0
Linux:	<USER_HOME>/jasperreports-server-4.0
Mac OSX	/Applications/jasperreports-server-4.0

2.1.4 Selecting Tomcat Configuration

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

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

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

2.1.5 Selecting MySQL Configuration

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

The first option is to choose a bundled MySQL. If you choose this option, the installer puts an instance of MySQL 5 onto your system. Click **Next**. The default MySQL port 3306 will be used. The installer will also create a MySQL database user with administrator privileges and credentials of jasperdb/password. If the installer finds that port 3306 is already in use, you are prompted to pick an alternate port. In this case, pick an alternative port value and click **Next**.

Values to be entered or set to defaults for the Bundled MySQL configuration:

Parameter	Default Value and Description
Port	3306 - User must choose an alternate port if 3306 is in use.
Database User Name	Hard coded default: jasperdb - The installer creates this user which is used to connect to the JasperReports Server database
Database User Password	Hard coded default: password - The installer uses this password for the jasperdb user.

To improve system security, Jaspersoft recommends that you change all default passwords as soon as possible.

The second option is to choose an existing MySQL. If you already have an instance of MySQL running on your system, then you can choose this option. Choose the existing MySQL option and click **Next**. You are prompted for the location of MySQL, and the port to use. Note that the MySQL instance must reside on your local machine (i.e. localhost or 172.0.0.1). Enter the correct location for MySQL or click **Browse** to locate and select another location. Click **Next**. You are prompted for the root database account password of the MySQL root administrative user. The dbuser administrator account “root” is used by default. Enter the root dbuser password and click **Enter**.

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

Parameter	Default Value and Description
Binary Directory	The directory where the mysql and mysqladmin binaries are located.
Port	The port number that MySQL uses (default is 3306).
IP or Host Name	The value is hard coded to 172.0.0.1. Note that your existing MySQL instance must reside on the local machine.
MySQL Root Password	Password of the database administrative dbuser: root. The installer cannot handle special characters at the end of a password string. Incompatible characters include: & ; $
Defaults Used	**Hardcoded Default Values Used or Created**
MySQL Root User Name	The default administrative dbuser is “root”
Database User Name	jasperdb - The installer creates this dbuser which is used to connect to jasperserver database.
Database User Password	password - The installer creates this password for the jasperdb dbuser.

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

2.1.6 Installing Sample Data

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

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

2.1.7 Installing Jaspersoft iReport Designer

Jaspersoft iReport Designer (hereafter called “iReport”) is the leading GUI-based JasperReports Library creation tool. It has the capability of communicating directly with a JasperReports Server instance and can thus retrieve existing JasperReports Library from a JasperReports Server instance for editing, uploading or execution.

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

If you would like to install iReport click **Yes**.

2.1.8 Ready to Install

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

2.1.9 Installation Complete Screen

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

- **View Release Notes** - If you choose to view the Release Notes, they are displayed in a new window. If you are running from the command line, you can page through the Release Notes by pressing the Enter key.
- **Launch JasperReports Server Now** - If you choose to launch JasperReports Server from the installer, the installer exits and the application server starts. There is a 25 second or so pause to allow the server to start up. When this pause is complete, the login page appears in your system default Browser. For information on logging in, see section **3.4, "Logging Into JasperReports Server," on page 24**.

> Starting from the installer is dependent on your Tomcat and MySQL configuration choices. The start/stop scripts only control the bundled applications that you chose to be installed. For more information, see **Chapter 3, "Starting and Stopping JasperReports Server," on page 23**.
>
> Also, if you chose to view the Release Notes, JasperReports Server will not startup until you close the Release Notes.

- **Opt-in for JasperServer Heartbeat** - The JasperReports Server heartbeat will help Jaspersoft create better products by improving our understanding of customer installation environments. When the heartbeat is enabled, the server sends anonymous system and version information to Jaspersoft via https. For more information, see section **5.11.1, "JasperReports Server Heartbeat," on page 36**.

You can later enable or disable the heartbeat by modifying the jasperserver-pro/WEB-INF/applicationContext-logging.xml file. For additional information on enabling and disabling the heartbeat component refer to the *JasperReports Server Administrator Guide*.

2.1.10 Logging into JasperReports Server

You should now be ready to log into the server. For information on default login credentials, go to section **3.4, "Logging Into JasperReports Server," on page 24**.

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

2.2 Post-Installation Steps

2.2.1 Updates Made by the Installer During Installation

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

Updates made to the application server

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

File or Directory	Updates
Windows: bin/setclasspath.bat Linux: bin/setclasspath.sh	Modifies JAVA_OPTS to add -Djs.license.directory
Windows: bin/setenv.bat Linux: bin/setevn.sh	This file gets newly created. Sets increased Java memory allocation values to JAVA_OPTS. For additional settings, refer to section **5.9, “Setting Java JVM Options,” on page 35**.
Tomcat 5: common/lib Tomcat 6: lib	Adds MySQL JDBC driver.

Updates made to the MySQL database

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

MySQL Updates	Description
Database `jasperserver` created	This is the JasperReports Server repository database. This database holds all of system information, such as users, roles, datasources, and report definitions.
Database user `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.

2.2.2 Installer Output Log File Location

The installer creates a log during installation that records information as the installation progresses. If you encounter any problems when you install JasperReports Server, it can be helpful to look at the installer log for any potential errors. You can find the installer log in the following locations:

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

2.2.3 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. Once 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. However, note that 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, Jaspersoft recommends that you clear the work directory before upgrading your license. For instance, in Tomcat, you would do the following:

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

By default, the license is found in the <js-install> directory. However, it can be located anywhere, so long as the -Djs.license.directory Java Environment Variable points to its location (defined in the Tomcat startup scripts). Note that the name of the file should be 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 is granting access properly. For information about license errors, see the troubleshooting section **A.3, “License Not Found Error,” on page 118**.

For additional license configuration options, refer to sections **5.8, "Setting Up the JasperReports Server License," on page 35** and **6.2, "Setting Up the JasperReports Server License," on page 46**.

2.2.4 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 (section **2.1.3**), you will probably have to manually configure Tomcat with the file location.

Follow the steps below to examine and/or 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 menu at **Start > All Programs > Apache Tomcat > Tomcat Configuration**.
2. Click **Configure** and select the **Java** tab.
3. At the bottom of the **Java Options** field, enter the following:

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

   For example:

   ```
   -Djs.license.directory="C:\Program Files\jasperreports-server-4.0"
   ```
4. Stop and restart the application server.

You should now be able to run JasperReports Server.

2.2.5 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.

To check your Java JVM settings refer to section **5.9, "Setting Java JVM Options," on page 35**.

CHAPTER 3 STARTING AND STOPPING JASPERREPORTS SERVER

This chapter contains the following sections:

3.1 Using the Start/Stop Menu With Windows

Before JasperReports Server is started, the database it depends on must be running. Then, JasperReports Server is started by starting the application server where JasperReports Server is deployed. If you chose to install a bundled Tomcat and a bundled MySQL, then the start/stop scripts will allow you to start both applications with a single script.

To start or stop JasperReports Server, do the following:

- From the Windows Start menu:

  Click **Start > All Programs > JasperReports Server 4.0 > JasperReports Server Service > Start Service**.

  Click **Start > All Programs > JasperReports Server 4.0 > JasperReports Server Service > Stop Service**.
- Additionally, JasperReports Server can be managed from the Services Admin Tool

  Go to the Windows Services Tool Panel

  Look for: jasperreportsTomcat

  Look for: jasperreportsMySQL

  These can be individually started and stopped from this location.
- When JasperReports Server is running, in the Windows Task Manager you can see the following processes:

  mysqld.exe

  tomcat6.exe

3.2 Using Start/Stop Scripts Without Bundled Installation with Windows

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

For example, if you have an existing Tomcat and installed the bundled MySQL, the scripts and menus specified in the previous section would only start and stop the MySQL application. For the existing Tomcat, you would use the existing start and stop scripts provided by the Tomcat application.

3.3 Using Start/Stop Apps With Mac OSX

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

/Applications/jasperreports-server-4.0

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

jasperServerStart.app

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

jasperServerStop.app

Support for the Dock:

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

jasperServerStart.app

jasperServerStop.app

jasperServerLogin.app

3.4 Logging Into JasperReports Server

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

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

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

Windows

Under Windows, you can launch the login page from the desktop of its host by clicking **Start > All Programs > JasperReports Server 4.0 > JasperReports Server Login**.

Mac OSX

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

<js-install>/jasperServerLogin

For example: /Applications/jasperreports-server-4.0/jasperServerLogin

Support for the Dock:

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

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

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. Delete these users when you stop using them:

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

3.5 Starting the Included Jaspersoft iReport Designer

If you chose to install iReport as part of the JasperReports Server installation, you may start iReport from the Windows Start menu. To do this, click **Start > All Programs > JasperReports Server 4.0 > Start iReport Designer**.

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

To start on Mac OSX:

Double-click iReport-mac.dmg

Double-click Jaspersoft iReport Professional Designer

3.6 JasperReports Server Log Files

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

Tomcat:	<tomcat>/webapps/jasperserver-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.

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

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

Chapter 4 Uninstalling JasperReports Server

4.1 Windows

Under Windows, click **Start > All Programs > JasperReports Server 4.0 > Uninstall JasperReports Server** to uninstall JasperReports Server.

Note for Windows XP:

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

4.2 Linux

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

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

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

4.3 Mac OSX

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

Navigate to the <js-install> folder

For example: /Applications/jasperreports-server-4.0

Click the uninstall app to launch the uninstaller

4.4 Uninstall Survey

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

Chapter 5 Installing the WAR File Distribution

In addition to the installer binaries, the JasperReports Server application is distributed as a stand-alone WAR file distribution. This distribution is packaged as a ZIP file. Customers who do not wish to use the installer or who have target configurations other than those supported by the installer should use the WAR file distribution.

This chapter contains the following sections:

5.1 Applications Supported by the WAR File Distribution

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

Database	App Server	Instructions Located In
MySQL PostgreSQL Oracle SQL Server DB2	Apache Tomcat JBoss GlassFish	This chapter
	WebSphere	**Chapter 7, "Installing the WAR File for WebSphere," on page 61**
	WebLogic	**Chapter 8, "Installing the WAR File for WebLogic," on page 75**

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

5.2 Obtaining the WAR File Distribution Zip

The WAR file distribution comes in a file named jasperreports-server-4.0-bin.zip in the compressed ZIP format. To download the WAR file distribution, contact Jaspersoft technical support or your sales representative.

5.3 Unpacking the WAR File Distribution Zip

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

Choose a top level directory location to unpack the ZIP file to. The ZIP file creates the directory jasperreports-server-4.0-bin.

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

Operating System	Example Location	Referenced As
Windows	C:\Program Files\jasperreports-server-4.0-bin	<js-install>
Windows	C:\jasperreports-server-4.0-bin	<js-install>
Linux	/home/<user>/jasperreports-server-4.0-bin	<js-install>

5.4 Introduction to Buildomatic Scripts

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

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

For JasperReports Server release 4.0, there are now "auto-install" shell scripts included that automate the installation steps. These scripts are located here:

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

5.5 Pre-Installation Steps

5.5.1 Checking Your Java Installation

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

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

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

To check your Java version from the command line run:

```
java -version
```

5.5.2 About Bundled Apache Ant

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

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

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

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

On Linux and Solaris, the js-ant commands may not be compatible with all shells. If you have errors, use the `bash` shell explicitly. For more information, see section **A.16, "Troubleshooting on Solaris," on page 122**.

5.5.3 Checking Your Application Server

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

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

For Tomcat and JBoss:

When running the buildomatic install scripts, Tomcat and JBoss should be stopped.

For Glassfish:

When running the buildomatic install scripts, Glassfish should be running. To start the GlassFish application server, run a command similar to the following:

```
asadmin start-domain domain1
```

5.5.4 Checking Your Database Server

To run JasperReports Server you will need to have a database available. The buildomatic scripts support automatic installation to the MySQL, PostgreSQL, Oracle, SQL Server, and DB2 databases. When running the buildomatic scripts or performing any manual installation, your database server should be running.

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

User	`jasperserver`
Password	`password`

The password is case-sensitive.

When you complete the evaluation or testing of your JasperReports Server instance, you should change the default password. Leaving it weakens the security of your installation.

5.6 Configuring the Buildomatic Scripts

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

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

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

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

5.6.1 Creating your Default Master Properties File

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

1. Copy the file for your database:

Database	Master Properties File
MySQL	<js-install>/buildomatic/sample_conf/mysql_master.properties
PostgreSQL	<js-install>/buildomatic/sample_conf/postgresql_master.properties
Oracle	<js-install>/buildomatic/sample_conf/oracle_master.properties
SQL Server	<js-install>/buildomatic/sample_conf/sqlserver_master.properties
DB2	<js-install>/buildomatic/sample_conf/db2_master.properties

2. And rename to:

   <js-install>/buildomatic/default_master.properties

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

Database	Sample Property Values

Oracle	`appServerType=tomcat6 [tomcat5, jboss, glassfish, skipAppServerCheck[*]]`
	`appServerDir=c:\\apache-tomcat-6.0.26 [for example[†]]`
	`sysUsername=sys as sysdba`
	`sysPassword=password`
	`dbUsername=jasperserver`
	`dbPassword=password`
	`dbHost=hostname`

Database	Sample Property Values

SQL Server	`appServerType=tomcat6 [tomcat5, jboss, glassfish, skipAppServerCheck*]`
	`appServerDir=c:\\apache-tomcat-6.0.26 [for example†]`
	`dbUsername=sa`
	`dbPassword=sa`
	`dbHost=localhost`
	If your app server runs on Java 1.5, follow the instructions in section **6.3.5, "Buildomatic: Change your Deployed JDBC Driver," on page 50** after modifying these properties.

DB2	`appServerType=tomcat6 [tomcat5, jboss, glassfish, skipAppServerCheck*]`
	`appServerDir=c:\\apache-tomcat-6.0.26 [for example†]`
	`dbUsername=db2admin`
	`dbPassword=password`
	`dbHost=localhost`
	For DB2 8.x, follow the instructions in section **6.3.5, "Buildomatic: Change your Deployed JDBC Driver," on page 50** after modifying these properties.

MySQL	`appServerType=tomcat6 [tomcat5, jboss, glassfish, skipAppServerCheck*]`
	`appServerDir=c:\\apache-tomcat-6.0.26 [for example†]`
	`dbUsername=root`
	`dbPassword=password`
	`dbHost=localhost`

PostgreSQL	`appServerType=tomcat6 [tomcat5, jboss, glassfish, skipAppServerCheck*]`
	`appServerDir=c:\\apache-tomcat-6.0.26 [for example†]`
	`dbUsername=postgres`
	`dbPassword=postgres`
	`dbHost=localhost`

* When the property `appServerType` is set to `skipAppServerCheck`, buildomatic skips any application server validation. Use this setting when installing JasperReports Server with WebSphere and WebLogic, because they are not directly supported by Buildomatic.

† Backslashes in paths must be doubled in properties files, for example `appServerDir=c:\\tomcat\\test`. For WebLogic 10, make sure there are no spaces in the path given in `appServerDir`.

5.7 Installing JasperReports Server

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

JasperReports Server 4.0 comes with new Windows batch and Linux shell scripts that automatically handle the full installation of JasperReports Server. These are referred to as the "auto-install" scripts.

Deploying to Glassfish. The Glassfish application server must be running before executing any of the install commands. For more information, see section **5.15, "Deploying to GlassFish," on page 40**.

Deploying to the DB2 Database. If you are deploying to the DB2 database, for more information see section **5.16, "Manual Steps for DB2," on page 40**.

If you would like to run a pre-install validation test, you can run a command such as: "js-install.bat test". For more information, see section **5.12.3, "Error Running Auto-Install Scripts (js-install.bat/sh)," on page 37**.

5.7.1 Windows Auto-Install

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

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

You may now proceed to the sections below on the JasperReports Server License, the Java JVM Options, and Starting JasperReports Server (**5.8, "Setting Up the JasperReports Server License," on page 35**). For troubleshooting see section **5.12, "Troubleshooting your JasperReports Server Configuration," on page 37**.

5.7.2 Linux Auto-Install

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

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

You may now proceed to the sections below on the JasperReports Server License, the Java JVM Options, and Starting JasperReports Server (**5.8, "Setting Up the JasperReports Server License," on page 35**). For troubleshooting see section **5.12, "Troubleshooting your JasperReports Server Configuration," on page 37**.

5.7.3 Auto-Install with Minimal Data

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

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

You may now proceed to the sections below on the JasperReports Server License, the Java JVM Options, and Starting JasperReports Server (**5.8, "Setting Up the JasperReports Server License," on page 35**).

5.7.4 Errors

If you encounter errors during the auto-install script execution, you should refer to the Troubleshooting section **5.12, "Troubleshooting your JasperReports Server Configuration," on page 37**.

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

5.8 Setting Up the JasperReports Server License

JasperReports Server Professional or Enterprise requires a license to run. The JasperReports Server WAR file distribution comes with an evaluation license that is valid only for a number of days. Contact Technical Support or your sales representative to get your commercial license.

The JasperReports Server license file is included in the root of the <js-install> directory and is named jasperserver.license. It 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.

5.8.1 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 that is running the application server. So, all that is required is to determine the location of the application user home directory and to copy the Jasperserver license file to this location.

Copy the file:

<js-install>/jasperserver.license

To:

Windows:	C:\Documents and Settings\<user>\jasperserver.license
Windows 7:	C:\Users\<user>\jasperserver.license
Windows 7:	C:\jasperserver.license (if Win 7 Existing Tomcat as a Windows Service)
Linux:	/home/<user>/jasperserver.license

5.8.2 Alternative JasperReports Server License Set Up

If you prefer your license to be located in a directory of your own choosing, you can modify your application server startup script and set a JAVA_OPT value to explicitly point to that directory.

For this configuration, refer to one of the following sections: **6.2.2, “Alternate License Setup for Tomcat,” on page 47** or **6.2.4, “Alternate License Location for Existing Tomcat as a Windows Service,” on page 48**.

5.9 Setting Java JVM Options

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

For Tomcat and JBoss these values can be directly set by editing the shell scripts which start the application server. See section **6.1.1, “Tomcat and JBoss JVM Options,” on page 43** for detailed steps.

For GlassFish, you can set the JVM options using the asadmin utility or by editing the domain.xml config directly. See section **6.1.4, “GlassFish JVM Options,” on page 45** for detailed steps.

5.10 Starting JasperReports Server

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

Tomcat:	<tomcat>/bin/startup.bat (Windows) *or* <tomcat>/bin/startup.sh (Linux)
JBoss:	<jboss>/bin/run.bat (Windows) *or* <jboss>/bin/run.sh (Linux)
GlassFish:	asadmin start-domain domain1

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

To view the JasperReports Server application logs, see section **3.6, "JasperReports Server Log Files," on page 25**.

5.11 Logging into JasperReports Server

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

Login by going to the following URL:

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

Example:

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

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

Use the following credentials to log into the system:

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

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

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

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

5.11.1 JasperReports Server Heartbeat

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

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

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

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

You can also manually enable or disable the heartbeat by modifying the jasperserver-pro/WEB-INF/applicationContext-logging.xml file. To disable, set the enabled property to false as shown below:

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

For additional information on enabling and disabling the heartbeat component refer to the *JasperReports Server Administrator Guide*.

5.12 Troubleshooting your JasperReports Server Configuration

5.12.1 JasperReports Server Startup Problems

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

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

For more information on resolving these types of errors, refer to troubleshooting section **A.2, "Database Connectivity Errors," on page 116**.

5.12.2 Error Running a Report

If you have trouble running reports in your new JasperReports Server instance, refer to troubleshooting section **A.5, "Error Running a Report," on page 118**.

5.12.3 Error Running Auto-Install Scripts (js-install.bat/sh)

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

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

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

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

Additionally, the auto-install scripts can be run in a "test" mode. This will help debug issues such as a mis-typed DB password. If there is a typing change you need to make to your default_master.properties file, you can make the edit and re-run the js-install script. The js-install script will always use the most current values found in the default_master.properties file.

Run auto-install Script in Validation Test Mode

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

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

In test mode, the js-install scripts will check your settings to see if any setting can be determined to be incorrect. The application server location will be validated and the ability to connect to the specified database will be validated. Your system will not be altered when executing in test mode.

5.13 Running the Import and Export Utilities

The buildomatic scripts automatically configure the database information needed by the buildomatic import and export functionality. This functionality is invoked via ant targets used by buildomatic and located in the following directory:

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

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

For complete information on the standard import-export options refer to the *JasperReports Server Administrator Guide*.

5.13.1 Running Export from Buildomatic

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

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

Linux: `./js-ant export -DexportFile=<filename> -DexportArgs=\"<export-options>\"`

The export file format can be a ZIP file or it can be a set of files under a new directory name. If you specify the `.zip` extension for your output filename then a ZIP archive will automatically be created. Otherwise, a directory with files and sub-directories will be created as a non-compressed set of files. The `exportArgs` argument requires double quotes (`"`) and can contain more than one export option, as shown in the Windows examples below:

```
js-ant export-help-pro

js-ant export -DexportFile=my-domains.zip
  -DexportArgs="--uris /organizations/organization_1/Domains"


js-ant export -DexportFile=my-reports-and-users.zip
  -DexportArgs="--uris /organizations/organization_1/reports
  --users jasperadmin|organization_1,joeuser|organization_1"

js-ant export -DexportFile=my-datasources
  -DexportArgs="--uris /organizations/organization_1/datasources --roles ROLE_USER"

js-ant export -DexportFile=js-everything.zip -DexportArgs="--everything"
```

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

```
./js-ant export-help-pro

./js-ant export -DexportFile=my-reports-and-users.zip
  -DexportArgs=\"--uris /organizations/organization_1/reports
  --users 'jasperadmin\|organization_1,joeuser\|organization_1'\"
```

5.13.2 Running Import from Buildomatic

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

Windows: `js-ant import -DimportFile=<filename> [-DimportArgs="<import-options>"]`

Linux: `./js-ant import -DimportFile=<filename> [-DimportArgs=\"<import-options>\"]`

The imported file is handled as a ZIP archive if its name ends in `.zip`, otherwise it will be handled as a directory. The `importArgs` argument is optional, it can contain more than one import option. On Linux, all double quotes (`"`) must be escaped with a backslash (`\`).

The following examples on Windows are typical import commands:

```
js-ant import-help-pro

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

js-ant import -DimportFile=my-datasources -DimportArgs="--update"
```

The following examples on Linux are typical import commands:

```
./js-ant import-help-pro

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

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

5.13.3 Running the Import Export Shell Scripts

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

You can find them here:

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

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

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

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

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

For information on configuring and running the import-export shell scripts, see **Chapter 14, “Configuring the Import-Export Utilities,” on page 109**.

5.14 Pre-Test Validation mode of Auto-Install Scripts

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

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

5.14.1 Additional Command for Buildomatic Settings

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

<js-install>/buildomatic/build_conf/default

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

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

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

The first target will clear the configuration template files found in buildomatic/build_conf/default directory. The second will re-build the configuration settings.

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

5.15 Deploying to GlassFish

The GlassFish application server must first be running as described in section **5.5.3, "Checking Your Application Server," on page 31**. Check that GlassFish is running then run:

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

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

- JVM options will be set as described in section **5.9, "Setting Java JVM Options," on page 35**.
- The GlassFish application server will be stopped. You will restart the server in a subsequent step.

5.16 Manual Steps for DB2

The buildomatic scripts are not able to automatically connect to a remote DB2 database and carry out Admin operations. For this reason, it is necessary to manually create the databases.

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

Use commands similar to the ones below:

Commands	Description
`db2 create database jsprsrvr using codeset utf-8 territory us`	Creates the JasperReports Server database with required settings.
`db2 create database sugarcrm` db2 create database foodmart	(Optional) Creates the sample databases.

Further considerations:

- If you are using DB2 8.1, you should 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

Now, you may continue with the normal buildomatic steps in section **5.7, "Installing JasperReports Server," on page 33**.

5.16.1 Creating Databases Manually

If you encounter an error when running one of the create db targets (create-sugarcrm-db, create-foodmart-db, or create-js-db) you may create the JasperReports Server database manually using the database administration tool for your particular database type. After creating the database, you can continue with the remaining buildomatic steps. Additionally, keep in mind that the sugarcrm and foodmart databases are optional, sample databases.

For procedures on MySQL, Oracle, and DB2, see section **6.4, "Additional Notes on Databases," on page 51**.

5.17 Manual Buildomatic Install Steps

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

Older manual buildomatic install target steps:

Commands	Description
`cd <js-install>/buildomatic`	Go to the buildomatic scripts directory

`js-ant create-js-db`
`js-ant create-sugarcrm-db`
`js-ant create-foodmart-db`
	Creates the JasperReports Server and sample databases.
	For DB2, see section **5.8, "Setting Up the JasperReports Server License," on page 35**.

`js-ant load-sugarcrm-db`
`js-ant load-foodmart-db`
`js-ant update-foodmart-db`
	(Optional) Loads sample data into the sample databases.

`js-ant init-js-db-pro`
`js-ant import-minimal-pro`
	Initializes database, loads core application 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 or JBoss. For GlassFish, see **Setting Up the JasperReports Server License** below. For WebLogic and WebSphere, skip this step and return to the instructions in the corresponding chapter.

On Linux and Solaris, the js-ant commands may not be compatible with all shells. If you have errors, use the `bash` shell explicitly. For more information, see section **A.16, "Troubleshooting on Solaris," on page 122**.

CHAPTER 6 ADDITIONAL INSTALLATION INFORMATION

This chapter contains the following sections:

6.1 Setting JVM Options for Application Servers

JasperReports Server runs better with certain Java options for the JVM in which its application server is running. The options you need and how you set them depends on your version of Java, your application server, and how it is deployed. In addition, there is a setting to support localization when running with an Oracle database.

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

6.1.1 Tomcat and JBoss JVM Options

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

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

JVM Options on Windows, continued	
Options for Java 1.5 and Java 1.6	set JAVA_OPTS=%JAVA_OPTS% -Xms512m -Xmx1024m -XX:PermSize=32m -XX:MaxPermSize=128m -Xss2m -XX:+UseConcMarkSweepGC -XX:+CMSClassUnloadingEnabled -XX:+CMSPermGenSweepingEnabled
For Oracle	set JAVA_OPTS=%JAVA_OPTS% -Doracle.jdbc.defaultNChar=true
Additional options for Java 1.6 or JBoss 4.2	set JAVA_OPTS=%JAVA_OPTS% -Djavax.xml.soap.MessageFactory=org.apache.axis.soap.MessageFactoryImpl -Djavax.xml.soap.SOAPConnectionFactory=org.apache.axis.soap.SOAPConnectionFactoryImpl -Djavax.xml.soap.SOAPFactory=org.apache.axis.soap.SOAPFactoryImpl -Djavax.xml.transform.TransformerFactory=org.apache.xalan.processor.TransformerFactoryImpl

JVM Options on Linux	
Tomcat file	<tomcat>/bin/setclasspath.sh *or* <tomcat>/bin/setenv.sh
JBoss file	<jboss>/bin/run.sh
Options for Java 1.5 and Java 1.6	export JAVA_OPTS="$JAVA_OPTS -Xms512m -Xmx1024m -XX:PermSize=32m -XX:MaxPermSize=128m -Xss2m -XX:+UseConcMarkSweepGC -XX:+CMSClassUnloadingEnabled -XX:+CMSPermGenSweepingEnabled"
For Oracle	export JAVA_OPTS="$JAVA_OPTS -Doracle.jdbc.defaultNChar=true"
Additional options for Java 1.6 or JBoss 4.2	export JAVA_OPTS="$JAVA_OPTS -Djavax.xml.soap.MessageFactory=org.apache.axis.soap.MessageFactoryImpl -Djavax.xml.soap.SOAPConnectionFactory=org.apache.axis.soap.SOAPConnectionFactoryImpl -Djavax.xml.soap.SOAPFactory=org.apache.axis.soap.SOAPFactoryImpl -Djavax.xml.transform.TransformerFactory=org.apache.xalan.processor.TransformerFactoryImpl"

Add your JAVA_OPTS setting directly below the following lines:

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

6.1.2 Bundled Tomcat as a Windows Service JVM Options

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

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

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

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

Edit service.bat

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

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

Update the line above to increase the Java heap:

```
-Xms1024M;-Xmx2048M
```

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

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

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

6.1.3 Existing Tomcat as a Windows Service JVM Options

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

1. Launch the Tomcat configuration application from the Windows Start menu:

   Start > Programs > Apache Tomcat > Configure Tomcat
2. In the Apache Tomcat Properties dialog, click the **Java** tab.
3. In the Java Options field, add your JAVA_OPTS values according to the table above.

   Enter only the options preceded by `-X` or `-D`, not `set JAVA_OPTS=%JAVA_OPTS%`.

   Enter only one java option setting per line
4. For instance, add like shown below:

   ```
   -Xms512m
   -Xmx1024m
   -XX:PermSize=32m
   -XX:MaxPermSize=128m
   -Xss2m
   ```
5. Click Apply, then click OK.

6.1.4 GlassFish JVM Options

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

6.1.4.1 Setting GlassFish JVM Options with asadmin Command

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

```
asadmin create-jvm-options -Xms512m:-Xmx1024m:-XX\:PermSize=32m:
  -XX\:MaxPermSize=128m:-Xss2m:-XX\:+UseConcMarkSweepGC:
  -XX\:+CMSClassUnloadingEnabled:-XX\:+CMSPermGenSweepingEnabled:
  -Djavax.xml.soap.MessageFactory=org.apache.axis.soap.MessageFactoryImpl:
  -Djavax.xml.soap.SOAPConnectionFactory=org.apache.axis.soap.SOAPConnectionFactoryImpl:
  -Djavax.xml.soap.SOAPFactory=org.apache.axis.soap.SOAPFactoryImpl:
  -Doracle.jdbc.defaultNChar=true
```

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

Now, restart the application server with the following commands:

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

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

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

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

6.1.4.2 Setting GlassFish JVM Options by Editing domain.xml

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

```
<jvm-options>-Xms512m -Xmx1024m -XX:PermSize=32m -XX:MaxPermSize=128m -Xss2
  -XX:+UseConcMarkSweepGC -XX:+CMSClassUnloadingEnabled
  -Djavax.xml.soap.MessageFactory=org.apache.axis.soap.MessageFactoryImpl
  -Djavax.xml.soap.SOAPConnectionFactory=org.apache.axis.soap.SOAPConnectionFactoryImpl
  -Djavax.xml.soap.SOAPFactory=org.apache.axis.soap.SOAPFactoryImpl
  -Doracle.jdbc.defaultNChar=true
</jvm-options>
```

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

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

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

6.2 Setting Up the JasperReports Server License

JasperReports Server Professional or Enterprise requires a license to run. The JasperReports Server WAR file distribution comes with an evaluation license that is valid for a number of days. Please contact Technical Support or your sales representative to get your commercial license.

The JasperReports Server license file is named jasperserver.license and is found in the <js-install> directory. See **5.8, "Setting Up the JasperReports Server License," on page 35** for more information about the license.

Under Windows, for the new JasperReports Server 4.0 release, both Tomcat and MySQL "bundled" components are installed as Windows Services by default. You can see the status of these services in the Administrative Services panel.

Under Windows, these components are still executed from the <js-install> directory location. For instance, the bundled Apache Tomcat would be installed to this location by default:

C:\Program Files\jasperreports-server-4.0\apache-tomcat (Windows XP)

C:\Program Files (x86)\jasperreports-server-4.0\apache-tomcat (Windows 7 64bit)

6.2.1 License Configuration for Application Servers

JasperReports Server will recognize the license file when it is located in the home directory of the user that runs your application server.

Copy: <js-install>/jasperserver.license

To: C:\Documents and Settings\<user>\jasperserver.license (Windows XP)

C:\Users\<user>\jasperserver.license (Windows 7)

C:\jasperserver.license (Window 7, Existing Tomcat as Windows Service. User SYSTEM.)

/home/<user>/jasperserver.license (Linux)

/Users/<user>/jasperserver.license (Mac OSX)

6.2.2 Alternate License Setup for Tomcat

If your license will not be located in the home directory of the application server user, then 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:

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"
   ```

6.2.3 Alternate License Setup for Bundled Tomcat as a Windows Service

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

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

Here are the steps to change the JasperReports Server bundled Tomcat to point to a different jasperserver.license location:

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

Edit service.bat

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

```
"%EXECUTABLE%" //US//%SERVICE_NAME% ++JvmOptions

"-Djs.license.directory=C:\Program Files\jasperreports-server-4.0;

-XX:PermSize=32m;-XX:MaxPermSize=128m;-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 will need to re-install the service. From a Windows cmd shell:

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

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

6.2.4 Alternate License Location for Existing Tomcat as a Windows Service

Windows XP:

If you have an existing Tomcat as a Windows Service under Windows XP, you may be required to put your license in a special location if you want it to be picked up by default. This has been reported by some customers. The location would be the following:

C:\Documents and Settings\LocalService\jasperserver.license

This location can be unusual because the LocalService directory is not displayed when you click on the Documents and Settings directory in Windows Explorer. If you go to the Explorer Address Bar and type `LocalService`, it will be displayed. You can copy the jasperserver.license file to this location if your Tomcat as a Windows Service is unable to find the license in the Documents and Settings/<user> location described in the sections above.

Windows 7:

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

C:\jasperserver.license

6.2.5 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:

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 Additional Buildomatic Configuration Information

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

6.3.1 Buildomatic: Generated Property Files

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

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

<js-install>/buildomatic/build_conf/default

Here are some of the key configuration files:

js.jdbc.properties

js.quartz.properties

js-glassfish-ds.xml

js-jboss-ds.xml

maven_settings.xml - (used for source code build)

More generated property files:

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

In this directory you will find config files such as:

META-INF/context.xml

WEB-INF/hibernate.properties

WEB-INF/js.quartz.properties

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

6.3.2 Buildomatic: SQL Scripts Location

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

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

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

<js-install>/buildomatic/install_resources/sql/mysql/js-pro-create.ddl

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

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

<js-install>/buildomatic/install_resources/sql/mysql/sugarcrm.zip

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

6.3.3 Buildomatic: Database Creation Statements Location

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

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

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

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

6.3.4 Buildomatic: JDBC Driver Locations

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

Database	Buildomatic JDBC Driver Location
DB2	<js-install>/buildomatic/conf_source/db/db2/jdbc/db2jcc-9.5.jar (for DB2 9.x) <js-install>/buildomatic/conf_source/db/db2/jdbc/alt-drivers/db2jcc-8.2.jar (for DB2 8.x)
MySQL	<js-install>/buildomatic/conf_source/db/mysql/jdbc/mysql-connector-java-5.1.10.jar
Oracle	<js-install>/buildomatic/conf_source/db/oracle/jdbc/ojdbc5.11.2.0.jar <js-install>/buildomatic/conf_source/db/oracle/jdbc/ojdbc14.10.2.0.jar
PostgreSQL	<js-install>/buildomatic/conf_source/db/postgresql/jdbc/postgresql-8.1-407.jdbc3.jar
SQL Server	<js-install>/buildomatic/conf_source/db/sqlserver/jdbc/sqljdbc-1.6.jar (for Java 6) <js-install>/buildomatic/conf_source/db/sqlserver/jdbc/sqljdbc-1.5.jar (for Java 5)
Vertica (as a data source only)	<js-install>/buildomatic/conf_source/db/vertica/jdbc/vertica_3.5_jdk_5.jar

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

Example for Oracle 5 driver:

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

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

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

6.3.5 Buildomatic: Change your Deployed JDBC Driver

When you run the buildomatic target `deploy-webapp-pro` the JDBC driver for your specified database will be copied to your application server.

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

<js-install>/buildomatic/default_master.properties

You will set the maven.jdbc.artifactId and the maven.jdbc.version to point to the name of the driver you would like to use:

maven.jdbc.artifactId=<first-part-of-filename>
maven.jdbc.version=<version-part-of-filename>

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

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

Change the deployed JDBC driver example for Oracle:

Say, for example, you would like to use the Oracle4 JDBC driver. You would like this driver to be automatically deployed to your application server when you run the `deploy-webapp-pro` target. However, we can see by the settings in the following file that the default driver that will be used will be the following:

<js-install>/buildomatic/conf_source/db/oracle/db.properties

maven.jdbc.artifactId=ojdbc5

maven.jdbc.version=11.2.0

So the driver used is: ojdbc5-11.2.0.jar

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

maven.jdbc.artifactId=ojdbc14

maven.jdbc.version=10.2.0

Now, when you run deploy-webapp-pro, the 14 driver will be used.

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

To deploy other JDBC JAR files:

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

6.3.6 Buildomatic: JasperReports Server WAR File Location

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

<js-install>/jasperserver-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/mysql/jdbc/mysql-connector-java-5.1.10.jar
  Copied to <tomcat>/lib

6.3.7 Buildomatic: Sample Data Catalog ZIP Files

Buildomatic includes export files which hold the JasperReports Server sample data (that have examples of new features). This sample data is loaded when you run the buildomatic target `import-sample-data-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-mysql-minimal-pro.zip

js-catalog-mysql-pro.zip

js-catalog-postgresql-minimal-pro.zip

js-catalog-postgresql-pro.zip

js-catalog-oracle-minimal-pro.zip

js-catalog-oracle-pro.zip

6.4 Additional Notes on Databases

This section provides additional information on the following databases:

- **Notes on the MySQL Database**
- **Notes on the Oracle Database**
- **Notes on the DB2 Database**

For each database, there are commands to:

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

6.4.1 Notes on the MySQL Database

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

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

6.4.1.1 Manual Creation of the JasperReports Server Database

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

Run the following commands:

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

mysql -u root -p
mysql>create database jasperserver character set utf8;
mysql>grant all on *.* to jasperdb@localhost identified by 'password';
mysql>flush privileges; (reload privilege tables)
mysql>use jasperserver;
mysql>source js-pro-create.ddl
mysql>source quartz.ddl
mysql>exit
```

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

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

6.4.1.2 Manual Import of Default Minimal Data

First, follow the instructions in **Chapter 14, “Configuring the Import-Export Utilities,” on page 109**.

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

Then run the following commands to perform the import:

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

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

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

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

6.4.1.3 Manual Creation of Sample Databases

Run the following commands:

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

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

6.4.2 Notes on the Oracle Database

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

These commands should be run from the Windows or Linux command line.

6.4.2.1 Manual Creation of the JasperReports Server Database

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

Run the following commands:

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

sqlplus /nolog (start sqlplus client)
SQL> connect SYS/password@ORCL as SYSDBA (use your SYS password, your SID)
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
```

6.4.2.2 Manual Import of Minimal Data

First, follow the instructions in **Chapter 14, "Configuring the Import-Export Utilities," on page 109**.

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

Then run the following commands to perform the import:

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

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

```
js-import --input-zip ../buildomatic/install_resources/export/js-catalog-oracle-pro.zip
```

6.4.2.3 Manual Creation of Sample Databases

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.

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

Set the NLS_LANG variable that Oracle should use for the sugarcrm database:

Windows:	set NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P1
Linux:	export NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P1

Run the following commands:

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

sqlplus /nolog (start sqlplus client)
SQL> connect SYS/password@ORCL as SYSDBA (use your SYS password, your SID)
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-oracle.sql (populate data - runs slowly)
SQL> connect foodmart/password@ORCL
SQL> @foodmart-oracle.sql (populate data - runs slowly. Make sure file is unzipped)
SQL> @supermart-update.sql
SQL> exit
```

6.4.3 Notes on the DB2 Database

6.4.3.1 Manual Creation of the JasperReports Server Database

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

The example commands below have been tested at Jaspersoft. The commands to be used on your DB2 instance may be different.

These commands should be run from the Windows or Linux command line.

Change to the following directory:

```
cd <js-install>/buildomatic/install_resources/sql/db2
```

Run the following commands in the DB2 command window:

```
db2 create database jsprsrvr using codeset utf-8 territory us
db2 -tf js-pro-create.ddl
db2 -tf quartz.ddl
```

6.4.3.2 Manual Import of Default Users

First, follow the instructions in **Chapter 14, "Configuring the Import-Export Utilities," on page 109**.

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

Then run the following commands to perform the import:

```
cd <js-install>/buildomatic
js-import --input-zip ../buildomatic/install_resources/export/js-catalog-db2-minimal-pro.zip
```

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

```
js-import --input-zip ../buildomatic/install_resources/export/js-catalog-db2-pro.zip
```

6.4.3.3 Manual Creation of Sample Databases

Change to the following directory:

```
cd <js-install>/buildomatic/install_resources/sql/db2
```

Run the following commands in the DB2 command window:

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

6.5 Notes on the Hibernate Properties File

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

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

Within the jasperserver-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.

Here are some example hibernate property values.

MySQL:	`metadata.hibernate.dialect=org.hibernate.dialect.MySQLDialect`
Oracle:	`metadata.hibernate.dialect=com.jaspersoft.ji.hibernate.dialect.OracleUnicodeDialect`
DB2:	`metadata.hibernate.dialect=org.hibernate.dialect.DB2Dialect`

6.6 Notes on Database Connections for Tomcat

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

When the buildomatic target `deploy-webapp-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

Tomcat will often create a copy of the context.xml file with a changed name that will be read instead of the one found in the jasperserver-pro war file. This is often a source of confusion for Tomcat users who attempt change their database settings. If you change your settings, you should delete the file in this location:

<tomcat>/conf/Catalina/localhost

6.7 Notes on Data Source Definitions for JBoss

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

When the buildomatic target `deploy-webapp-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

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

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

6.7.1 Notes on Extra JBoss Configuration Steps

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

For more information, refer to troubleshooting section **A.12, "JBoss Modifications," on page 120**.

6.8 Notes on Database Connections for GlassFish

After setting up the buildomatic configuration for your database, the GlassFish data source definition file js-glassfish-ds.xml will be automatically created with the appropriate settings. When the buildomatic target `deploy-webapp-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 datasource definition manually, you can run a command similar to the following:

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

6.9 Report Scheduling Configuration with Quartz

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

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

<app-server-path>/jasperserver-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

There are four main configurations to be discussed in this section:

- Mail Server Configuration
- Quartz Driver Delegate Class
- Report Scheduler Web URI

- Quartz Table Prefix

6.9.1 Mail Server Configuration Settings

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

<table>
<tr><th colspan="3">Configuration File</th></tr>
<tr><td colspan="3">&lt;app-server&gt;/&lt;deployment&gt;/WEB-INF/js.quartz.properties</td></tr>
<tr><th colspan="2">Property</th><th>Description</th></tr>
<tr><td colspan="2"><code>report.scheduler.mail.sender.host</code></td><td>The name of the computer hosting the mail server.</td></tr>
<tr><td colspan="2"><code>report.scheduler.mail.sender.username</code></td><td>The name of the user in the mail server that JasperReports Server can use.</td></tr>
<tr><td colspan="2"><code>report.scheduler.mail.sender.password</code></td><td>The password of the mail server user.</td></tr>
<tr><td colspan="2"><code>report.scheduler.mail.sender.from</code></td><td>The address that appears in the <b>From</b> field on email notifications.</td></tr>
<tr><td colspan="2"><code>report.scheduler.mail.sender.protocol</code></td><td>The protocol that the mail server uses. JasperReports Server only supports SMTP.
Note: Your entry must be lower case. For example: <code>smtp</code></td></tr>
<tr><td colspan="2"><code>report.scheduler.mail.sender.port</code></td><td>The port number that the mail server uses. For SMTP, the default is typically 25 (values other than 25 may not work in earlier JasperServer versions).</td></tr>
<tr><th colspan="3">Configuration File</th></tr>
<tr><td colspan="3">&lt;app-server&gt;/&lt;deployment&gt;/WEB-INF/applicationContext-report-scheduling.xml</td></tr>
<tr><th>Property</th><th>Bean</th><th>Description</th></tr>
<tr><td><code>javaMailProperties key="mail.smtp.auth"</code></td><td><code>reportScheduler MailSender</code></td><td>If your mail server requires authentication, change this property from <code>false</code> to <code>true</code>.</td></tr>
</table>

6.9.2 Database settings for the Quartz Driver Delegate Class

The Quartz driver delegate class is a class which Quartz uses to interact with the JDBC driver. For the PostgreSQL and DB2 databases it needs a non-default setting.

If you installed using buildomatic these settings are handled automatically.

To set this value manually, edit the following file:

<table>
<tr><th colspan="3">Configuration File</th></tr>
<tr><td colspan="3">&lt;app-server&gt;/&lt;deployment&gt;/WEB-INF/js.quartz.properties</td></tr>
<tr><th>Property</th><th>Database</th><th>Value</th></tr>
</table>

quartz.delegateClass	MySQL	org.quartz.impl.jdbcjobstore.StdJDBCDelegate
	PostgreSQL	org.quartz.impl.jdbcjobstore.PostgreSQLDelegate
	Oracle	org.quartz.impl.jdbcjobstore.StdJDBCDelegate
	SQL Server	org.quartz.impl.jdbcjobstore.StdJDBCDelegate
	DB2	org.quartz.impl.jdbcjobstore.DB2v8Delegate

6.9.3 Settings for the Report Scheduler Web URI

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

If you installed using buildomatic these settings are handled automatically.

To set this value manually, edit the following file:

Configuration File		
<app-server>/<deployment>/WEB-INF/js.quartz.properties		
Property	**App Server**	**Example Value**
report.scheduler. web.deployment.uri	Apache Tomcat	http://localhost:8080/jasperserver-pro
	JBoss	http://localhost:8080/jasperserver-pro
	GlassFish	http://localhost:8080/jasperserver-pro
	WebSphere	http://localhost:9080/jasperserver-pro
	WebLogic	http://localhost:7001/jasperserver-pro

To manually edit, you would edit the following file:

6.9.4 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 using buildomatic these settings are handled automatically.

To set this value manually, edit the following file:

Configuration File	
<app-server>/<deployment>/WEB-INF/js.quartz.properties	
Property	**Description**
quartz.tablePrefix	The prefix for the quartz table, including any schema name, for example `JSPRSRVR.QRTZ_` for DB2.

6.9.5 Settings for Import-Export

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

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

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

Configuration File	
<js-install>/buildomatic/conf_source/iePro/js.quartz.properties	
Property	**Description**
`quartz.delegateClass`	Set to the same value as described in section **6.9.2, "Database settings for the Quartz Driver Delegate Class," on page 57**.
`quartz.tablePrefix`	Set to the same value as described in section **6.9.4, "Settings for the Quartz Table Prefix," on page 58**

6.10 Notes on Updating XML/A Connection Definitions

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

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

There are two sample OLAP views that use this connection:

- Foodmart Sample XMLA OLAP View
- SugarCRM Sample XMLA OLAP View

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

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

Chapter 7 Installing the WAR File for WebSphere

JasperReports Server supports deployment on the IBM WebSphere Application Server. Deployment is performed by the WebSphere administrator using the WebSphere Administrative Console.

This chapter contains the following sections:

7.1 Obtaining the WAR File Distribution Zip

The WAR file distribution comes in a file named jasperreports-server-4.0-bin.zip in the compressed ZIP format. To download the WAR file distribution, contact Jaspersoft technical support or your sales representative.

7.2 Unpacking the WAR File Distribution Zip

Refer to section **5.3, "Unpacking the WAR File Distribution Zip," on page 30**.

7.3 Setting Up the JasperReports Server Database

JasperReports Server requires its own database to store information such as users, organizations, and the repository. In addition, the WAR file distribution includes two sample databases containing data for optional demos. For evaluation, Jaspersoft recommends you install the sample databases and the demos that use them. In a production environment, you only need to set up the database. In either case, the databases must be created and initialized before JasperReports Server is deployed in WebSphere.

To setup the database and optional sample databases, you can use either the buildomatic scripts or perform the commands manually. For simplicity, Jaspersoft recommends you use the buildomatic scripts.

- To use the buildomatic scripts, refer to section **5.6, "Configuring the Buildomatic Scripts," on page 32** and section **5.7, "Installing JasperReports Server," on page 33**. The condition for using the buildomatic scripts is that WebSphere must be installed in the default location.
- To perform the commands manually, refer to section **6.4, "Additional Notes on Databases," on page 51**.

In either case, follow the steps that describe the commands for your specific database vendor.

7.4 Configuring Database Connections

By default, the database configuration files included with the JasperReports Server WAR file distribution are set to connect to a database with the following settings:

Database Setting	MySQL	Oracle	DB2	SQL Server	PostgreSQL
Host	localhost	localhost	localhost	localhost	localhost
Name or SID	jasperserver	Orcl	jsprsrvr	jasperserver	jasperserver
User	root	jasperserver	db2admin	sa	postgres
Password	password	password	password	sa	postgres
Port	3306	1521	50000	1433	5432
Hibernate Dialect	MySQLDialect	OracleUnicode Dialect	DB2SQLDialect	SQLServerDialect	PostgresqlNoBlob Dialect
Quartz Driver Delegate	StdJDBCDelegate	StdJDBCDelegate	DB2v8Delegate	StdJDBCDelegate	PostgreSQL Delegate

The settings in the table above are used in the following sections to configure JDBC providers, database users, and JDBC data sources in WebSphere, according to your database:

- **7.4.1, "Configuring JDBC Data Sources for MySQL," on page 62**
- **7.4.2, "Configuring JDBC Data Sources for Oracle," on page 64**
- **7.4.3, "Configuring JDBC Data Sources for DB2," on page 65**

The Hibernate and Quartz information is used in section **7.5, "Configuring Hibernate and Quartz in the WAR File," on page 67**. For SQL Server and PostgreSQL, follow similar procedures using the values from the table above.

7.4.1 Configuring JDBC Data Sources for MySQL

To define a JDBC provider:

1. In the Administrative Console, navigate to **Resources > JDBC > JDBC Providers**.
2. Click **New** button to create a new JDBC Provider.
3. Enter the following values:

Field Name	Value
Database type	User-defined
Implementation class name	com.mysql.jdbc.jdbc2.optional.MysqlConnectionPoolDataSource
Name	MySQL JDBC Provider

4. Click **Next** and enter the following JAR file, including its path:

   <js-install>\buildomatic\conf_source\db\mysql\jdbc\mysql-connector-java-<ver>-bin.jar

   The version number <ver> must be 3.1.14 or greater. You may also copy the JAR file to a location in your WebSphere deployment and specify that location here.
5. Click **Next**, review the information and click **Finish**.
6. On the JDBC Providers page, click "**Save** directly to the master configuration."

To define a database user:

1. In the Administrative Console, navigate to **Security > Secure administration, application and infrastructure**.
2. In the Authentication section, expand Java Authentication and Authorization Service, and click **J2C authentication data**.
3. Click **New** and enter the following user alias, user ID, and password. These are the credentials that will be used to access the database:

Field Name	Value
Alias	mysql_jasperdb
User ID	root
Password	password

4. Click **OK**, review the new user in the table, and click "**Save** directly to the master configuration" again.

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

1. In the Administrative Console, navigate to **Resources > JDBC > Data Sources**.
2. Click **New** button to create a new JDBC Data Source.
3. Enter the following values:

Field Name	Value
Data source name	jasperserver
JNDI name	jdbc/jasperserver
Component-managed authentication alias	<node>/mysql_jasperdb

4. Click **Next** and choose **Select an existing JDBC provider**, then select MySQL JDBC Provider from the drop-down list.
5. Click **Next** and specify the default helper class (com.ibm.websphere.rsadapter.GenericDataStoreHelper). Also select the checkbox to use this data source in container managed persistence (CMP).
6. Click **Next**, review the summary information and click **Finish**.
7. In the list of JDBC data sources, click the link for the newly created **jasperserver** data source.
8. In the section labeled Additional Properties on the right, click **Custom Properties**.
9. Click on the filter icon button, select **Name** as the filter and enter `url` as the search term.
10. Click **Go**, then click on the **url** property. Enter the following value and save the change:

    ```
    jdbc:mysql://localhost:3306/jasperserver?useUnicode=true&amp;characterEncoding=UTF-8
    ```
11. Go back to the list of JDBC data sources, select the checkbox for the **jasperserver** data source, and click **Test** to verify the connection.

12. If you plan to run the sample reports, repeat step 2 through step 11 with 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>/mysql_jasperdb	<node>/mysql_jasperdb

13. Click "**Save** directly to the master configuration."

7.4.2 Configuring JDBC Data Sources for Oracle

To define a JDBC provider:

1. In the Administrative Console, navigate to **Resources > JDBC > JDBC Providers**.
2. Click **New** button to create a new JDBC Provider.
3. Enter the following values:

Field Name	Value
Database type	Oracle
Provider type	Oracle JDBC Driver
Implementation type	Connection pool data source
Name	Oracle JDBC Driver

4. Click **Next** and enter the following path:

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

   You may also copy the ojdbc5-11.2.0.jar file to a location in your WebSphere deployment and specify that location here.
5. Click **Next**, review the information and click **Finish**.
6. On the JDBC Providers page, click "**Save** directly to the master configuration."
7. On the JDBC Providers page, click the newly created **Oracle JDBC Driver** to edit it.
8. Locate the filename ojdbc14.jar and replace it with the filename ojdbc5-11.2.0.jar, then save the changes.

To define database users:

1. In the Administrative Console, navigate to **Security > Secure administration, application and infrastructure**.
2. In the Authentication section, expand Java Authentication and Authorization Service, and click **J2C authentication data**.
3. Click **New** and enter the following user alias, user ID, and password for each of the three users. These are the credentials that will be used to access the database:

Field Name	Values		
Alias	jasperserver_user	foodmart_user	sugarcrm_user
User ID	jasperserver	foodmart	sugarcrm
Password	password	password	password

4. Click **OK** after entering the information for each user.
5. Review the new users in the table and click "**Save** directly to the master configuration."

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

1. In the Administrative Console, navigate to **Resources > JDBC > Data Sources**.

2. Click **New** button to create a new JDBC Data Source.
3. Enter the following values:

Field Name	Value
Data source name	jasperserver
JNDI name	jdbc/jasperserver
Component-managed authentication alias	<node>/jasperserver_user

4. Click **Next** and choose **Select an existing JDBC provider**, then select Oracle JDBC Provider from the drop-down list.
5. 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

6. Click **Next**, review the summary information and click **Finish**.
7. In the list of JDBC data sources, select the checkbox for the **jasperserver** data source, and click **Test** to verify the connection.
8. If you plan to run the sample reports, repeat step 2 through step 7 with 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

9. Click "**Save** directly to the master configuration."

7.4.3 Configuring JDBC Data Sources for DB2

To define a JDBC provider:

1. In the Administrative Console, navigate to **Resources > JDBC > JDBC Providers**.
2. Click **New** button to create a new JDBC Provider.
3. Enter the following values:

Field Name	Value
Database type	DB2
Provider type	DB2 Universal JDBC Driver Provider
Implementation type	Connection pool data source
Name	DB2 Universal JDBC Driver Provider

4. Click **Next** and enter the following path:

DB2 8.x: <js-install>\buildomatic\conf_source\db\db2\jdbc\alt-drivers
DB2 9.x: <js-install>\buildomatic\conf_source\db\db2\jdbc

You may also copy the db2jcc-x.y.jar file to a location in your WebSphere deployment and specify that location here. If JasperReports Server is deployed on the same host as DB2, delete the following file to avoid conflicts:

<db2>/SQLLIB/java/db2jcc.jar

5. Click **Next**, review the information and click **Finish**.
6. On the JDBC Providers page, click “**Save** directly to the master configuration.”
7. On the JDBC Providers page, click the newly created **DB2 Universal JDBC Driver Provider** to edit it.
8. Locate the filename db2jcc.jar and replace it with the appropriate filename, then save the changes:

DB2 8.x:	db2jcc-8.2.jar
DB2 9.x:	db2jcc-9.5.jar

To define a database user:

1. In the Administrative Console, navigate to **Security > Secure administration, application and infrastructure**.
2. In the Authentication section, expand Java Authentication and Authorization Service, and click **J2C authentication data**.
3. Click **New** and enter the following user alias, user ID, and password. These are the credentials that will be used to access the database:

Field Name	Value
Alias	db2admin_user
User ID	db2admin
Password	password

4. Click **OK**, review the new user in the table, and click “**Save** directly to the master configuration.”

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

1. In the Administrative Console, navigate to **Resources > JDBC > Data Sources**.
2. Click **New** button to create a new JDBC Data Source.
3. Enter the following values:

Field Name	Value
Data source name	jasperserver
JNDI name	jdbc/jasperserver
Component-managed authentication alias	<node>/db2admin_user

4. Click **Next** and choose **Select an existing JDBC provider**, then select DB2 Universal JDBC Driver Provider from the drop-down list.
5. Click **Next** and enter the following values:

Field Name	Value
Database name	jsprsrvr
Driver type	4
Server name	localhost
Port number	50000
Use this data source in CMP	selected

6. Click **Next**, review the summary information and click **Finish**.
7. In the list of JDBC data sources, click the link for the newly created **jasperserver** data source.

8. In the section labeled Additional Properties on the right, click **Custom Properties**.
9. Edit the following properties, including adding the ones that don't exist, then save the changes:

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

10. Go back to the list of JDBC data sources, select the checkbox for the **jasperserver** data source, and click **Test** to verify the connection.
11. If you plan to run the sample reports, repeat **step 2** through **step 10** with 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>/db2admin_user	<node>/db2admin_user

Field Name	Value	
Database name	foodmart	sugarcrm
Driver type	4	4
Server name	localhost	localhost
Port number	50000	50000
Use this data source in CMP	selected	selected

Property Name	Value	
currentSchema	FOODMART	SUGARCRM
resultSetHoldability	1	1

The other properties (fullyMaterializeLobData, fullyMaterializeInputStreams, progressiveStreaming, and progresssiveLocators) are not needed for the sample databases.

12. Click "**Save** directly to the master configuration."

7.5 Configuring Hibernate and Quartz in the WAR File

Before deploying the JasperReports Server WAR file, you must update the Hibernate and Quartz settings that it contains, as described in the following procedure.

1. The WAR file is an archive format in a single file. To access the configuration files it contains, you must extract them with the following commands:

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

```
"%JAVA_HOME%\bin\jar" xf jasperserver-pro.war WEB-INF/js.quartz.properties
```

The `jar` command will create the WEB-INF directory in the current location and place the extracted files there.

2. Open the file WEB-INF/hibernate.properties for editing and locate the `metadata.hibernate.dialect` property. Set its value according to your database:

MySQL:	`metadata.hibernate.dialect=org.hibernate.dialect.MySQLDialect`
Oracle:	`metadata.hibernate.dialect=com.jaspersoft.ji.hibernate.dialect.OracleUnicodeDialect`
DB2:	`metadata.hibernate.dialect=org.hibernate.dialect.DB2Dialect`

3. Open the file WEB-INF/js.quartz.properties for editing and locate the `report.scheduler.web.deployment.uri` property. Set its value to the hostname, port, and path where you intend to deploy JasperReports Server, for example:

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

If you are using PostgreSQL or DB2, set the Quartz delegate class in the js.quartz.properties file:

PostgreSQL:	`quartz.delegateClass=org.quartz.impl.jdbcjobstore.PostgreSQLDelegate`
DB2:	`quartz.delegateClass=org.quartz.impl.jdbcjobstore.DB2v8Delegate`
All others:	`quartz.delegateClass=org.quartz.impl.jdbcjobstore.StdJDBCDelegate`

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 the individual configuration files, you must replace them into the WAR file archive. Use the following command:

```
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
```

6. Delete the WEB-INF directory that was created, along with the edited files it contains.

7.5.1 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. 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, for WebSphere, a new work manager class is used to run scheduled report jobs. 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.properites`.

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.

7.5.2 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.

7.6 Setting Up the JasperReports Server License

JasperReports Server Professional or Enterprise requires a license to run. The JasperReports Server WAR distribution comes with an evaluation license that is valid for a number of 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

You must put your license file into a home folder of a user account being used to start WebSphere Application Server. This account may differ from the logged user account. For example, if WebSphere is running on Windows Server as a service, the system account is used to start the WebSphere, and its home folder would be:

%SystemRoot%\system32\config\systemprofile

7.7 Deploying the JasperReports Server WAR to the Application Server

Now that the databases are created and all files are configured, you can deploy the JasperReports Server WAR file in WebSphere.

1. In the Administrative Console, navigate to **Applications > Install New Application**.
2. Enter the following information:

Field Name	Value
Path to the new application	Local File System
Full path	<js-install>/jasperserver-pro.war
Context root	jasperserver-pro

3. Click **Next**. On the Select installation options page, leave all the default settings.
4. Click **Next**. On the Map modules to servers page, make sure the JasperReports Server module is mapped to the cell, node, and server that you want.
5. Click **Next**. Set the correct target JNDI name for each of the resource references.
6. Click **Next**. On the Map virtual hosts page, choose the virtual host where JasperReports Server is exposed, usually the default one.
7. Click **Next**, review the summary information and start the installation process.
8. Make sure JasperReports Server is successfully installed, then click “**Save** directly to the master configuration.”

7.8 Setting Java JVM Options

In order for the JasperReports Server analysis XML/A functionality to work, special Java JVM options need to be set. This is because both WebSphere and JasperReports Server provide a full web services implementation and there are class conflicts between the two. JVM options also provide the optimal resources for running JasperReports Server.

To configure your Java JVM options:

1. Select **Application Servers > (server name) > Process Definition > Java Virtual Machine > Generic JVM Options**.
2. In the text box named **Generic JVM Options**, paste in the following JVM options that will explicitly specify JasperReports Server classes for AXIS and Xalan, as well as optimize JVM resources:

Generic JVM Options on Windows	
Options for all databases	set JAVA_OPTS=%JAVA_OPTS% -Dclient.encoding.override=UTF-8 -Xms128m -Xmx512m -XX:PermSize=32m -XX:MaxPermSize=128m -Xss2m -XX:+UseConcMarkSweepGC -XX:+CMSClassUnloadingEnabled -XX:+CMSPermGenSweepingEnabled -Djavax.xml.soap.MessageFactory=org.apache.axis.soap.MessageFactoryImpl -Djavax.xml.soap.SOAPConnectionFactory=org.apache.axis.soap.SOAPConnectionFactoryImpl -Djavax.xml.soap.SOAPFactory=org.apache.axis.soap.SOAPFactoryImpl -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	export JAVA_OPTS="$JAVA_OPTS -Dclient.encoding.override=UTF-8 -Xms128m -Xmx512m -XX:PermSize=32m -XX:MaxPermSize=128m -Xss2m -XX:+UseConcMarkSweepGC -XX:+CMSClassUnloadingEnabled -XX:+CMSPermGenSweepingEnabled -Djavax.xml.soap.MessageFactory=org.apache.axis.soap.MessageFactoryImpl -Djavax.xml.soap.SOAPConnectionFactory=org.apache.axis.soap.SOAPConnectionFactoryImpl -Djavax.xml.soap.SOAPFactory=org.apache.axis.soap.SOAPFactoryImpl -Djavax.xml.transform.TransformerFactory=org.apache.xalan.processor.TransformerFactoryImpl"
For Oracle	export JAVA_OPTS="$JAVA_OPTS -Doracle.jdbc.defaultNChar=true"

You can cut and paste these options from <js-install>/samples/extra/java-settings/java-settings-websphere.txt.

3. Click Save on the console task bar.
4. Restart the application server.

7.9 Restarting the Application Server

You have completed the configuration steps for JasperReports Server. Because JAVA options and other configuration changes were made, you should restart the application server.

7.10 Starting JasperReports Server

1. In the Administrative Console, navigate to **Applications > Enterprise Applications**.
2. Select the checkbox next to the **jasperserver-pro** application, then click **Start** to start the application.

7.11 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>:9080/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 administrative identity to log in to 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 a "Page cannot be found, HTTP 404" error, you may be running at a WebSphere patch level that needs a further configuration. Refer to troubleshooting section **A.13, "WebSphere: Page Not Found Error on Login," on page 121**.

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 section **5.11.1, "JasperReports Server Heartbeat," on page 36**.

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

7.12 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 sent to users to let them know that new reports are available.

For more information about setting up report scheduling, refer to section **6.9, "Report Scheduling Configuration with Quartz," on page 56**.

7.13 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 section **6.10, "Notes on Updating XML/A Connection Definitions," on page 59**.

7.14 Restarting JasperReports Server

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

7.15 Troubleshooting your JasperReports Server Configuration

7.15.1 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 on resolving these types of errors, refer to troubleshooting section **A.2, "Database Connectivity Errors," on page 116**.

7.15.2 Error Running Report

If you have trouble running reports in your new JasperReports Server instance, refer to troubleshooting section **A.5, "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 section **7.13, "Updating XML/A Connection Definitions (Optional)," on page 71**.

7.15.3 Filter Error

If you encounter the following error:

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

it could be caused by an incorrect ampersand setting on your data source configuration.

The data source line needs to have `&amp;` and not `&` in order to be evaluated correctly.

The URL in section **7.4.1, "Configuring JDBC Data Sources for MySQL," on page 62** should look like the following:

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

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

7.15.4 Error Creating Internationalized Name

If you encounter errors in JasperReports Server when creating resources with internationalized names, and you have an Oracle database, you need to configure your Oracle JDBC driver. Be sure to set the Oracle-specific option listed in the tables of section **7.8, "Setting Java JVM Options," on page 70**.

7.15.5 Xerces Error

If you have a WebSphere log with the following error:

```
SRVE0068E: Uncaught exception thrown in one of the service methods of the servlet:
jasperserver. Exception thrown: org.springframework.web.util.NestedServletException:
Handler processing failed; nested exception is
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.
```

It is caused by a conflict between the IBM JDK which is used by WebSphere and the xercesImpl-2.6.2 library which is bundled with 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 JasperReports Server's xercesImpl library to a newer version. For example, xercesImpl-2.8.1 does not cause the error.

7.15.6 OLAP View Fails With Exception

If you see the following error:

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

```
    ...)
    org/aspectj/lang/reflect/MethodSignature;
```

it could caused by AspectJ requiring class loaders to be tried out in a specific order. Change the default class loader policy with 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	Classes loaded with application class loader first

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

7.15.7 Analysis Options Page Throws Exception

If you see the following error:

```
com.ibm.ws.jsp.JspCoreException: JSPG0049E: /WEB-INF/jsp/olap/editProperties.jsp failed to
compile
```

it could be caused by the JSP compiler using Java 1.3 by default. Try either of the following solutions:

- Add a custom property to the Web Container:

Property Name	Value
com.ibm.ws.jsp.jdkSourceLevel	15

- Add the following property to the WEB-INF/ibm-web-ext.xmi file:

```
<jspAttributes xmi:id="JSPAttribute_113" name="jdkSourceLevel" value="15"/>
```

Chapter 8 Installing the WAR File for WebLogic

JasperReports Server supports deployment on the WebLogic Application Server. Deployment is performed by the WebLogic administrator using the WebLogic Administrative Console or domain config.xml.

This chapter contains the following sections:

8.1 Obtaining the WAR File Distribution Zip

The WAR file distribution comes in a file named jasperreports-server-4.0-bin.zip in the compressed ZIP format. To download the WAR file distribution, contact Jaspersoft technical support or your sales representative.

8.2 Unpacking the WAR File Distribution Zip

Refer to section **5.3, “Unpacking the WAR File Distribution Zip,” on page 30**.

8.3 Setting Up the JasperReports Server Database

JasperReports Server requires its own database to store information such as users, organizations, and the repository. In addition, the WAR file distribution includes two sample databases containing data for optional demos. For evaluation, Jaspersoft recommends you install the sample databases and the demos that use them. In a production environment, you only need to set up the database. In either case, the databases must be created and initialized before JasperReports Server is deployed in WebLogic.

To setup the database and optional sample databases, you can use either the buildomatic scripts or perform the commands manually. For simplicity, Jaspersoft recommends you use the buildomatic scripts.

- To use the buildomatic scripts, refer to section **5.6, "Configuring the Buildomatic Scripts," on page 32** and section **5.7, "Installing JasperReports Server," on page 33**. The condition for using the buildomatic scripts is that WebLogic must be installed in the default location.
- To perform the commands manually, refer to section **6.4, "Additional Notes on Databases," on page 51**.

In either case, follow the steps that describe the commands for your specific database vendor.

8.4 Configuring Database Connections

By default, the database configuration files included with the JasperReports Server WAR file distribution are set to connect to a database with the following settings:

Database Setting	MySQL	Oracle	DB2	SQL Server	PostgreSQL
Host	localhost	localhost	localhost	localhost	localhost
Name or SID	jasperserver	Orcl	jsprsrvr	jasperserver	jasperserver
User	root	jasperserver	db2admin	sa	postgres
Password	password	password	password	sa	postgres
Port	3306	1521	50000	1433	5432
Hibernate Dialect	MySQLDialect	OracleUnicode Dialect	DB2SQLDialect	SQLServerDialect	PostgresqlNoBlob Dialect
Quartz Driver Delegate	StdJDBCDelegate	StdJDBCDelegate	DB2v8Delegate	StdJDBCDelegate	PostgreSQL Delegate

The following sections show how to configure database connections for a MySQL database in WebLogic. The Hibernate and Quartz information is used in section **8.5, "Configuring Hibernate and Quartz in the WAR File," on page 77**. For other databases, follow similar procedures using values from the table above.

8.4.1 Configuring Data Sources

1. In Administrative Console window, navigate to **JDBC > Data Sources**.
2. Click **New** for each of the data source columns in the following table, and enter the corresponding values:

Parameter Name	JasperReports Server	Foodmart Example	Sugar CRM Example
Name	JasperServerDataBase	FoodmartDataBase	SugarcrmDataBase
JNDI Name	JasperServerDataBase	FoodmartDataBase	SugarcrmDataBase
Database Type	MySQL	MySQL	MySQL

Database Driver	MySQL's Driver (Type 4) Versions: using com.mysql.jdbc.Driver	MySQL's Driver (Type 4) Versions: using com.mysql.jdbc.Driver	MySQL's Driver (Type 4) Versions: using com.mysql.jdbc.Driver
Supports Global Transactions	Selected	Selected	Selected
One-Phase Commit	Selected	Selected	Selected

8.4.2 Setting Connection Properties

The following table gives sample connection properties for a MySQL database:

Parameter Name	JasperReports Server	Foodmart Example	Sugar CRM Example
Database Name	jasperserver	foodmart	sugarcrm
Host Name	localhost	localhost	localhost
Port	3306	3306	3306
Database User Name	root	root	root
Password	password	password	password
Confirm Password	password	password	password

8.4.3 Testing Database Connection

In the Test Database Connection page, you should check the Data Base URL.

For SugarCRM and Foodmart, it should be correct by default:

jdbc:mysql://localhost:3306/sugarcrm
jdbc:mysql://localhost:3306/foodmart.

The URL for JasperServer Data Base should be changed to the following:

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

8.4.4 Selecting Targets

All data sources should be checked **AdminServer**.

8.5 Configuring Hibernate and Quartz in the WAR File

Before deploying the JasperReports Server WAR file, you must update the Hibernate and Quartz settings that it contains, as described in the following procedure.

1. The WAR file is an archive format in a single file. To access the configuration files it contains, you must extract them with the following commands:

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

   The `jar` command will create the WEB-INF directory in the current location and place the extracted files there.

2. Open the file WEB-INF/hibernate.properties for editing and locate the `metadata.hibernate.dialect` property. Set its value according to your database:

MySQL:	`metadata.hibernate.dialect=org.hibernate.dialect.MySQLDialect`
Oracle:	`metadata.hibernate.dialect=com.jaspersoft.ji.hibernate.dialect.OracleUnicodeDialect`
DB2:	`metadata.hibernate.dialect=org.hibernate.dialect.DB2Dialect`

3. Open the file WEB-INF/js.quartz.properties for editing and locate the `report.scheduler.web.deployment.uri` property. Set its value to the hostname, port, and path where you intend to deploy JasperReports Server, for example:

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

   If you are using PostgreSQL or DB2, set the Quartz delegate class in the js.quartz.properties file:

PostgreSQL:	`quartz.delegateClass=org.quartz.impl.jdbcjobstore.PostgreSQLDelegate`
DB2:	`quartz.delegateClass=org.quartz.impl.jdbcjobstore.DB2v8Delegate`
All others:	`quartz.delegateClass=org.quartz.impl.jdbcjobstore.StdJDBCDelegate`

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 the individual configuration files, you must replace them into the WAR file archive. Use the following command:

   ```
   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
   ```

6. Delete the WEB-INF directory that was created, along with the edited files it contains.

8.5.1 Additional Change for Mail Server Authentication

If your mail server requires authentication, edit the applicationContext-report-scheduling.xml file in the same manner:

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.

8.6 Additional Changes for WebLogic 10.3.0

Edit the `weblogic.xml` file as described below:

1. Extract the `weblogic.xml` file from the WAR archive:

   ```
   "%JAVA_HOME%\bin\jar" xf jasperserver-pro.war WEB-INF/weblogic.xml
   ```

2. Open the file for editing and locate the following element:

   ```
   <session-descriptor>
     <cookie-http-only>false</cookie-http-only>
   </session-descriptor>
   ```

3. Remove the entire session-descriptor element shown above (or comment it out).
4. Save the file and replace it in the archive:

   ```
   "%JAVA_HOME%\bin\jar" uf jasperserver-pro.war WEB-INF/weblogic.xml
   ```

5. Delete the WEB-INF directory that was created, along with the file it contains.

8.7 Preparing the WebLogic Domain

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

8.7.1 Editing the Domain Configuration File

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

It is important that the `stuck-thread-max-time` element appear above the `listen-address` element before the closing `</server>` tag.

8.7.2 Preparing the Domain Library

For WebLogic 9.2 only, copy the following file:

<js-install>/buildomatic/conf_source/db/mysql/jdbc/mysql-connector-java-5.1.10.jar

To: <wl-domain>/lib/

8.7.3 Setting Java Properties

Edit your WebLogic startup script to include the settings described in the following tables, according to your platform. Enter each command on a single line, and substitute the location of your JasperReports Server license file where necessary:

WebLogic Startup Settings on Windows	
Filename	<wl-domain>\bin\startWebLogic.cmd
All versions of WebLogic	set JAVA_OPTIONS=%JAVA_OPTIONS% -Djs.license.directory=C:\Jasperserver-pro\ -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
WebLogic 10 and 10.3	set JAVA_OPTIONS=%JAVA_OPTIONS% -Xms256m -Xmx1024m -XX:PermSize=128m -XX:MaxPermSize=256m -Xss2m -XX:+UseConcMarkSweepGC -XX:+CMSClassUnloadingEnabled -XX:+CMSPermGenSweepingEnabled

WebLogic Startup Settings on Linux	
Filename	<wl-domain>/bin/startWebLogic.sh
All versions	export JAVA_OPTIONS="$JAVA_OPTIONS -Djs.license.directory=/home/<user>/weblogic/jasperlicense/ -Dfile.encoding=UTF-8 -Dcom.sun.xml.namespace.QName.useCompatibleSerialVersionUID=1.0 -Djavax.xml.soap.SOAPConnectionFactory=org.apache.axis.soap.SOAPConnectionFactoryImpl -Djavax.xml.soap.MessageFactory=org.apache.axis.soap.MessageFactoryImpl -Djavax.xml.soap.SOAPFactory=org.apache.axis.soap.SOAPFactoryImpl"
WebLogic 10 and 10.3	export JAVA_OPTIONS="$JAVA_OPTIONS -Xms256m -Xmx1024m -XX:PermSize=128m -XX:MaxPermSize=256m -Xss2m -XX:+UseConcMarkSweepGC -XX:+CMSClassUnloadingEnabled -XX:+CMSPermGenSweepingEnabled"

For WebLogic 9.2, you must also update the JRockit JDK from `jrockit90_150_04` to `jrockit_150_11`. For more information about JDKs with WebLogic 9.2, refer to the product documentation.

8.8 Deploying JasperReports Server to WebLogic

Now that the databases are created and all files are configured, you can deploy the JasperReports Server WAR file in WebLogic.

1. In the Administrative Console, click the **Lock & Edit** button and navigate to **Deployments**.
2. On the Deployments page click the **Install** button.
3. Select the path to <js-install>.
4. Select the servers or clusters to which you want to deploy JasperReports Server.
5. 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

8.9 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**.

8.10 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 in to 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 section **5.11.1, "JasperReports Server Heartbeat," on page 36**.

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

8.11 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 section **6.9, "Report Scheduling Configuration with Quartz," on page 56**.

8.12 Restarting JasperReports Server

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

8.13 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 7001. Follow the procedure in section **6.10, "Notes on Updating XML/A Connection Definitions," on page 59**.

8.14 Troubleshooting Your JasperReports Server Configuration

8.14.1 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 on resolving these types of errors, refer to troubleshooting section **A.2, "Database Connectivity Errors," on page 116**.

8.14.2 Error Running Report

If you have trouble running reports in your new JasperReports Server instance, refer to troubleshooting section **A.5, "Error Running a Report," on page 118**.

Chapter 9 Upgrading from JasperServer 3.7 to 4.0

This chapter describes the steps necessary to carry out an upgrade from JasperServer 3.7 to JasperReports Server 4.0. These steps use the JasperReports Server WAR File Distribution ZIP release package and the included buildomatic scripts for the upgrade procedure.

The product name has changed slightly in 4.0. It is now known as JasperReports Server 4.0.

This chapter contains the following sections:

9.1 Standard Upgrade Steps

This section lists the standard upgrade steps. These general steps are always available with each new JasperReports Server release.

1. Back up your 3.7 JasperServer instance
2. Export your 3.7 repository data
3. Download and setup the 4.0 JasperReports Server WAR file distribution zip
4. Run auto-upgrade script

If your instance of JasperServer 3.7 has any custom modifications or extensions, you will need to keep track of these and re-integrate them into your 4.0 instance after the upgrade is complete.

9.2 Backing Up Your JasperServer 3.7 Instance

First you must backup your JasperServer WAR file and your jasperserver database so that they can be restored in case there is a problem with the upgrade. These steps are performed from the command line in a Windows or Linux shell.

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

1. Back up the jasperserver-pro directory in Tomcat to a backup directory:
   a. cd <tomcat>
   b. mkdir js-3.7-war-backup
   c. copy <tomcat>/webapps/ jasperserver-pro to <tomcat>/js-3.7-war-backup
   d. delete the <tomcat>/webapps/jasperserver-pro directory
2. Back up the jasperserver database. Go to the location where you originally unpacked the 3.7 WAR file distribution zip or installed from the JasperServer 3.7 installer:
   a. cd <js-install-3.7> (the location of your original unpacked 3.7 WAR file distribution)
   b. Run the following command:

```
Windows: mysqldump --user=root --password=<password> jasperserver > js-db-3.7-dump.sql
Linux:   mysqldump --user=root --password=<password> --host=127.0.0.1 jasperserver >
         js-db-3.7-dump.sql
```

If you installed the previous release from the installer, specify `--user=jasperdb` in this command.

If you receive a packet size error, see section **A.2.6, "Maximum Packet Size in MySQL," on page 117**.

Jaspersoft has tested the `mysqldump` utility for backing up and restoring MySQL databases, but there are other MySQL backup mechanisms, some of which may work better for your JasperServer installation.

9.3 Export Your 3.7 Repository Data

You will need to export your 3.7 repository data using the JasperReports Server export utility. There are two ways to export. One is using the buildomatic scripts (if you originally installed using buildomatic). Or you will use the js-export.bat/.sh script found in the <js-install>/scripts folder.

9.3.1 Export Using Buildomatic Scripts

If you have buildomatic and your `default_master.properties` file configured, you can export your 3.7 repository data using the following commands:

1. `cd <js-install-3.7>/buildomatic`
2. Run buildomatic with the export target:

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

This operation uses the export option `--everything` which exports all your repository data.

Note the location of this export file so that you can point to it for the 4.0 upgrade.

9.3.2 Export Using js-export Script

To use the js-export.bat/.sh script, you will move to the <js-install-3.7>/scripts folder. If you are using the MySQL database then the js-export script should already be configured to run. If you are using a different database, or you have changed database passwords you will need to update the js-export configuration. For information on configuring the 3.7 import-export utility see section **14.6, "Configuring the Import-Export Utility for JasperServer 3.7," on page 113**.

Run the following commands:

1. `cd <js-install-3.7>/scripts`
2. Run the export script:

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

This operation uses the export option `--everything` which exports all your repository data.

Note the location of this export file so that you can point to it for the 4.0 upgrade.

9.4 Preparing the JasperReports Server 4.0 WAR File Distribution

We will use the buildomatic scripts included in the 4.0 WAR file distribution ZIP release package in order to carry out the upgrade. Follow the steps in the sections listed below to obtain and unpack the WAR file distribution ZIP file:

1. Follow steps in section **5.2, "Obtaining the WAR File Distribution Zip," on page 30**.
2. Follow steps in section **5.3, "Unpacking the WAR File Distribution Zip," on page 30**.

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

```
<js-install-4.0>
```

9.5 Configuring Buildomatic for Your Database and Application Server

This upgrade procedure is based on using the "buildomatic" scripts which are included with the WAR File Distribution ZIP release package. The buildomatic scripts are based on the `ant` utility and require the Java Development Kit (JDK) to run. If you don't have Java available in your environment, see section **9.9.1, "Java Path for Buildomatic When Upgrading," on page 87**.

Follow the configuration steps that match your database and application server in section **5.6, "Configuring the Buildomatic Scripts," on page 32**. Below is an example configuration using the MySQL database.

9.5.1 Example Buildomatic Configuration (using MySQL)

All upgrade configuration is handled by a single file that is named `default_master.properties`. Jaspersoft provides a sample configuration file for each database. You must specify your database credentials and your application server location, and rename the file to `default_master.properties`.

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

You must rename and copy the sample file to this location: `<js-install-4.0>/buildomatic`.

1. Locate the `mysql_master.properties` file:

Database	Master Properties File
MySQL	<js-install-4.0>/buildomatic/sample_conf/mysql_master.properties

2. Copy the file to `<js-install-4.0>/buildomatic`.
3. Rename the file `default_master.properties`.
4. Edit `default_master.properties` for your database and application server:

Database	Sample Property Values

MySQL	appServerType=tomcat6 (or tomcat5, jboss, glassfish)
	appServerDir=c:\\apache-tomcat-6.0.26 (for example)
	dbUsername=root
	dbPassword=password
	dbHost=localhost

9.6 Upgrading to JasperReports Server 4.0

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

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

Commands	Description
cd <js-install-4.0>/buildomatic	
js-upgrade-newdb.bat <path>/js-3.7-export.zip	(Windows) Upgrade jasperserver-pro war file, drop and recreate database, import 3.7 data
js-upgrade-newdb.sh <path>/js-3.7-export.zip	(Linux) Upgrade jasperserver-pro war file, data, drop and recreate database, import 3.7 data

You can use a fully path or a relative path. If the path you specify has spaces in it, you should use quotes "<path>/js-3.7.-export.zip"

On MySQL, if you receive an error about packet size, see section **A.2.6, "Maximum Packet Size in MySQL," on page 117**.

9.7 Starting JasperReports Server 4.0

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

9.8 Logging Into JasperReports Server 4.0

If your application server and JasperReports Server 4.0 were started cleanly, you can now prepare to login.

9.8.1 Clearing Your Browser Cache

Before you log into 4.0, make sure and clear your Browser cache. JavaScript files, which enable UI elements of JasperReports Server, are typically cached by the Browser. The cache should be cleared to ensure that the most current files are used.

For JasperReports Server 4.0, the UI has been significantly enhanced. It will be very important to clear the browser cache.

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

9.8.2 Logging Into JasperReports Server

Login using the following URL, user IDs, and passwords:

URL: http://localhost:8080/jasperserver-pro

User ID	Password	Description
superuser	<your-password>	System-wide administrator
jasperadmin	<your-password>	Administrator for the default organization

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

Your JasperReports Server instance has now been upgraded to 4.0. If there are problems on startup or login refer to troubleshooting section **A.2, "Database Connectivity Errors," on page 116**.

9.9 Additional Notes on JasperReports Server Upgrade

9.9.1 Java Path for Buildomatic When Upgrading

In order to run the buildomatic scripts (using the `js-ant` utility), you must have the Java JDK 1.5 or 1.6 in your path. Run the buildomatic upgrade commands with the same Java version that you use to run your previous release of JasperReports Server.

If you installed the previous release of JasperReports Server from the installer and did not change its Java environment, it uses the Java 1.5 bundled in <js-install-3.7>/java. Set your command shell to use this version before you run the commands to upgrade:

Windows:
```
set JAVA_HOME="<js-install-3.7>/java"
set PATH=%PATH%;%JAVA_HOME%\bin
```

Linux:
```
export JAVA_HOME=<js-install-3.7>/java
export PATH=$PATH:$JAVA_HOME/bin
```

9.9.2 Handling JasperReports Server Customizations

If you made modifications or customizations to your JasperServer 3.7 application, these configurations are typically found in the WEB-INF/applicationContext-*.xml set of files.

Configuration modifications such as client specific security classes or LDAP server configurations, need to be hand copied from the older 3.7 environment and re-integrated into the new 4.0environment.

9.9.3 Clearing the Application Server Work Directory

Application servers have work directories where JSP files are compiled and cached and other objects are stored. Before you update to a new WAR file or license, the `work` directory should be cleared. The buildomatic `deploy-webapp` target automatically clears the application server's `work` directory but it is a good practice to double-check (in case of permission issues, etc).

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

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

9.9.4 Clearing the Application Server Temp Directory

JasperReports Server uses caching to speed operations within the application. In the application server, caching files are created and stored for this caching functionality. Typically, these cached files are stored in a `temp` directory. It is a good practice to clear this temp folder to avoid any conflicts after the upgrade is complete.

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

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

There is a known issue in Release 4.0 where permission changes on files and folders will not be in effect unless the application server `temp` folder is cleared. For Tomcat, for instance, the location to clear is <tomcat>/temp.

9.9.5 Clearing the Repository Cache Table

In the jasperserver database, compiled JasperReports Library resources are cached in the `JIRepositoryCache` table for increased efficiency at runtime. In some cases, you may encounter errors running reports after an upgrade. Because the JasperReports Library JAR is typically updated with each new JasperReports Server release, old cached items can get out of date and thus cause errors at runtime. If you encounter errors that mention a JasperReports Library "local class incompatible," you should check your repository cache table. In summary, you can clear your jasperserver database cache table whether there are errors or not as part of this upgrade process.

To manually clear this table, run a SQL command similar to the following:

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

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

9.9.6 Updating the XML/A Connections (Optional)

When you upgrade your sample data to 4.0, your XML/A connection sample data will be updated. XML/A connections use JasperReports Server login accounts for authentication. Because of this, and because you would normally modify your default `jasperadmin` password as a standard security procedure, your XML/A connection may fail due to a mismatched password.

If you would like to update your XML/A connections, refer to section **6.10, "Notes on Updating XML/A Connection Definitions," on page 59**.

9.9.7 Upgrading the Liferay Portal

JasperReports Server can be configured to run with the Liferay Portal. If your JasperReports Server is set up to run with Liferay you must do the following steps as part of the upgrade process.

1. You will need to delete the webapps/Jaspersoft folder of the application server hosting Liferay. This deletes libraries used in older versions that conflict with libraries in the latest version.
2. Once this folder is deleted, you can deploy the new portlet WAR.

For more information, refer to the *JasperReports Server Administrator Guide*.

9.10 Older Manual Upgrade Steps

This section has some of the older, manual upgrade steps that were in place before the "auto-upgrade" shell scripts were implemented for JasperReports Server release 4.0. These are here as a reference. It is recommended that you use the auto-upgrade steps from the top of this upgrade chapter.

9.10.1 Manual Upgrade Steps

Older buildomatic targets upgrade steps:

Commands	Description
`cd <js-install-4.0>/buildomatic`	

`js-ant drop-js-db`
`js-ant create-js-db`
`js-ant init-js-db-pro`
	This will delete your jasperserver db. Make sure it is backed up.

`js-ant import-minimal-pro`	
js-ant import-upgrade -DimportFile="`<path-and-filename>`"	The -DimportFile should point to the <path> and <filename> of the js-3.7-export.zip file you created earlier. On Windows, you must use double quotes (`"`) if your path or filename contains spaces. On Linux, you must use double quotes escaped with a backslash (`\"`) in this case.
`js-ant import-sample-data-upgrade-pro`	This step is optional; it loads the 3.7 sample data. The old sample data is overwritten, so you may need to redo certain changes such as configuring the sample data sources for your database.
`js-ant deploy-webapp-pro`	Delete existing 3.7 war file, deploy 4.0 war file.

On MySQL, if you receive an error about packet size, see section **A.2.6, "Maximum Packet Size in MySQL," on page 117**.

CHAPTER 10 UPGRADING USING SQL UPGRADE SCRIPTS

SQL based database upgrade scripts are not available for all database types. Please check the ReleaseNotes.txt document to see availability.

Also, the product name has changed slightly in 4.0. It is now known as JasperReports Server 4.0.

Note that SQL based database upgrade scripts are not always available for a particular JasperReports Server release. And, in general, there is very limited support for using SQL upgrade scripts to upgrade databases other than MySQL. Please check the release notes to see if there is a script available for this use for your database. Instead it is recommended that you upgrade using the process described in **Chapter 9, "Upgrading from JasperServer 3.7 to 4.0," on page 83**.

This section uses an auto-upgrade shell script to carry out the upgrade. It is very similar to the auto-upgrade steps used in the previous chapter except the jasperserver database will not be dropped and recreated. Instead, a SQL script will be used to upgrade the database to a 4.0 database.

This chapter contains the following sections:

- **Upgrade Steps**
- **Backing Up Your JasperServer 3.7 Instance**
- **Preparing the JasperReports Server 4.0 WAR File Distribution**
- **Configuring Buildomatic for Your Database and Application Server**
- **Upgrading to JasperReports Server 4.0**
- **Starting JasperReports Server 4.0**
- **Logging Into JasperReports Server 4.0**
- **Additional Information on Post-Upgrade Steps**

10.1 Upgrade Steps

These are the general steps used in this section.

1. Back up your 3.7 JasperServer instance
2. Download and setup the 4.0 JasperReports Server WAR file distribution zip
3. Run auto-upgrade script

If your instance of JasperServer 3.7 has any custom modifications or extensions, you will need to keep track of these and re-integrate them into your 4.0 instance after the upgrade is complete.

10.2 Backing Up Your JasperServer 3.7 Instance

First you must backup your JasperServer WAR file and your jasperserver database so that they can be restored in case there is a problem with the upgrade. These steps are performed from the command line in a Windows or Linux shell.

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

10.2.1 Backing Up Your JasperServer WAR File

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

1. Go to the <tomcat> directory.
2. Make a new directory named js-3.7-war-backup.
3. Copy <tomcat>/webapps/ jasperserver-pro to <tomcat>/js-3.7-war-backup.
4. Delete the <tomcat>/webapps/jasperserver-pro directory.

10.2.2 Backing Up Your JasperServer Database

Go to the location where you originally unpacked the 3.7 WAR file distribution zip.

1. Go to the <js-install-3.7>   (the location of your original unpacked 3.7 WAR file distribution)
2. Run the following command:

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

Linux:
```
mysqldump --user=root --password=<password> --host=127.0.0.1 jasperserver >
js-db-3.7-dump.sql
```

If you receive an error about packet size, see section **A.2.6, “Maximum Packet Size in MySQL,” on page 117**.

10.3 Preparing the JasperReports Server 4.0 WAR File Distribution

We will use the buildomatic scripts included in the 4.0 WAR file distribution ZIP release package in order to carry out the upgrade. Follow the steps in the sections listed below to obtain and unpack the WAR file distribution ZIP file:

1. Follow steps in section **5.2, “Obtaining the WAR File Distribution Zip,” on page 30**.
2. Follow steps in section **5.3, “Unpacking the WAR File Distribution Zip,” on page 30**.

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

```
<js-install-4.0>
```

10.4 Configuring Buildomatic for Your Database and Application Server

This upgrade procedure is based on using the “buildomatic” scripts which are included with the WAR File Distribution ZIP release package. The buildomatic scripts are based on the `ant` utility and require the Java Development Kit (JDK) to run. If you don’t have Java available in your environment, see section **9.9.1, “Java Path for Buildomatic When Upgrading,” on page 87**.

Follow the configuration steps that match your database and application server in section **5.6, “Configuring the Buildomatic Scripts,” on page 32**. Below is an example configuration using the MySQL database.

10.4.1 Example Buildomatic Configuration (using MySQL)

All upgrade configuration is handled by a single file that is named `default_master.properties`. Jaspersoft provides a sample configuration file for each database. You must specify your database credentials and your application server location, and rename the file to `default_master.properties`.

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

You must rename and copy the sample file to this location: `<js-install-4.0>/buildomatic`.

1. Locate the `mysql_master.properties` file:

Database	Master Properties File
MySQL	<js-install-4.0>/buildomatic/sample_conf/mysql_master.properties

2. Copy the file to `<js-install-4.0>/buildomatic`.
3. Rename the file to `default_master.properties`.
4. Edit `default_master.properties` for your database and application server:

Database	Sample Property Values

MySQL	`appServerType=tomcat6 (or tomcat5, jboss, glassfish)`
	`appServerDir=c:\\apache-tomcat-6.0.26 (for example)`
	`dbUsername=root`
	`dbPassword=password`
	`dbHost=localhost`

10.5 Upgrading to JasperReports Server 4.0

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

Make sure you have backed up your `jasperserver` database before proceeding.
Make sure you have backed up your JasperServer 3.7 WAR file before proceeding.

Commands	Description
`cd <js-install-4.0>/buildomatic`	
`js-upgrade-samedb.bat`	Upgrade jasperserver-pro war file, in-place upgrade the jasperserver database (for Windows)
`js-upgrade-samedb.sh`	Upgrade jasperserver-pro war file, in-place upgrade the jasperserver database (for Linux)

Your JasperReports Server instance has now been upgraded to 4.0. If there are problems on startup or login refer to troubleshooting section **A.2, "Database Connectivity Errors," on page 116**.

10.6 Starting JasperReports Server 4.0

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

10.7 Logging Into JasperReports Server 4.0

If your application server and JasperReports Server 4.0were started cleanly, you can now prepare to login.

10.7.1 Clearing Your Browser Cache

Before you log into 4.0, make sure and clear your Browser cache. JavaScript files, which enable UI elements of JasperReports Server, are typically cached by the Browser. The cache should be cleared to ensure that the most current files are used.

For JasperReports Server 4.0, the UI has been significantly enhanced. It will be very important to clear the browser cache.

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

10.7.2 Logging Into JasperReports Server

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

10.8 Additional Information on Post-Upgrade Steps

There is additional information on optional steps that can be done after the main upgrade steps are complete. See section **9.9, "Additional Notes on JasperReports Server Upgrade," on page 87**.

10.8.1 Clearing the Repository Cache Table

In the jasperserver database, compiled JasperReports Library resources are cached in the `JIRepositoryCache` table for increased efficiency at runtime. In some cases, you may encounter errors running reports after an upgrade. Because the JasperReports Library JAR is typically updated with each new JasperReports Server release, old cached items can get out of date and thus cause errors at runtime. If you encounter errors that mention a JasperReports Library "local class incompatible," you should check your repository cache table. In summary, you can clear your jasperserver database cache table whether there are errors or not as part of this upgrade process.

To manually clear this table, run a SQL command similar to the following:

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

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

10.9 Running Buildomatic DB Upgrade Steps Manually

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

Commands	Description
`cd <js-install-4.0>/buildomatic`	
`js-ant upgrade-3.7-4.0-pro`	Execute SQL script for database upgrade to 4.0.
`js-ant import-minimal-pro`	Loads themes and other core resources for 4.0.
`js-ant import-sample-data-upgrade-pro`	(Optional) This step is optional. Loads the 4.0 sample data.
`js-ant deploy-webapp-pro`	Delete existing 3.7 war file, deploy 4.0 war file.

CHAPTER 11 UPGRADE NOTES FOR JASPERSERVER 3.5

If you are upgrading from an older JasperServer version such as 3.5 then you must first upgrade to JasperServer 3.7 before upgrading to 4.0.

Upgrading directly from JasperServer 3.5 to 4.0 is not a “certified” (i.e. fully QA tested) procedure.

The steps to carry out this operation are fully documented in the *JasperReports Server Installation Guide* for the 3.7 release. You will need to download the JasperServer 3.7 release package to get the relevant files and documentation.

To download the JasperServer 3.7 WAR file distribution zip package, go to the Jaspersoft Technical Support product downloads area. Or contact Technical Support or your sales representative.

The product name has changed slightly in 4.0. It is now known as JasperReports Server 4.0.

Chapter 12 Upgrading From Community Project

If you are currently running a Community Edition (CE) instance of JasperReports Server 4.0 and you would like to upgrade to JasperReports Server Professional or Enterprise, you may follow the instructions in this chapter. The upgrade steps are the same for Professional and Enterprise, so for simplicity we will use the term Professional for this upgrade section.

The steps in this section will use the JasperReports Server Professional WAR File Distribution release package and the included buildomatic scripts to carry out the upgrade.

This CE to Pro upgrade procedure is only valid for upgrade within a major JasperReports Server release. For instance, 4.0.x CE to 4.0.x Professional.

This chapter contains the following sections:

- **General Procedure**
- **Backing Up Your JasperReports Server CE Instance**
- **Exporting Your CE Repository Data**
- **Preparing the JasperReports Server Pro 4.0 WAR File Distribution**
- **Configuring Buildomatic for Your Database and Application Server**
- **Upgrading to JasperReports Server Pro**
- **Starting JasperReports Server Pro**
- **Logging Into JasperReports Server Pro**
- **Additional CE to Pro Upgrade Notes**

12.1 General Procedure

In order to carry out the upgrade procedure, you will be doing the following main steps:

1. Back up your JasperReports Server CE instance.
2. Export your CE repository data.
3. Upgrade your instance to JasperReports Server Professional.
4. Import your CE repository data.

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

12.2 Backing Up Your JasperReports Server CE Instance

First you must backup your JasperReports Server CE WAR file and your jasperserver database so that they can be restored in case there is a problem with the upgrade. These steps are performed from the command line in a Windows or Linux shell.

The following instructions are for the Tomcat application server and the MySQL database. For other application servers you should do steps similar to what is described below. For other databases, consult your DB administration documentation for back up information.

12.2.1 Backing Up Your JasperReports Server CE WAR File

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

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

12.2.2 Backing Up Your JasperReports Server Database

Go to the location where you originally unpacked your CE WAR file distribution zip.

1. Go to the <js-install-ce> directory.
2. Run the following command:

Windows:

```
mysqldump --user=root --password=<password> jasperserver > js-db-ce-dump.sql
```

Linux:

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

If you installed JasperReports Server CE from the installer, you may specify `--user=jasperdb` in this command.

If you receive an error about packet size, see section **A.2.6, “Maximum Packet Size in MySQL,” on page 117**.

12.3 Exporting Your CE Repository Data

You should first check to see that you have the following file in place in your buildomatic directory: default_master.properties. This file holds the settings that are specific to your JasperReports Server instance. Such as you application server type and location, and your database type and location.

You should see this file in the following location:

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

If you do not have a `default_master.properties` file in place, then you should go to the following section for instructions on how to set up this file. See section **5.6.1, “Creating your Default Master Properties File,” on page 32**.

To export your CE repository data:

1. `cd <js-install-ce>/buildomatic`
2. Run buildomatic with the export target:

Windows:

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

Linux:

```
./js-ant export-everything-ce -DexportFile=js-ce-export.zip
```

This operation uses the export option `--everything` which preserves all your repository data.

Remember the path to your exported file, because you will need to specify it when you import into JasperReports Server Professional.

12.4 Preparing the JasperReports Server Pro 4.0 WAR File Distribution

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

- Follow steps in section **5.2, "Obtaining the WAR File Distribution Zip," on page 30**.
- Follow steps in section **5.3, "Unpacking the WAR File Distribution Zip," on page 30**.

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

```
<js-install-pro>
```

12.5 Configuring Buildomatic for Your Database and Application Server

This upgrade procedure is based on using the buildomatic scripts which are included with the WAR File Distribution ZIP release package. The buildomatic scripts are based on the `ant` utility and require the Java Development Kit (JDK) to run. If you don't have Java available in your environment, see section **9.9.1, "Java Path for Buildomatic When Upgrading," on page 87**.

Follow the configuration steps that match your database and application server in section **5.6, "Configuring the Buildomatic Scripts," on page 32**. Below is an example configuration using the MySQL database.

12.5.1 Example Buildomatic Configuration (using MySQL)

All upgrade configuration is handled by a single file which will be named `default_master.properties`. You will need to specify your database credentials and your application server location.

There is a sample configuration file for each database. This procedure uses MySQL as an example (the same general logic applies to other databases).

The sample file will be renamed and copied to the following location: `<js-install-pro>/buildomatic`

1. Copy: `mysql_master.properties`

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

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

Database	Sample Property Values

MySQL	`appServerType=tomcat6 (or tomcat5, jboss, glassfish)`
	`appServerDir=c:\\apache-tomcat-6.0.26 (for example)`
	`dbUsername=root`
	`dbPassword=password`
	`dbHost=localhost`

12.6 Upgrading to JasperReports Server Pro

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

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

Commands	Description
`cd <js-install-pro>/buildomatic`	

`js-ant drop-js-db`
`js-ant create-js-db`
`js-ant init-js-db-pro`
	This will delete your jasperserver db. Make sure it is backed up.

`js-ant import-minimal-for-upgrade-pro`	Creates superuser and default tenant structure.
`js-ant import-upgrade -DimportFile="<path-and-filename>"`	The -DimportFile argument should point to the js-ce-export.zip file you created earlier. On Windows, you must use double quotes (`"`) if your path or filename contains spaces. On Linux, you must use double quotes escaped with a backslash (`\"`) in this case.
`js-ant import-minimal-pro`	Load resources needed by JasperReports Server pro.
`js-ant import-sample-data-upgrade-pro`	(Optional) This step is optional. Loads the pro sample data.
`js-ant deploy-webapp-ce-to-pro`	Delete CE war file, deploy pro war file.

On MySQL, if you receive an error about packet size, see section **A.2.6, "Maximum Packet Size in MySQL," on page 117**.

12.7 Starting JasperReports Server Pro

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

12.8 Logging Into JasperReports Server Pro

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

12.8.1 Clearing Your Browser Cache

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

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

12.8.2 Logging Into JasperReports Server Pro

Login using the following URL, user IDs, and passwords:

URL: http://localhost:8080/jasperserver-pro

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

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

Your JasperReports Server instance has now been upgraded from Community Edition (CE) to Professional. If there are problems on startup or login refer to troubleshooting section **A.2, "Database Connectivity Errors," on page 116**.

12.9 Additional CE to Pro Upgrade Notes

This section has additional notes for both before upgrade and after upgrade.

12.9.1 Notes on XML/A Configuration After Upgrade to Pro

XML/A connection definitions contain a username and password definition in order to make the Web Services connection to the server. The Professional edition of JasperReports Server supports multi-tenancy which allows multiple organizations to co-existing on a single instance. JasperReports Server users must belong to a specific organization (except for superuser). The default organization in JasperReports Server Professional is `organization_1`.

Therefore after upgrade, any XML/A connections that you have will need to be updated to include the Organization that the user belongs to. After the upgrade operation, users will belong to the "default" organization that is part of the core data setup in the Pro edition.

In addition, the XML/A connection specifies an instance URI. This URI will need to be updated to point to a Pro URI.

You should edit your XML/A connections to specify the following:

- Change user IDs from, for example:

  "`jasperadmin`" to "`jasperadmin|organization_1`"

  "`joeuser`" to "`joeuser|organization_1`"

- Change the URI value from, for instance:

  ```
  http://localhost:8080/jasperserver/xmla
  ```

- To:

  ```
  http://localhost:8080/jasperserver-pro/xmla
  ```

Chapter 13 Changing Password Encryption in JasperReports Server

By default, password encryption is enabled in JasperReports Server and passwords are stored as cipher text in the database. System administrators can change the encryption algorithm, as well as specify the salt key used to initiate the encryption algorithm.

This chapter describes the procedure to enable password encryption if you have a JasperReports Server instance without encryption turned on. For more information on encryption options refer to the *JasperReports Server Administrator Guide*.

This chapter contains the following sections:

- **Backing Up Your JasperReports Server Database**
- **Stopping Your Application Server**
- **Running the Repository Export Utility**
- **Specifying Encryption Settings in the JasperReports Server WAR**
- **Specifying Encryption Settings for the Import Utility**
- **Recreating the JasperReports Server Database**
- **Importing Your Repository Data**
- **Starting the Application Server**
- **Logging Into JasperReports Server**

13.1 Backing Up Your JasperReports Server Database

As a precaution, you must back up your jasperserver database in case there is any problem while enabling encryption.

To back up the default MySQL database, go to the <js-install> directory and run the following command:

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

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

If you installed JasperReports Server from the installer, you may specify `--user=jasperdb` in this command. If you receive an error about packet size, see section **A.2.6, “Maximum Packet Size in MySQL,” on page 117**.

For Oracle, Microsoft SQL Server, and PostgreSQL databases, refer to your product documentation for details.

13.2 Stopping Your Application Server

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

13.3 Running the Repository Export Utility

The repository export utility writes out all of the JasperReports Server repository objects to a set of XML and binary format files. The output of the export operation is known as an export catalog.

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

```
Windows: js-export.bat --everything --output-dir js-backup-catalog
Linux:   js-export.sh --everything --output-dir js-backup-catalog
```

For information on running the export utility, refer to **Chapter 14, "Configuring the Import-Export Utilities," on page 109**.

13.4 Specifying Encryption Settings in the JasperReports Server WAR

JasperReports Server uses the Spring configuration and security to enable and configure encryption. These options can allow you to have a strong encryption setup. This section is focused on the minimal configuration necessary for enabling encryption. For more specific information in the *JasperReports Server Administrator Guide* about the security algorithms and settings.

1. Open the following file for editing:

   <tomcat>/jasperserver-pro/WEB-INF/ApplicationContext-security.xml

2. In the definition of the `daoAuthenticationProvider` bean, there is a commented-out reference to the `passwordEncoder` bean. Look for the section of the XML file that starts with:

   ```
   <bean id="daoAuthenticationProvider"
   ```

   In this bean definition, uncomment the reference to passwordEncoder. This causes the passwordEncoder logic to be used. After removing the commenting characters the line should look like the following:

   ```
   <property name="passwordEncoder"><ref local="passwordEncoder"/></property>
   ```

3. Enable encryption in the `passwordEncoder` bean by modifying the `allowEncoding` property. Change the value from `false` to `true` so that it looks like the following:

   ```
   <property name="allowEncoding"><value>true</value></property>
   ```

4. If the default DESede algorithm is used, the `secretKey` represents the salt key and must be 24 characters. By default, the `keyInPlainText` property is true, meaning the key can be in plain text to make it easier to enter, for example:

   ```
   <property name="keyInPlainText"><value>true</value></property>
   <property name="secretKey"><value>jaspersoftInSanFrancisco</value></property>
   ```

   The text `jaspersoftInSanFrancisco` is 24 characters long, therefore the two properties above work with their default values. However, for better security, we recommend that they be changed.

5. The last two properties may be left unchanged. They are set to DESede by default. The default values are the following:

   ```
   <property name="secretKeyAlgorithm"><value>DESede</value></property>
   <property name="cipherTransformation"><value>DESede/CBC/PKCS5Padding</value></property>
   ```

   As described in the *JasperReports Server Administrator Guide*, the `secretKey`, `secretKeyAlgorithm`, and `cipherTransformation` property settings must be consistent with each other. For example, different algorithms expect different key lengths.

6. Save and close the file. Encryption is now enabled for the JasperReports Server application upon the next restart.

13.4.1 Specifying Encryption Settings - Reference Table

The information in the table below is a summary of the options described in the section above.

The following table describes the available password encryption configuration options:

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

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

13.5 Specifying Encryption Settings for the Import Utility

Before starting JasperReports Server, you must convert the plain text passwords that are currently stored in the repository export catalog that you created in section **13.1, "Backing Up Your JasperReports Server Database," on page 103**. These plain-text passwords need to be converted to cipher text and reloaded into the database in order to successfully login after the server restarts. To do this, you must add the same encryption settings to the configuration file that is used by the import and export utilities.

1. Open the following configuration file for editing:

   <js-install>/buildomatic/conf_source/iePro/applicationContext-security.xml

2. This file contains the `passwordEncoder` bean definition, the same as in the JasperReports Server WAR, only by itself. Perform the same modifications to this file as in the procedure in section **13.4, "Specifying Encryption Settings in the JasperReports Server WAR," on page 104**.

13.6 Recreating the JasperReports Server Database

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

13.6.1 Dropping and Recreating in MySQL

1. Change directory to <js-install>/buildomatic/install_resources/sql/mysql.
2. Log into your MySQL client:

```
mysql -u root -p
```

3. Drop the `jasperserver` database, create a new one, and load the `jasperserver` schema:

```
mysql>drop database jasperserver;
mysql>create database jasperserver character set utf8;
mysql>use jasperserver;
mysql>source js-pro-create.ddl;
mysql>source quartz.ddl;
```

13.6.2 Dropping and Recreating in Oracle

1. Change directory to <js-install>/buildomatic/install_resources/sql/oracle.
2. Log into your SQLPlus client:

```
sqlplus jasperadmin/password@MY_SID
```

3. Drop the `jasperserver` database, create a new one, and load the `jasperserver` schema:

```
SQL> drop tablespace jasperserver including contents and datafiles;
SQL> create tablespace jasperserver datafile 'jasperserver.dbf' size 100m reuse \
autoextend on next 50m maxsize 1000m extent management local autoallocate \
segment space management auto;
SQL> connect jasperadmin/password@MY_SID
SQL> @js-pro-create.ddl
SQL> @quartz.ddl
```

13.6.3 Dropping and Recreating in Microsoft SQL Server

1. Change directory to <js-install>/buildomatic/install_resources/sql/sqlserver.
2. Drop the `jasperserver` database, create a new one, and load the `jasperserver` schema:

```
sqlcmd -S localhost\jasperserver -Q "DROP DATABASE jasperserver"
sqlcmd -S localhost\jasperserver -Q "CREATE DATABASE jasperserver"
sqlcmd -S localhost\jasperserver -d jasperserver -i js-pro-create.ddl
sqlcmd -S localhost\jasperserver -d jasperserver -i quartz.ddl
```

13.6.4 Dropping and Recreating in PostgreSQL

1. Change directory to <js-install>/buildomatic/install_resources/sql/postgresql
2. Start `psql` using an administrator account such as `postgres`:

```
psql -U postgres
```

3. Drop the `jasperserver` database, create a new one, and load the `jasperserver` schema:

```
drop database jasperserver;
create database jasperserver encoding='utf8';
\c jasperserver
\i js-pro-create.ddl
\i quartz.ddl
```

13.7 Importing Your Repository Data

The import utility reloads all of your repository data. As the data is being saved to the repository, the password fields that were plain text are encrypted using the encryption settings you made in the sections above.

To import your backup catalog to the repository:

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

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

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

For information on running the import utility, see **Chapter 14, "Configuring the Import-Export Utilities," on page 109**.

13.8 Starting the Application Server

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

13.9 Logging Into JasperReports Server

You can now log into JasperReports Server.

Enter your user ID and password in the same manner as you did before encryption was turned on. You can check the contents of the `JIUser` table in the `jasperserver` database and examine the password column to see that the password is no longer stored in plain text.

Chapter 14 Configuring the Import-Export Utilities

The import and export utilities let you add resources to or extract resources from the JasperReports Server repository. Typically, users export data from their previous instance and import it into their new installation when upgrading JasperReports Server. The import utility is also used at installation time in order to load the sample data into the repository.

Please refer to the *JasperReports Server Administrator Guide* for more information on command options for the import and export utilities.

This chapter contains the following sections:

14.1 Introduction

The import-export functionality can be run using the auto-configured buildomatic Ant scripts or it can be run using the shell based js-import.sh/bat and js-export.sh/bat scripts.

The different command styles would look like the following:

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

These two ways of running import-export commands are being merged so that all database configuration work will be done automatically by the buildomatic/default_master.properties. However, as of JasperReport Server version 4.0, this merging work is not complete. This means that if you are running the js-import.sh/bat or js-export.sh/bat shell scripts, you might have to do manual configuration depending on the database used.

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

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

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

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

Known issue in Release 4.0: After an import operation, if the import included permission changes to files or folders, you should clear your application server temp folder because there are cached permissions which could interfere with the new permissions. For Tomcat, the location to clear is <tomcat>/temp.

14.2 Import-Export Configuration Files

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

File or Location	Purpose
<js-install>/buildomatic/js-import.bat and .sh	Import scripts for Windows and Linux, respectively
<js-install>/buildomatic/js-export.bat and .sh	Export scripts for Windows and Linux, respectively
<js-install>/buildomatic/default_master.properties	File that you must edit (already in place if you installed from the binary installer)
<js-install>/buildomatic/build_conf/default/ js.jdbc.properties	Database and hibernate dialect settings file (put in place after you run "js-ant gen-config")
<js-install>/buildomatic/conf_source/iePro/ applicationContext-*.xml	Spring configuration files
<js-install>/buildomatic/conf_source/iePro/lib	All of the JasperReports Server jar files and the JDBC driver location

14.3 Changing Your Configuration Settings

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

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

14.3.1 First Create a default_master.properties File

If you don't have a <js-install>/buildomatic/default_master.properites file in place then you should create one. For instance, copy and rename buildomatic/sample_conf/mysql_master.properties to buildomatic/default_master.properties. Then edit default_master.properties for you local settings. For more information see section **5.6.1, "Creating your Default Master Properties File," on page 32.**

Do the following:

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

14.3.2 Here are the Configuration Locations

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

- <js-install>/buildomatic/build_conf/default/js.jdbc.properties
- <js-install>/buildomatic/build_conf/default/js.quartz.properties (only for DB2 and PostgreSQL)

<js-install>/buildomatic/conf_source/iePro/lib:

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

14.3.3 Check your js.jdbc.properties

The following tables gives sample settings for each database.

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

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

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

If your repository contains international characters, you may need to perform additional configuration for the import and export utilities. See section **A.10, “Exporting a Repository That Contains UTF-8,” on page 119**.

Table 14-1 JDBC Settings in the js.jdbc.properties File

Database	Sample Property Values

MySQL	metadata.hibernate.dialect=org.hibernate.dialect.MySQLDialect
	metadata.jdbc.driverClassName=com.mysql.jdbc.Driver
	metadata.jdbc.url=jdbc:mysql://**localhost:3306**/jasperserver?useUnicode=true&characterEncoding=UTF-8
	metadata.jdbc.username=root
	metadata.jdbc.password=**password** *or*
	metadata.jdbc.encryptedPassword=**encrypted-password**

Oracle	metadata.hibernate.dialect=com.jaspersoft.ji.hibernate.dialect.OracleUnicodeDialect
	metadata.jdbc.driverClassName=oracle.jdbc.OracleDriver
	metadata.jdbc.url=jdbc:oracle:thin:@**localhost:1521:orcl**
	metadata.jdbc.username=jasperadmin
	metadata.jdbc.password=**password** *or*
	metadata.jdbc.encryptedPassword=**encrypted-password**

DB2	metadata.hibernate.default_schema=**JSPRSRVR**
	metadata.hibernate.dialect=org.hibernate.dialect.DB2SQLDialect
	metadata.jdbc.driverClassName=com.ibm.db2.jcc.DB2Driver
	metadata.jdbc.url=jdbc:db2://localhost:50000/jsprsrvr:driverType=4;fullyMaterializeLobData=true;fullyMaterializeInputStreams=true;progressiveStreaming=2;progresssiveLocators=2;currentSchema=**JSPRSRVR**;
	metadata.jdbc.username=db2admin
	metadata.jdbc.password=**password** *or*
	metadata.jdbc.encryptedPassword=**encrypted-password**
	metadata.jndi=jdbc/jasperserver
	metadata.upperCaseNames=true

Table 14-1 JDBC Settings in the js.jdbc.properties File, continued

Database	Sample Property Values

SQL Server	`metadata.hibernate.dialect=org.hibernate.dialect.SQLServerDialect`
	`metadata.jdbc.driverClassName=`
	`com.microsoft.sqlserver.jdbc.SQLServerDriver`
	`metadata.jdbc.url= jdbc:sqlserver://`**`localhost:1433;`**
	`databaseName=jasperserver;SelectMethod=cursor`
	`metadata.jdbc.username=sa`
	`metadata.jdbc.password=`**`sa`** *or*
	`metadata.jdbc.encryptedPassword=`**`encrypted-sa`**

PostgreSQL	`metadata.hibernate.dialect=`
	`com.jaspersoft.hibernate.dialect.PostgresqlNoBlobDialect`
	`metadata.jdbc.driverClassName=org.postgresql.Driver`
	`metadata.jdbc.url=jdbc:postgresql://`**`localhost:5432`**`/jasperserver`
	`metadata.jdbc.username=`**`postgres`**
	`metadata.jdbc.password=`**`postgres`** *or*
	`metadata.jdbc.encryptedPassword=`**`encrypted-postgres`**

14.3.4 If DB2 or PostgreSQL Check your js.quartz.properties

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

<js-install>/buildomatic/build_conf/default

Table 14-2 Quartz Settings in the js.quartz.properties File

Database	Sample Property Values

DB2	`quartz.delegateClass=org.quartz.impl.jdbcjobstore.DB2v8Delegate`
	`quartz.tablePrefix=JSPRSRVR.QRTZ_`

PostgreSQL	`quartz.delegateClass=org.quartz.impl.jdbcjobstore.PostgreSQLDelegate`
	`quartz.tablePrefix=QRTZ_`

14.4 Deploying a Database Driver

In order for the import-export shell scripts to run, they will need the proper JDBC driver. This allows a connection to be made to the JasperReports Server repository database.

Put the appropriate JDBC driver JAR into the following directory:

<js-install>/buildomatic/conf_source/iePro/lib

All Jaspersoft distributed JDBC drivers can be found at this location:

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

14.5 Running Import or Export

To see that the import and export shell scripts are properly configured, you can run the scripts using the `--help` option (with two dashes `--`) that displays the command options:

```
Windows: js-import.bat --help
         js-export.bat --help
Linux:   js-import.sh --help
         js-export.sh --help
```

If your repository contains international characters, you may need to perform additional configuration for the import and export utilities. See section **A.10, “Exporting a Repository That Contains UTF-8,” on page 119**.

For complete information on the standard import-export options refer to the *JasperReports Server Administrator Guide*.

14.5.1 Import-Export Updates for 4.0

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

If you have the Pro feature Auditing turned on, JasperReports Server will generate audit information which will be stored in the repository database. It is possible for a large number of audit resources to be generated.

As of 4.0, audit resources will not be imported or exported without specifically requesting themas part of the import or export operation.

If you are upgrading from 3.7 to 4.0, using the standard procedures, the audit event resources will not be included, by default, on the import into 4.0. This will allow your import to 4.0 operation to run much faster in the case where there are a large number of access resources. To include these access events, set `includeAccessEvents=true` in your `default_master.properties` file.

14.6 Configuring the Import-Export Utility for JasperServer 3.7

You may need to configure your 3.7 import-export utility as part of the upgrade to 4.0 process.

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

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

<js-install-3.7>/scripts/config

<js-install-3.7>/scripts/lib

To configure import-export for your database type and/or to handle database password changes you have made in your system, you would update the following files in the same manner as is described in the sections above:

<js-install-3.7>/scripts/config/js.jdbc.properties

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

<js-install-3.7>/scripts/lib

Appendix A Troubleshooting

This appendix contains the following sections:

A.1 Installer Freezes

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

You can find the installer log in the following locations:

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

If you have tried multiple installs, make sure you view the most recent install log file.

A.2 Database Connectivity Errors

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

A.2.1 Testing the Database Connection

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

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

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

A.2.1.1 Logging Into MySQL

Start MySQL from the command line and try to log in directly using the `jasperdb` user, for example:

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

or

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

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

A.2.1.2 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.2.1.3 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.2.2 Configuration File Locations

JasperReports Server configuration properties are found in the following files, according to your application server:

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
JBoss:	<jboss>/server/default/deploy/js-mysql-ds.xml *or* js-oracle-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.2.3 Special Configuration Case under Tomcat

If you installed JasperReports Server using the WAR file distribution file and handled the steps manually, Tomcat may have an additional (confusing) database configuration.

The special case occurs when you have deployed the jasperserver.war file into the Tomcat webapps directory. Valid JasperReports Server WAR deployments can be based on a single file (jasperserver-pro.war) or an “unpacked” WAR file directory (jasperserver-pro directory).

If you use a single WAR file for deployment under Tomcat, Tomcat takes the following steps (for instance in Tomcat 5.5):

- Unpack the jasperserver-pro.war file into a new directory named jasperserver-pro.
- Take the jasperserver-pro/META-INF/context.xml file and copy it to a new file:

  <tomcat>/conf/Catalina/Localhost/jasperserver-pro.xml

This database configuration in <tomcat>/conf tree overrides the context.xml found in your jasperserver directory. If you are having database trouble in this scenario, it is recommended that you keep things simple by:

1. Deleting your <tomcat>/webapps/jasperserver-pro.war file. This causes the jasperserver directory to be used.
2. Deleting your <tomcat>/Catalina/Localhost/jasperserver-pro.xml. This causes the META-INF/context.xml from your jasperserver-pro directory to be used.

A.2.4 Connect to Installed/Bundled Version of MySQL

These steps are for connecting under Linux.

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

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

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

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

```
... /home/devuser/jasperreports-server-4.0/mysql/bin/mysqld_safe --port=3306 \
--socket=/home/devuser/jasperreports-server-4.0/mysql/tmp/mysql.sock ...
```

3. Then run a command similar to the following:

```
/home/devuser/jasperreports-server-4.0/mysql/bin/mysql -u jasperdb -p \
--socket=/home/devuser/jasperreports-server-4.0/mysql/tmp/mysql.sock
```

A.2.5 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.2.6 Maximum Packet Size in MySQL

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

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

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

```
max_allowed_packet = 4M
```

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

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

A.3 License Not Found Error

Normally, the JasperReports Server installer includes an evaluation license file that you replace with a commercial license file, as described in section **2.2.3, "Installing a New License File," on page 21**. If JasperReports Server returns an error after you replace the license file, make sure you did the following:

- Cleared your application server's work directory, as explained in section **2.2.3**.
- Set the `Djs.license.directory` property in your application server startup environment properly:

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

  For example, in Linux the command might be:

  ```
  -Djs.license.directory="/opt/jasperserver-pro-4.0
  ```

  The specified directory must contain the license file, which should be named jasperserver.license. The property is typically set for your application server in the environment startup script. It must give 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

  For additional information about licenses, see section **5.8, "Setting Up the JasperReports Server License," on page 35**.

A.4 License Not Found 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 section **2.2.4, "License File for Existing Tomcat as Windows Service," on page 22**.

A.5 Error Running a Report

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

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

1. Log into JasperReports Server as a user with administrative permissions and locate the report unit that returns errors.
2. Select the report and click the **Edit** button in the toolbar to identify the data source the report uses. The data source name is found on the fourth edit page.
3. Select this data source in the repository and click the **Edit** button in the toolbar.
4. Review the information specified for this data source.
5. Click the **Test Connection** button in order to validate the connection.
6. Click **Save** or **Cancel** when you are done.
7. Test your report. If it still returns errors, edit the data source again and try checking other values, such as the port used by the database.

A.6 Database Error after Changing MySQL Port Number

The default port for MySQL is 3306. If you entered a different port when you installed MySQL, the JasperReports Server installer configures them to communicate properly. If the MySQL port number has changed, or if you encounter a problem, check the database configuration files to verify your port number.

If it is incorrect, change it to the correct port number, save the file, and restart the application server. For more information, see section **A.2.2, "Configuration File Locations," on page 116**.

A.7 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 Java Out of Memory Error

If you encounter a Java out of memory error, it is suggested that you increase your Java heap size setting. See section **5.9, "Setting Java JVM Options," on page 35**. It is recommended that you add `-Xms128m -Xmx512m` to your `JAVA_OPTS` setting, but you may increase that to `-Xms512m -Xmx1024m` or larger if your server can support higher settings.

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

A.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.10 Exporting a Repository That Contains UTF-8

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

A.10.1 Error During JasperServer 1.2 Export

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

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

it may be due to an incorrect character in the file scripts/ji-export-util/jdbc.properties. Check the URL in this file; it should look like the following:

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

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

A.10.2 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 you repository database on an Oracle RDBMS, modify the last line of both <js-install>/scripts/js-export.* files as follows:

From: `java -classpath ...`

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

A.11 Importing Scheduled Jobs with Update Option

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

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

A.12 JBoss Modifications

A.12.1 JBoss 4.2 XML/A Connection Fix

JBoss 4.2 includes the JBossWS service as a standard, default feature. JasperReports Server has web services support for XML/A connections. The web services classes in JasperReports Server and JBoss can conflict and cause the following error when attempting to utilize a JasperReports Server XML/A connection:

```
javax.xml.soap.SOAPException: Unable to create message factory for
SOAP: org.jboss.ws.core.soap.MessageFactoryImpl
```

To prevent the web services class conflict, set the special Java JVM options for JBoss 4.2, as described in section **6.1, "Setting JVM Options for Application Servers," on page 43**.

A.12.2 JBoss Large INFO Log Message on Analysis Drill-through

JBoss has an internal mechanism to track and log information on unclosed JDBC connections. JasperServer Analysis leaves a connection open for performance reasons when doing an analysis drill-through. In this case, JBoss puts a large INFO level message into the server.log. To silence this INFO message

1. Open the JBoss log4j configuration file for editing:

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

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

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

A.12.3 JBoss 4.0 Log4j Error on Startup

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

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

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

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

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

A.12.4 JBoss 5.0.1 and 5.1.x Error

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

```
org.jboss.xb.binding.JBossXBRuntimeException: Failed to create a new SAX parser
Caused by: java.lang.ClassCastException
```

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

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

When running the buildomatic scripts to deploy to JBoss, the xercesImpl-3.7.jar file is automatically deleted in order to fix this problem.

A.13 WebSphere: 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 Acegi framework for authentication and it is mapped using a filter chain with the `/*` url pattern.

WebSphere provides a special property to solve this problem:

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.14 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

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

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

A.15 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.15.1 Missing Java JDK

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

```
[exec] [ERROR] BUILD FAILURE
[exec] [INFO] ---------------------------------------------------
[exec] [INFO] Compilation failure
[exec] Unable to locate the Javac Compiler in:
[exec]   c:\Program Files\Java\jdk1.6.0_10\jre\..\lib\tools.jar
[exec] Please ensure you are using JDK 1.5 or above and
[exec] not a JRE (the com.sun.tools.javac.Main class is required).
[exec] In most cases you can change the location of your Java
[exec] installation by setting the JAVA_HOME environment variable.
```

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

If you are upgrading JasperReports Server, you can also use the Java 1.5 JDK bundled in the previously installed version, as described in **9.9.1, "Java Path for Buildomatic When Upgrading," on page 87**.

A.15.2 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.15.3 Older Apache Ant Version

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

```
ant -version
```

A.16 Troubleshooting on Solaris

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

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

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

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

A.17 Disabling User Session Persistence in Application Servers

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

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

The errors appear in the JasperReports Server log when users log in after the application server has been restarted. The errors do not appear to users, and they have no impact on JasperReports Server operations.

Because JasperReports Server user sessions are not persistent, you can configure your application server to disable persistence and avoid the error. For example, in Apache-Tomcat 5.5 and 6.0, edit the file <tomcat>/conf/context.xml and locate the following lines:

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

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

A.18 Glassfish with PostgreSQL and Java 1.6

If you are running Glassfish with PostgreSQL under Java 1.6, you can get the following error:

```
java.lang.AbstractMethodError: org.postgresql.jdbc3.Jdbc3Connection.getClientInfo()Ljava/
util/Properties
```

Glassfish will need to be configured to not use Java 1.6 methods on the PostgreSQL JDBC driver.

You can run the following `asadmin` commands to get around this issue:

```
<glassfish>/bin/asadmin set
  domain.resources.jdbc-connection-pool.jasperserver_pool.property.JDBC30DataSource=true
<glassfish>/bin/asadmin set
  domain.resources.jdbc-connection-pool.sugarcrm_pool.property.JDBC30DataSource=true
<glassfish>/bin/asadmin set
  domain.resources.jdbc-connection-pool.foodmart_pool.property.JDBC30DataSource=true
```

The same can be done for any additional jdbc-connection-pools.