



JASPERREPORTS® SERVER INSTALLATION GUIDE

RELEASE 6.4

<http://www.jaspersoft.com>

Copyright ©2005-2017 TIBCO Software Inc. All Rights Reserved. TIBCO Software Inc.

This is version 0717-JSP64-46 of the *TIBCO JasperReports Server Installation Guide*.

TABLE OF CONTENTS

Chapter 1 Introduction	9
1.1 Conventions	10
1.2 Supported Platforms	11
1.3 Using IBM JDK 1.7	11
1.4 JasperReports Server Distributions	11
1.4.1 Installer Support	12
1.4.2 WAR File Binary Distribution Support	13
1.5 Release Notes	14
1.6 System Requirements	15
1.7 Support for Internationalization	15
Chapter 2 Installing JasperReports Server	17
2.1 Pre-installation Steps	17
2.2 Starting the Installer	18
2.3 Accepting the License Agreement	18
2.4 Choosing Installation Type	18
2.5 Choosing an Installation Directory	19
2.6 Selecting a Tomcat Configuration	19
2.7 Selecting a PostgreSQL Configuration	20
2.7.1 Choosing the Bundled PostgreSQL	20
2.7.2 Choosing an Existing PostgreSQL on a Local Host	20
2.7.3 Using an Existing PostgreSQL on a Remote Host	21
2.7.4 Enabling Connections to a Remote Host	22
2.8 Installing Sample Data	22
2.9 Completing the Installation	22
2.10 Post-installation Steps	23
2.10.1 Updates Made by the Installer During Installation	23
2.10.2 Installer Output Log File Location	24
2.10.3 Setting your Java JVM Options	24
2.10.4 Installing a New License File	24
2.10.5 License File for Existing Tomcat as Windows Service	24
2.11 Starting and Stopping the Server	25
2.11.1 Start/Stop Menu — Windows	25

2.11.2 Start/Stop Scripts — Linux	26
2.11.3 Start/Stop Apps — Mac OSX	27
2.12 Logging into JasperReports Server	28
2.13 Log Files	29
2.14 Uninstalling the Server	29
2.14.1 Windows	29
2.14.2 Linux	29
2.14.3 Mac OSX	30
2.14.4 Uninstall Survey	30
Chapter 3 Installing the WAR File Distribution	31
3.1 Applications Supported by the WAR File Distribution	31
3.1.1 Database and Application Server Support	31
3.1.2 Operating System Support for Bash Shell	32
3.2 Installing the WAR File Using js-install Scripts	32
3.3 Additional Steps for Using DB2 and js-install Scripts	35
3.4 Additional Steps for Using JBoss EAP or Wildfly	36
3.5 Starting the Server	37
3.6 Logging into the Server	37
3.6.1 JasperReports Server Heartbeat	37
3.7 Troubleshooting Your Server Configuration	38
3.7.1 Startup Problems	38
3.7.2 Error Running a Report	38
3.7.3 Error Running js-install Scripts (js-install.bat/sh)	38
3.7.4 Problem Connecting to a Cloud Database Instance	39
3.8 Installing the WAR File Manually	39
Chapter 4 JVM Options, License Setup, Working with JDBC Drivers	43
4.1 Setting JVM Options for Application Servers	43
4.1.1 Tomcat and JBoss JVM Options	43
4.1.2 Changing JVM Options for Tomcat as a Windows Service	45
4.1.3 Changing JVM Options for Bundled Tomcat on Linux	46
4.1.4 Changing GlassFish JVM Options	46
4.2 Setting Up the JasperReports Server License	48
4.2.1 Default License Configuration for All Application Servers	48
4.2.2 User-Defined License Location	48
4.3 Working With JDBC Drivers	50
4.3.1 Open Source JDBC Drivers	50
4.3.2 Commercial JDBC Drivers	51
4.3.3 Working with Oracle RAC	54
4.3.4 Application Server Copy-to Locations	56
4.4 Locating and Changing Buildomatic Configuration Files	56
4.4.1 Regenerating Buildomatic Settings	56
4.4.2 Locating Buildomatic-Generated Property Files	56
4.4.3 Buildomatic Location for JasperReports Server WAR File	57
4.4.4 Buildomatic Location for SQL Scripts	57
4.4.5 Buildomatic Location for Database Creation Scripts	58

4.4.6	Buildomatic Location for Sample Data Catalog ZIP Files	58
4.4.7	Hibernate Properties Settings	58
4.4.8	Database Connection Configuration Files	58
4.5	Configuring Report Scheduling	59
4.5.1	Mail Server Configuration Settings	60
4.5.2	Database Settings for the Quartz Driver Delegate Class	61
4.5.3	Settings for the Report Scheduler Web URI	61
4.5.4	Settings for the Quartz Table Prefix	62
4.5.5	Settings for Import-Export	62
4.5.6	Setting Properties in the default_master.properties File	63
4.6	Updating XML/A Connection Definitions	63
Chapter 5	Installing the WAR File for WebSphere	65
5.1	Procedure for Installing and Deploying the WAR File in WebSphere	65
5.1.1	Installing WebSphere and a Database	65
5.1.2	Preparing Server Files	66
5.1.3	Configuring CSRFGuard, Hibernate, and Quartz Settings	67
5.1.4	Configuring a JDBC Provider in WebSphere	68
5.1.5	Deploying the WAR File in WebSphere	72
5.1.6	Setting JVM Options	72
5.2	Configuring Other Database Connections	74
5.2.1	Defining a JNDI Name and Sample Data Sources for MySQL	74
5.2.2	Defining a JNDI Name and Sample Data Sources for DB2	75
5.2.3	Defining a JNDI Name and Sample Data Sources for Oracle	78
5.2.4	Defining a JNDI Name and Sample Data Sources for SQL Server	80
5.3	Starting and Restarting JasperReports Server	81
5.4	Logging into the Server	81
5.5	Configuring Report Scheduling	82
5.5.1	Additional Fix for Scheduled Report with JNDI Data Source	82
5.5.2	Additional Change for Mail Server Authentication	83
5.6	Updating XML/A Connection Definitions (Optional)	83
5.7	Troubleshooting your Configuration	83
5.7.1	Startup Problems	83
5.7.2	Error Running Report	83
5.7.3	Filter Error Using MySQL	83
5.7.4	Error Creating Internationalized Name	84
5.7.5	Xerces Error	84
5.7.6	OLAP View Fails With Exception	84
Chapter 6	Installing the WAR File for WebLogic	87
6.1	Procedure for Installing the WAR File for WebLogic	87
6.2	Setting Java Properties	95
6.3	Configuring Other Database Connections	96
6.3.1	Configuring TIBCO JDBC Driver Connections	96
6.3.2	Configuring Databases Using the Vendor's Driver	99
6.4	Starting the Server	101
6.5	Logging into the Server	101

6.6	Configuring Report Scheduling	101
6.7	Restarting the Server	102
6.8	Updating XML/A Connection Definitions (Optional)	102
6.9	Troubleshooting Your JasperReports Server Configuration	102
6.9.1	Startup Problems	102
6.9.2	Error Running Report	102
Appendix A	Troubleshooting	103
A.1	Binary Installer Freezes	103
A.1.1	Installer Log Files	103
A.1.2	Installer DebugTrace Mode	104
A.2	Error Running Buildomatic Scripts	104
A.2.1	Missing Java JDK	104
A.2.2	Forgot to Copy the File ant-contrib.jar	104
A.2.3	Failure with '\$' Character in Passwords in Buildomatic Scripts	105
A.2.4	Older Apache Ant Version	105
A.3	Unable to Edit Files on Windows 7	105
A.4	Bash Shell for Solaris, IBM AIX, HP UX and FreeBSD	105
A.5	Linux Installer Issue with Unknown Host Error	106
A.6	Installation Error with Windows Path	107
A.7	Mac OSX Issues	107
A.7.1	Problem Starting JasperReports Server on Mac	107
A.8	Database-related Problems	108
A.8.1	Database Privileges Required By JasperReports Server	108
A.8.2	Database Connectivity Errors	109
A.8.3	Case-sensitive Collation in SQL Server	110
A.8.4	Configuring the TIBCO Oracle or SQL Server Driver for NTLM Authentication	110
A.8.5	Maximum Packet Size in MySQL	110
A.8.6	Case Sensitivity for Table and Column Names	111
A.8.7	PostgreSQL: Job Scheduling Error	111
A.8.8	Invalid SQL statement Error with TIBCO JDBC Driver Under WebLogic	111
A.8.9	Performance Issues with Oracle JDBC Queries	112
A.8.10	Using an Oracle Service Name	112
A.8.11	Error Running a Scheduled Report	112
A.8.12	Error Running a Report	113
A.8.13	Save Error with DB2 Database	113
A.8.14	BeanDefinitionStoreException with DB2 with Vendor's Driver	113
A.8.15	JDBC Driver Loading Error on Import/Export from WebLogic or WebSphere	114
A.9	Application Server-related Problems	114
A.9.1	Memory Issues Running Under Tomcat	114
A.9.2	Java Out of Memory Error	114
A.9.3	Configuration File Locations	114
A.9.4	Context.xml on Tomcat: Special Case	115
A.9.5	Tomcat Installed Using apt-get/yum	115
A.9.6	GlassFish Modifications	116
A.9.7	JBoss Modifications	117

A.9.8 WebSphere Modifications	119
A.9.9 WebLogic Modifications	119
A.9.10 Disabling User Session Persistence in Application Servers	120
A.9.11 Session Error Using JasperReports Server and Tomcat 7	120
A.10 License-related Errors	121
A.10.1 License Not Found Errors	121
A.10.2 Failure to Unlock TIBCO JDBC Driver Error	121
A.10.3 License Not Found or License Corrupt Error with Tomcat as a Service	122
A.11 Problems Importing and Exporting Data from the Repository	122
A.11.1 Exporting a Repository That Contains UTF-8	122
A.12 Problems with Upgrade	123
A.12.1 Oracle Error on Upgrade when PL/SQL Not Enabled	123
A.12.2 DB2 Script Error on Upgrade	123
A.12.3 Include Audit Events on Upgrade	123
A.12.4 Overlay Upgrade Permissions Error with Bundled Installation	124
A.12.5 Overlay Upgrade Domain Issue with MySQL and MariaDB JDBC Driver	124
Appendix B Manually Creating the JasperReports Server Database	125
B.1 PostgreSQL	125
B.2 MySQL	126
B.3 Oracle	127
B.4 DB2	129
B.5 SQL Server	130

CHAPTER 1 INTRODUCTION

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



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

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

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

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

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

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

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

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

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

This chapter contains the following sections:

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

1.1 Conventions

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

Convention	Description
<js-install>	The root directory where JasperReports Server will be installed by the binary installer.
<js-install>	For manual installations, the directory where you unpack the WAR file distribution TIB_js-jrs_6.4.0_bin.zip. See 2.5, “Choosing an Installation Directory,” on page 19 for the default values.
<glassfish>	The directory where GlassFish is installed.
<java>	The directory where Java is installed.
<jboss>	The directory where JBoss is installed.
<postgresql>	The directory where PostgreSQL is installed. If you use our bundled instance of PostgreSQL, it's in the <js-install> directory.
<tomcat>	The directory where Apache Tomcat is installed. If you use our bundled instance of Tomcat, it's in <js-install> directory.

1.2 Supported Platforms

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

1.3 Using IBM JDK 1.7

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

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

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

This creates the `WEB-INF/csrf` folder in the current location and places the extracted file there.

- b. Rename the file from `Websphere.jrs.csrfguard.properties` to `jrs.csrfguard.properties` using the following command:

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

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

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

1.4 JasperReports Server Distributions

JasperReports Server has two main distribution packages.

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

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

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

1.4.1 Installer Support

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

1.4.1.1 Installer Naming for 32-bit and 64-bit

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

Installer Type	Naming
32-bit installer (Linux only)	TIB_js-jrs_6.4.0_installer-linux-x86.run
64-bit installer	TIB_js-jrs_6.4.0_installer-win-x64.exe TIB_js-jrs_6.4.0_installer-linux-x64.run TIB_js-jrs_6.4.0_installer-osx-x64.app.zip
Note: x86 is shorthand referring to the 386, 486, and 586 CPU architecture.	

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

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

1.4.1.2 Installer Distribution Components

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

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

1.4.1.3 Installing with Existing Components

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

- If you want to use an existing Tomcat, it must be on the local machine.
- If you want to use an existing PostgreSQL, it can be on a local or remote machine. If it's on a remote Linux machine, configure PostgreSQL to allow remote connections as described in [2.7.4, “Enabling Connections to a Remote Host,” on page 22](#).

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

1.4.1.4 Running Components as Windows Services

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

Control Panel > System and Security > Administrative Tools > Services

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

- jasperreportsPostgreSQL
- jasperreportsTomcat

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

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

1.4.1.5 Installer on Windows 7

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

1.4.2 WAR File Binary Distribution Support

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



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

The application server should reside on the local machine, but the target database can be on a remote server. Using a remote PostgreSQL database on some Linux platforms requires a change to its configuration file, as described in [2.7.4, “Enabling Connections to a Remote Host,” on page 22](#).

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

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

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

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

1.4.2.1 About Bundled Apache Ant

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

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

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

The bundled Apache Ant has an additional jar (ant-contrib.jar) that enables conditional logic in Ant. If you're running your own Ant, copy this jar to your Ant/lib folder.



On Linux and Solaris, the js-ant commands may not be compatible with all shells. If you have errors, use the `bash` shell explicitly. For more information, see [A.4, “Bash Shell for Solaris, IBM AIX, HP UX and FreeBSD,”](#) on page 105.

1.5 Release Notes

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

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

1.6 System Requirements

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

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

1.7 Support for Internationalization

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

CHAPTER 2 INSTALLING JASPERREPORTS SERVER

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

- **Pre-installation Steps**
- **Starting the Installer**
- **Accepting the License Agreement**
- **Choosing Installation Type**
- **Selecting a Tomcat Configuration**
- **Selecting a PostgreSQL Configuration**
- **Installing Sample Data**
- **Completing the Installation**
- **Post-installation Steps**
- **Starting and Stopping the Server**
- **Logging into JasperReports Server**
- **Log Files**
- **Uninstalling the Server**

2.1 Pre-installation Steps

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



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

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



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

2.2 Starting the Installer

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

`TIB_js-jrs_6.4.0_installer-win-x64.exe` (64 bit only)



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

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

`./TIB_js-jrs_6.4.0_installer-linux-x86.run` (32 bit)

`./TIB_js-jrs_6.4.0_installer-linux-x64.run` (64 bit)

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

`TIB_js-jrs_6.4.0_installer-osx-x64.app` (64 bit only)

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

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



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



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

2.3 Accepting the License Agreement

You must accept the license agreement or exit the installer.

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

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

2.4 Choosing Installation Type

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

Install All Option:

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

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

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

Custom Install:

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

2.5 Choosing an Installation Directory

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

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

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

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



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

2.6 Selecting a Tomcat Configuration

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

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

If you choose this option, the installer puts an instance of Tomcat 8 onto your system. Later, after choosing a bundled or existing database, you're prompted for the server port and shutdown port Tomcat will use. You can accept the default values or enter alternate values.

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

If you have an instance of Tomcat on your system, you can choose this option. Later, after choosing a bundled or existing database, you're prompted for the location of Tomcat.

- Browse to the folder where you installed Tomcat.

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

2.7 Selecting a PostgreSQL Configuration

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

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

2.7.1 Choosing the Bundled PostgreSQL

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

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

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

2.7.2 Choosing an Existing PostgreSQL on a Local Host

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



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

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

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



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

2.7.3 Using an Existing PostgreSQL on a Remote Host

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

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

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

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

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

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

or

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

You might also need to enable connections as described below.

2.7.4 Enabling Connections to a Remote Host

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

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

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

1. Locate the following PostgreSQL host-based authentication (hba) configuration file on the remote PostgreSQL server instance:
Windows: C:\Program Files\PostgreSQL\9.0\data\pg_hba.conf
Linux: /var/lib/pgsql/data/pg_hba.conf
2. Add the IP address of your local JasperReports Server installation machine to this file. For example, to allow the local installation machine with address 192.168.12.10 to connect to the PostgreSQL server, add this entry to the pg_hba.conf file:

```
host all 192.168.12.10/32 trust
```
3. Allow TCP/IP connections to the remote PostgreSQL server instance by making the following change to the postgresql.conf file on the remote machine:
From: listen_addresses = 'localhost'
To: listen_addresses = '*'
4. Restart PostgreSQL.
5. Using your local PostgreSQL client tools, verify that you can connect to the target remote PostgreSQL from the local installation machine, as described in [2.7.3, “Using an Existing PostgreSQL on a Remote Host,”](#) on page 21.

2.8 Installing Sample Data

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

- SugarCRM data that simulates three years of operations for a fictitious company that relies on the SugarCRM open source application.
- Foodmart data that simulates three years of operations for a fictitious company.
- JasperReports Server repository resources such as Reports, OLAP Views, Ad Hoc Topics, Domains, Data Sources, and Input Controls.

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

2.9 Completing the Installation

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

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



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

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

2.10 Post-installation Steps

2.10.1 Updates Made by the Installer During Installation

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

Updates made to the application server

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

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

Updates made to the PostgreSQL database

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

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

2.10.2 Installer Output Log File Location

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

2.10.3 Setting your Java JVM Options

You need to set your Java JVM options. There are number of files where you can do this; refer to [4.1, “Setting JVM Options for Application Servers,” on page 43](#).

2.10.4 Installing a New License File

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

To obtain a commercial license, contact [TIBCO Jaspersoft Technical Support](http://support.tibco.com) (<http://support.tibco.com>) or your sales representative.

To upgrade the evaluation license to a commercial one, copy the commercial license file over the evaluation license file.

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

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

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

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

The file name should be `jasperserver.license`

3. Restart the server

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

Restart JasperReports Server and log in to see if the license grants access. For information about license errors, see the troubleshooting section [A.10, “License-related Errors,” on page 121](#).

For additional license configuration options, refer to [4.2, “Setting Up the JasperReports Server License,” on page 48](#).

2.10.5 License File for Existing Tomcat as Windows Service

If you installed JasperReports Server into an existing Tomcat installation on a Windows system running as a Windows Service *and* the license file is not in the default location, because you didn't choose the default `<js-install>` installation directory ([2.4, “Choosing Installation Type,” on page 18](#)), manually configure Tomcat to locate the license file.

Follow these steps to examine and update the license location:

1. Open the Tomcat configuration tool by right-clicking the Tomcat icon in your quick-launch bar (usually in the lower-right corner of your desktop) or from the Windows 7 menu, expand **Start > All Programs > Apache Tomcat**. Right-click **Configure Tomcat** and select **Run as administrator**.
2. Select the **Java** tab.
3. At the bottom of the **Java Options** field, enter the following option:
`-Djs.license.directory=<js-install>`
 For example:
`-Djs.license.directory=C:\Jaspersoft\jasperreports-server-`
4. Stop and restart the application server.

You should now be able to run JasperReports Server.

2.11 Starting and Stopping the Server

- [Start/Stop Menu — Windows](#)
- [Start/Stop Scripts — Linux](#)
- [Start/Stop Apps — Mac OSX](#)

2.11.1 Start/Stop Menu — Windows

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

2.11.1.1 Start/Stop Menus — Bundled Tomcat and PostgreSQL

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

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

2.11.1.2 Additional Information about the Bundled Tomcat and PostgreSQL

JasperReports Server Windows Service Names:

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

- `jasperreportsPostgreSQL`
- `jasperreportsTomcat`

Preventing JasperReports Server from starting up automatically:

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

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

To Start JasperReports Server from the Windows Services Panel:

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

To Start JasperReports Server from the CMD Shell:

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

Running Processes:

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

- `postgres.exe`
- `tomcat7.exe`

2.11.1.3 Start/Stop Scripts – No Bundled Applications

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

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



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

1. Database server.
2. Application server.

2.11.2 Start/Stop Scripts – Linux

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

2.11.2.1 Manual Start/Stop

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

Start JasperReports Server:

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

Stop JasperReports Server:

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

To start and stop individual components:

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

2.11.2.2 Auto Start/Stop with Bundled Tomcat and PostgreSQL

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

<js-install>/scripts/linux/jasperserver

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

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

2.11.3 Start/Stop Apps – Mac OSX

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

/Applications/jasperreports-server-

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

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

jasperServerStart.app

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

jasperServerStop.app

The Mac lists the following information in the Activity Monitor:

- java
- or
- org.apache.catalina.startup.Bootstrap
- postgres

2.11.3.1 Start/Stop Apps – Mac Dock

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

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

2.11.3.2 Start/Stop JasperReports Server – Mac Terminal Shell

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

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

3. To start PostgreSQL, Tomcat, and JasperReports Server, enter:
`./ctlscript.sh start`
4. To shutdown PostgreSQL, Tomcat, and JasperReports Server, enter:
`./ctlscript.sh stop`
5. To start and stop individual components:

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

2.12 Logging into JasperReports Server

To log into JasperReports Server on any operating system:

1. Start JasperReports Server.
2. Open a supported browser: Firefox, Internet Explorer, Chrome, or Safari.
3. Log into JasperReports Server by entering the startup URL in your browser's address field. The URL depends upon your application server. If you installed the default, bundled Tomcat use:
`http://<hostname>:8080/jasperserver-pro`
 - <hostname> is the name or IP address of the computer hosting JasperReports Server.
 - 8080 is the default port number for the Apache Tomcat application server. If you used a different port when installing your application server, specify its port number instead of 8080.

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

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

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

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



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

To log into JasperReports Server on Windows:

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

To log into JasperReports Server on Mac OSX:

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

```
/Applications/<js-install>/jasperServerLogin
```

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

To use the Dock to log into JasperReports Server:

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

2.13 Log Files

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

Tomcat: `<tomcat>/webapps/jasperserver-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 output and logging levels in the `log4j.properties` file in the WEB-INF folder.

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

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

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

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

2.14 Uninstalling the Server

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

2.14.1 Windows

To uninstall JasperReports Server on Windows 7:

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

2.14.2 Linux

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

To uninstall JasperReports Server:

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

```
cd <js-install>
./uninstall
```
3. Respond Y or yes to the prompt that asks if you want to remove JasperReports Server from this computer.

2.14.3 Mac OSX

To use Finder to uninstall JasperReports Server:

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

2.14.4 Uninstall Survey

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

CHAPTER 3 INSTALLING THE WAR FILE DISTRIBUTION

For production environments, use the stand-alone WAR file distribution to install the JasperReports Server application. Download the WAR file distribution from [TIBCO Jaspersoft Technical Support](http://support.tibco.com) (<http://support.tibco.com>) or contact your sales representative. The WAR file distribution comes in a file named TIB_js-jrs_6.4.0_bin.zip in compressed ZIP format.

This chapter contains the following sections:

- **Applications Supported by the WAR File Distribution**
- **Installing the WAR File Using js-install Scripts**
- **Additional Steps for Using DB2 and js-install Scripts**
- **Starting the Server**
- **Logging into the Server**
- **Troubleshooting Your Server Configuration**
- **Installing the WAR File Manually**

3.1 Applications Supported by the WAR File Distribution

3.1.1 Database and Application Server Support

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

Database	Application Server	Instructions Located In
PostgreSQL MySQL DB2	Apache Tomcat JBossEAP/Wildfly GlassFish	This chapter.
Oracle SQL Server	WebSphere	Chapter 5, “Installing the WAR File for WebSphere,” on page 65
	WebLogic	Chapter 6, “Installing the WAR File for WebLogic,” on page 87

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

3.1.2 Operating System Support for Bash Shell

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

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

3.2 Installing the WAR File Using js-install Scripts

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

Prerequisites for installing the WAR file:

1. Install a supported version of the Java Development kit (JDK). See the *TIBCO Jaspersoft Platform Support* document on the [Documentation section](#) of the Jaspersoft Community website for a list.
2. Create and set the `JAVA_HOME` system environment variable to point to the Java JDK location.
3. Locate or install one of the following application servers:
 - Apache Tomcat 6, 7, or 8
 - JBoss EAP 6.x or 7.x or Wildfly 8.x, 9.x, 10.x (additional steps may be required for JBoss EAP or Wildfly. Please see [3.4, “Additional Steps for Using JBoss EAP or Wildfly,”](#) on page 36)
 - Glassfish 4.1 using the default domain (domain1)
If you use a custom domain with GlassFish, see [A.9.6, “GlassFish Modifications,”](#) on page 116.
4. Locate or install the PostgreSQL, MySQL, Oracle, SQL Server, or DB2 database. If you use DB2, follow the steps in [3.3, “Additional Steps for Using DB2 and js-install Scripts,”](#) on page 35.



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

For an optional pre-install validation test, run `js-install.bat/sh test`. See [3.7.3.1, “js-install Script Test Mode,”](#) on page 39 for more information.

To install the WAR file using js-install scripts:

The scripts are intended for the bash shell.



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

1. Extract all files from `TIB_js-jrs_6.4.0_bin.zip`. Choose a destination, such as `C:\Jaspersoft` on Windows, `/home/<user>` on Linux, or `/Users/<user>` on Mac.

The directory, `TIB_js-jrs_6.4.0_bin`, appears in the file location you choose.

2. Copy the `<database>_master.properties` file for your database from `sample_conf` and paste it to `buildomatic`:

- Copy from — `<js-install>/buildomatic/sample_conf/`
- Paste to — `<js-install>/buildomatic`

For example, if your database is PostgreSQL, copy `postgresql_master.properties` to `<js-install>/buildomatic`.

3. Rename the file you copied to `default_master.properties`.
4. Edit the `default_master.properties` file to add the settings for your database and application server.

Table 3-1 lists sample property values for each supported database.

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

Database	Sample Property Values
PostgreSQL	<pre>appServerType=tomcat [jboss-eap-6, wildfly, glassfish, skipAppServerCheck] appServerDir=c:\\Program Files\\Apache Software Foundation\\Tomcat 8.0.36 dbHost=localhost dbUsername=postgres dbPassword=postgres</pre>
MySQL	<pre>appServerType=tomcat [jboss-eap-6, wildfly, glassfish, skipAppServerCheck] appServerDir=c:\\Program Files\\Apache Software Foundation\\Tomcat 8.0.36 dbUsername=root dbPassword=password dbHost=localhost</pre>
All Oracle versions other than Oracle 12c with CDB/PDB (including 12c non-CDB)	<pre>appServerType=tomcat [jboss-eap-6, wildfly, glassfish, skipAppServerCheck] appServerDir=c:\\Program Files\\Apache Software Foundation\\Tomcat 8.0.36 dbUsername=jasperserver dbPassword=password sysUsername=system sysPassword=password dbHost=hostname</pre>
Oracle 12c with CDB/PDB	<p>Use settings for Oracle, except for the following changes:</p> <pre>dbUsername=c##jasperserver sid=cdb1</pre> <p>If you are using sample databases:</p> <pre>foodmart.dbUsername=c##foodmart sugarcrm.dbUsername=c##sugarcrm</pre>

Database	Sample Property Values
DB2	<pre>appServerType=tomcat [jboss-eap-6, wildfly, glassfish, skipAppServerCheck] appServerDir=c:\\Program Files\\Apache Software Foundation\\Tomcat 8.0.36 dbUsername=db2admin dbPassword=password dbHost=localhost</pre> <p>If you use DB2, follow the steps in 3.3, “Additional Steps for Using DB2 and js-install Scripts,” on page 35</p> <p>For DB2 8.x, change your deployed JDBC driver as described in 4.4, “Locating and Changing Buildomatic Configuration Files,” on page 56.</p>
SQL Server	<pre>appServerType=tomcat [jboss-eap-6, wildfly, glassfish, skipAppServerCheck] appServerDir=c:\\Program Files\\Apache Software Foundation\\Tomcat 8.0.36 dbUsername=sa dbPassword=sa dbHost=localhost</pre>



Note the following:

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

Backslashes in paths must be doubled in properties files, for example:

`appServerDir=C:\\Apache Software Foundation\\Tomcat 7.`

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

For Oracle 12c without CDB/PDB, do not use the `c##jasperserver dbUsername`. Use the standard `jasperserver dbUsername` instead.



On Linux, if Tomcat is installed using apt-get, yum, or rpm, see [A.9.5, “Tomcat Installed Using apt-get/yum,”](#) on page 115.

5. Password encryption

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

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

```
encrypt=true
```



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

6. Run the js-install scripts:

- Start your database server.
- Stop your application server.
- Open Command Prompt as Administrator on Windows or open a terminal window on Linux and Mac OSX.

- d. Run the `js-install` script:

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



For Oracle 12c, do not install sample databases. You must use `js-install.bat/sh minimal` with Oracle 12c.

If you encounter errors during the `js-install` script execution, see [3.7.3, “Error Running js-install Scripts \(js-install.bat/sh\),” on page 38](#).

- Set Java JVM Options (required), as described in [4.1, “Setting JVM Options for Application Servers,” on page 43](#).
- Set up the license (required) as described in [4.2, “Setting Up the JasperReports Server License,” on page 48](#).



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

3.3 Additional Steps for Using DB2 and js-install Scripts

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

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

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

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

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

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

- Continue installing JasperReports Server as described in [3.2, “Installing the WAR File Using js-install Scripts,” on page 32](#).

Further considerations:

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

3.4 Additional Steps for Using JBoss EAP or Wildfly

If you're using JBoss EAP or Wildfly as your application server and Oracle, SQL Server, or DB2 as your database, an additional set of steps is required to handle the JDBC driver. If you're using a driver different from the one supplied with JasperReports Server, you should have already downloaded a JDBC driver jar for your database type. (See [4.3, “Working With JDBC Drivers,” on page 50](#), if you have not yet done this.)

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

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

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

Look for the section "Setup JDBC Driver"

Uncomment and edit these two lines:

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

So that they look like this:

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

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

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

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

Look for the section "Setup JDBC Driver"

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

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

So that it looks like this:

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

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

Note: If your JDBC driver filename does not have a version number, you may need to rename the file and give it a version number.

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

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

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

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

3.5 Starting the Server

To run JasperReports Server:

Start your application server using one of these commands:

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

To view the JasperReports Server application logs, see [2.13, “Log Files,” on page 29](#).

3.6 Logging into the Server

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

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

Example:

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

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

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

Use the following credentials to log into JasperReports Server:

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

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



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

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

3.6.1 JasperReports Server Heartbeat

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

- Operating System and JVM type and version
- Application Server and Database type and version
- JasperReports Server type and version

- Unique, anonymous identifier value

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

```
heartbeat.enabled=false
```

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

3.7 Troubleshooting Your Server Configuration

This section helps you troubleshoot the most common installation problems.

3.7.1 Startup Problems

If you encounter a problem trying to run a new JasperReports Server, an incorrect database configuration is the likely culprit. Another common cause is a mistake in the application server configuration files. For information about resolving these types of errors, see [Appendix A, “Troubleshooting,” on page 103](#).

3.7.2 Error Running a Report

If you have trouble running reports in your new JasperReports Server instance, see [“Error Running a Report” in Appendix A, “Troubleshooting,” on page 103](#).

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

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

To troubleshoot problems running js-install scripts:

1. Open the output log file located in:

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

To recreate your default_master.properties settings:

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

The js-install script uses the current values in the default_master.properties file.

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

3.7.3.1 js-install Script Test Mode

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

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

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

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

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

3.7.4 Problem Connecting to a Cloud Database Instance

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

3.8 Installing the WAR File Manually

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

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

To install the WAR file distribution using manual buildomatic steps:

1. Start your database server.
2. Stop your application server.
3. Create and edit a default_master.properties file to add the settings in for your database and application server as described in 3.2, “Installing the WAR File Using js-install Scripts,” on page 32.

4. Open a Command Prompt as Administrator on Windows or open a terminal window on Linux or Mac. Run the following commands:

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

Commands	Description
<code>cd <js-install>/buildomatic</code>	Makes the buildomatic directory your current directory.
<code>js-ant create-js-db</code>	Creates the JasperReports Server repository database.
<code>js-ant create-sugarcrm-db</code> <code>js-ant create-foodmart-db</code>	(Optional) Creates the sample databases.
<code>js-ant load-sugarcrm-db</code> <code>js-ant load-foodmart-db</code>	(Optional) Loads sample data into the sample databases.
<code>js-ant update-foodmart-db</code>	(Optional) Initializes the sample databases
<code>js-ant init-js-db-pro</code> <code>js-ant import-minimal-pro</code>	Initializes the <code>jasperserver</code> database, loads core application data. Running <code>js-ant import-minimal-pro</code> is mandatory. The server cannot function without this data.
<code>js-ant import-sample-data-pro</code>	(Optional) Loads the demos that use the sample data.
<code>js-ant deploy-webapp-pro</code>	Configures and deploys the WAR file to Tomcat, JBoss, or Glassfish.



On non-Linux Unix platforms, the `js-ant` commands may not be compatible with all shells. If you have errors, use the `bash` shell explicitly. For more information, see [A.4, “Bash Shell for Solaris, IBM AIX, HP UX and FreeBSD,” on page 105](#).

If you encounter an error when running `create-sugarcrm-db`, `create-foodmart-db`, or `create-js-db`, you can create the JasperReports Server database manually using the database administration tool for your particular database type. To create the JasperReports Server database manually for PostgreSQL, MySQL, Oracle, SQL Server, or DB2, see [Appendix B, “Manually Creating the JasperReports Server Database,” on page 125](#).

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

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

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

5. Set Java JVM Options (required) as described in [4.1, “Setting JVM Options for Application Servers,”](#) on [page 43](#).
6. Set up the license (required) as described in [4.2, “Setting Up the JasperReports Server License,”](#) on [page 48](#).

CHAPTER 4 JVM OPTIONS, LICENSE SETUP, WORKING WITH JDBC DRIVERS

This chapter contains the following sections:

- **Setting JVM Options for Application Servers**
- **Setting Up the JasperReports Server License**
- **Working With JDBC Drivers**
- **Locating and Changing Buildomatic Configuration Files**
- **Configuring Report Scheduling**
- **Updating XML/A Connection Definitions**

4.1 Setting JVM Options for Application Servers

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

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



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

4.1.1 Tomcat and JBoss JVM Options

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



The following example settings are for 64-bit systems. For 32-bit systems, see “**Setting your Java JVM Options**” on page 24.

JVM Options on Windows (64 bit)	
Options for all app servers	<pre>set JAVA_OPTS=%JAVA_OPTS% -Xms1024m -Xmx2048m -Xss2m set JAVA_OPTS=%JAVA_OPTS% -XX:+UseConcMarkSweepGC set JAVA_OPTS=%JAVA_OPTS% -XX:+CMSCClassUnloadingEnabled</pre>
Additional options for JDK 1.7	<pre>set JAVA_OPTS=%JAVA_OPTS% -XX:PermSize=32m -XX:MaxPermSize=512m</pre>
For Oracle	<pre>set JAVA_OPTS=%JAVA_OPTS% -Doracle.jdbc.defaultNChar=true</pre>
Additional options for JBoss	<pre>set JAVA_OPTS=%JAVA_OPTS% -Djavax.xml.soap.MessageFactory=org.apache.axis.soap.MessageFactoryImpl set JAVA_OPTS=%JAVA_OPTS% -Djavax.xml.soap.SOAPConnectionFactory=org.apache.axis.soap.SOAPConnectionFactoryImpl set JAVA_OPTS=%JAVA_OPTS% -Djavax.xml.soap.SOAPFactory=org.apache.axis.soap.SOAPFactoryImpl</pre>
Additional option for JBoss EAP 6.1	<pre>set JAVA_OPTS=%JAVA_OPTS% -Djavax.xml.transform.TransformerFactory=org.apache.xalan.processor.TransformerFactoryImpl</pre>

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

JVM Options on Linux and Mac OSX (64 bit)	
Options for all app servers, all JDKs	<pre>export JAVA_OPTS="\$JAVA_OPTS -Xms1024m -Xmx2048m -Xss2m" export JAVA_OPTS="\$JAVA_OPTS -XX:+UseConcMarkSweepGC" export JAVA_OPTS="\$JAVA_OPTS -XX:+CMSCClassUnloadingEnabled"</pre>
Additional options for JDK 1.7	<pre>export JAVA_OPTS="\$JAVA_OPTS -XX:PermSize=32m -XX:MaxPermSize=512m "</pre>
For Oracle	<pre>export JAVA_OPTS="\$JAVA_OPTS -Doracle.jdbc.defaultNChar=true"</pre>
Additional options for JBoss	<pre>export JAVA_OPTS="\$JAVA_OPTS -Djavax.xml.soap.MessageFactory=org.apache.axis.soap.MessageFactoryImpl" export JAVA_OPTS="\$JAVA_OPTS -Djavax.xml.soap.SOAPConnectionFactory=org.apache.axis.soap.SOAPConnectionFactoryImpl" export JAVA_OPTS="\$JAVA_OPTS -Djavax.xml.soap.SOAPFactory=org.apache.axis.soap.SOAPFactoryImpl"</pre>
Additional option for JBoss EAP 6.1	<pre>export JAVA_OPTS="\$JAVA_OPTS -Djavax.xml.transform.TransformerFactory=org.apache.xalan.processor.TransformerFactoryImpl"</pre>

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

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

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

4.1.2 Changing JVM Options for Tomcat as a Windows Service

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

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

Start > Programs > Apache Tomcat > Configure Tomcat (Run as administrator)

2. In the Apache Tomcat Properties dialog, click the **Java** tab.
3. In the Java Options field, add your JAVA_OPTS values according to the tables above.
Enter only the options preceded by -X or -D, not set JAVA_OPTS=%JAVA_OPTS%.
Enter only one Java option setting per line.

4. For instance, on JDK 1.7, add options as follows:

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



These example settings are for 64-bit systems. For 32-bit systems, see **“Setting your Java JVM Options” on page 24**.

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

4.1.3 Changing JVM Options for Bundled Tomcat on Linux

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

1. Open the following file for editing:
`cd <js-install>/apache-tomcat/scripts/ctl.sh`
2. Look for the `start_tomcat()` function and locate the `JAVA_OPTS` variable inside it.
3. Modify the `JAVA_OPTS` values according to the tables above. For example, on JDK 1.7:

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



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

4. Save and close the `ctl.sh` file.
5. Stop and restart PostgreSQL and Tomcat as described in **2.11, “Starting and Stopping the Server,” on page 25**.

4.1.4 Changing GlassFish JVM Options

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

4.1.4.1 Setting GlassFish JVM Options with the asadmin Command

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

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

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

2. Restart the application server using the following commands:

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

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

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

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

4.1.4.2 Setting GlassFish JVM Options by Editing domain.xml

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

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

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

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

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

4.2 Setting Up the JasperReports Server License

JasperReports Server requires a license and comes with an evaluation license valid for 30 days. Please contact [TIBCO Jaspersoft Technical Support](http://support.tibco.com) (<http://support.tibco.com>) or your sales representative to get your commercial license.

The license file is in the following location:

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

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

- Expiration date, number of users, and/or number of CPUs
- 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.

4.2.1 Default License Configuration for All Application Servers

At startup JasperReports Server automatically looks for the `jasperserver.license` file in the home directory of the system user running the application server. **Table 4-1** lists the application server user home directories for supported operating systems.

To configure the license:

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

Table 4-1 License Locations

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

4.2.2 User-Defined License Location

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

4.2.2.1 Alternate License Setup for Tomcat

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

On Windows:

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

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

Alternatively, create an empty file called `<tomcat>/bin/setenv.bat`.

2. Below that line or in the new file, insert the following line:

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

For example:

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

On Linux and Mac OSX:

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

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

Alternatively, create an empty file called `<tomcat>/bin/setenv.sh`.

2. Below that line or in the new file, insert the following line:

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

For example:

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

4.2.2.2 Alternate License Setup for Bundled Tomcat as a Windows Service

The Windows binary installer installs the bundled Tomcat component as a Windows Service by default. The steps to specify a specific folder to hold the `jasperserver.license` are:

1. Open this file for editing:

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

2. Look for the second line of two lines that set JVM options, specifically the line which contains the license string. For example, `-Djs.license.directory=C:\Jaspersoft\jasperreports-server-`.

3. Update the line to point to your license location, for example:

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

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

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

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

4.2.2.3 Alternate License Location for Existing Tomcat as a Windows Service

Windows 7:

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

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

4.2.2.4 Alternate License Setup for JBoss

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

On Windows:

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

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

2. Below that line, insert the following line:

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

For example:

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

On Linux and Mac OSX:

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

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

2. Below that line, insert this 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"
```

4.3 Working With JDBC Drivers

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

4.3.1 Open Source JDBC Drivers

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

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

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

4.3.1.1 PostgreSQL Example

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

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

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

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

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

Uncomment and change:

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

To:

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

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

4.3.1.2 MySQL Example

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

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

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

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

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

Edit your `default_master.properties` file:

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

Uncomment and change:

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

To:

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

4.3.2 Commercial JDBC Drivers

As of version 5.6.1, JasperReports Server includes the TIBCO JDBC drivers for the following commercial databases. You can connect to these databases using the TIBCO JDBC driver without additional steps. The driver name is in the `<js-install>\buildomatic\conf_source\db\<your_database>\jdbc` directory in the following form:

- Oracle — `Tloracle-X.X.jar`
- SQL Server — `Tlsqlserver-X.X.jar`
- DB2 — `Tldb2-X.X.jar`



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

If you're using the default settings for the driver, you don't need to edit `default_master.properties`.

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

4.3.2.1 Download an Optional JDBC Driver Jar

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

Download Driver Jar from Vendor Website:

You can download a commercial JDBC driver from the vendor's website. Here are some sites where you can download packages for supported databases:

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

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

Collect Driver Jar from Existing Application:

You may already have a JDBC driver in an application running on your network. If so, you can simply copy that driver jar to the JasperReports Server install location.

4.3.2.2 Oracle Example

1. Copy your Oracle driver to the following directory:
`<js-install>/buildomatic/conf_source/db/oracle/native.jdbc`
2. Change to the `<js_install>/buildomatic` directory and open `default_master.properties` in a text editor.
3. Go to the Additional Settings section in this file.
4. Go to the first setup item, Setup Standard Oracle JDBC Driver.
5. Follow the instructions to uncomment the required properties and enable your driver. The following example shows how to set up `default_master.properties` to point to a driver named `ojdbc6-11.2.0.3.jar` using SID:

```
# 1) Setup Standard Oracle JDBC Driver
#
# Uncomment and modify the value to native
jdbcDriverMaker=native
#
# Uncomment and modify the value in order to change the default
# 1a) Driver will be found here: <path>/buildomatic/conf_source/db/oracle/native.jdbc
#
maven.jdbc.groupId=oracle
maven.jdbc.artifactId=ojdbc6
maven.jdbc.version=11.2.0.3
```

If you're using an Oracle service name instead of an SID, uncomment the line `serviceName=` and add your service name.

6. Save the `default_master.properties` file.

4.3.2.3 SQL Server Example

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

```
# 1) Setup Standard SQLServer JDBC Driver
#
# Uncomment and modify the value to native
jdbcDriverMaker=native
#
# Uncomment and modify the value in order to change the default
# Driver will be found here: <path>/buildomatic/conf_source/db/sqlserver/native.jdbc
#
maven.jdbc.groupId=sqlserver
maven.jdbc.artifactId=sqljdbc
maven.jdbc.version=1.6
```

6. Save the `default_master.properties` file.

4.3.2.4 DB2 Example

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

```
# 1) Setup Standard DB2 JDBC Driver
#
# Uncomment and modify the value to native
jdbcDriverMaker=native
#
# Uncomment and modify the value in order to change the default
# Driver will be found here: <path>/buildomatic/conf_source/db/db2/native.jdbc
#
maven.jdbc.groupId=ibm
maven.jdbc.artifactId=db2jcc
maven.jdbc.version=9.7
```

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

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

7. Save the default_master.properties file.

4.3.3 Working with Oracle RAC

As of JasperReports Server 6.1, you can use Oracle 12c RAC with the TIBCO JDBC Oracle driver. This driver works with Oracle 12c RAC with the following settings:

- Use Oracle 12C for non-CDB/PDB connection settings as described in **All Oracle versions other than Oracle 12c with CDB/PDB (including 12c non-CDB)**.
- Use `js-install.bat/sh minimal` with Oracle 12c. This option does not install sample databases.

To support additional functionality with Oracle RAC, such as load balancing with multiple servers, you need to configure your application server and manually set up the correct connection URL.

4.3.3.1 Tomcat Load Balancing Example

1. Change to the `<tomcat>/webapps/jasperserver-pro/META-INF` directory and open `context.xml` in a text editor.
2. Edit the `url` property by adding additional connection properties for data sources you want. The following example shows how to set up connection pool to use Oracle RAC with load balancing with a primary server and three alternate servers:

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

4.3.3.2 JBoss EAP/WildFly Load Balancing Example

1. Change to the `<jboss-install>/standalone/deployments/jasperserver-pro.war/WEB-INF` directory and open `js-jboss7-ds.xml` in a text editor.
2. Edit the `connection-url` tag for the data sources you want. The following example shows how to set up a connection pool to use Oracle RAC with load balancing with a primary server and three alternate servers.

```
<datasource jta="false" jndi-name="java:/jdbc/jasperserver" pool-name="jasperserver"
    enabled="true" use-cm="false">
    <connection-url>jdbc:tibcosoftware:oracle//server1:1521;ServiceName=SERVICE;AlternateServers=
(server2:1521,server3:1521,server4:1521);LoadBalancing=true</connection-url>
    <driver>TOracle-5.14.2.jar</driver>
    <security>
        <user-name>jasperserver</user-name>
        <password>password</password>
    </security>
```

```

<pool>
  <min-pool-size>5</min-pool-size>
  <max-pool-size>50</max-pool-size>
  <prefill>true</prefill>
</pool>
<validation>
  <validate-on-match>false</validate-on-match>
  <background-validation>false</background-validation>
  <check-valid-connection-sql>SELECT 1 FROM DUAL</check-valid-connection-sql>
</validation>
<statement>
  <share-prepared-statements>false</share-prepared-statements>
</statement>
</datasource>

```

4.3.3.3 Glassfish Load Balancing Example

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

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

```

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

```

4.3.3.4 Websphere Load Balancing Example

Using the Websphere console, set up the TIBCO Oracle driver as described in [5.2.2, “Defining a JNDI Name and Sample Data Sources for DB2,” on page 75](#). Then define custom properties as described in [Table 5-8, “Custom Properties for TIBCO JDBC Driver for Oracle,” on page 79](#). For example, set the AlternateServers and LoadBalancing properties.

Property Name	Value
serverName	server1
portNumber	1521
ServiceName	SERVICE
AlternateServers	(server2:1521,server3:1521,server4:1521)
LoadBalancing	true

4.3.3.5 WebLogic Load Balancing Example

Using the WebLogic console, set up the driver as described in [6.3.1.1, “Configuring a TIBCO JDBC Oracle Connection,” on page 96](#). Then set the URL property for your servers, for example:

```

jdbc:tibcosoftware:oracle//server1:1521;ServiceName=SERVICE;AlternateServers=
(server2:1521,server3:1521,server4:1521);LoadBalancing=true

```

4.3.4 Application Server Copy-to Locations

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

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

4.4 Locating and Changing Buildomatic Configuration Files

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

4.4.1 Regenerating Buildomatic Settings

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

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

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

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

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

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

4.4.2 Locating Buildomatic-Generated Property Files

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

The generated property files are in this location:

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

Some of the key configuration files are:

```
js.jdbc.properties
js.quartz.properties
js-glassfish-ds.xml
js-jboss-ds.xml
maven_settings.xml - (used for source code build)
```


More generated property files are:

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

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

```
META-INF/context.xml
```

```
WEB-INF/hibernate.properties
```

```
WEB-INF/js.quartz.properties
```

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

4.4.3 Buildomatic Location for JasperReports Server WAR File

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

```
<js-install>/jasperserver-pro.war
```

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

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

4.4.4 Buildomatic Location for SQL Scripts

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

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

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

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

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

```
<js-install>/buildomatic/install_resources/sql/postgresql/upgrade-postgresql-6.0.0-6.1.0-pro.sql
```

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

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



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

4.4.5 Buildomatic Location for Database Creation Scripts

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

Buildomatic attempts to create the `jasperserver` database via JDBC when the `create-js-db` target is executed. The scripts and property files used to create the `jasperserver` database are located in the following directory:

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

4.4.6 Buildomatic Location for Sample Data Catalog ZIP Files

Buildomatic includes export files that hold the JasperReports Server sample data (with examples of new features). This sample data is loaded when you run the buildomatic target `import-sample-data-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:

```
js-catalog-<db_name>-minimal-pro.zip
```

```
js-catalog-<db_name>-pro.zip
```

4.4.7 Hibernate Properties Settings

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

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

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

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

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

Hibernate property values are:

PostgreSQL:	<code>metadata.hibernate.dialect=com.jaspersoft.hibernate.dialect.PostgresqlNoBlobDialect</code>
MySQL 5.1:	<code>metadata.hibernate.dialect=org.hibernate.dialect.MySQLInnoDBDialect</code>
MySQL 5.5:	<code>metadata.hibernate.dialect=org.hibernate.dialect.MySQL5InnoDBDialect</code>
DB2:	<code>metadata.hibernate.dialect=com.jaspersoft.ji.hibernate.dialect.DB2JICustomDialect</code>
Oracle:	<code>metadata.hibernate.dialect=com.jaspersoft.ji.hibernate.dialect.OracleJICustomDialect</code>
SQL Server:	<code>metadata.hibernate.dialect=com.jaspersoft.ji.hibernate.dialect.SQLServerJICustomDialect</code>

4.4.8 Database Connection Configuration Files

4.4.8.1 Tomcat

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

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

Older versions of Tomcat will create a copy of the `context.xml` file with a changed name that will be read instead of the one found in the `jasperserver-pro` war file. This can be confusing for Tomcat users who try to change their database settings. If you change your settings, delete the file in this location:

```
<tomcat>/conf/Catalina/localhost/jasperserver-pro.xml
```

4.4.8.2 JBoss

When you've set up the buildomatic configuration for your database, the JBoss data source definition file will be automatically created with the appropriate settings for JasperReports Server.

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

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

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

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

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

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

For more information, refer to troubleshooting section [A.9.7, “JBoss Modifications,” on page 117](#).

4.4.8.3 Glassfish

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

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

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

To deploy the data source definition manually, run a command similar to the following:

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

4.5 Configuring Report Scheduling

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

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

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

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

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

The following configurations are discussed in this section:

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

4.5.1 Mail Server Configuration Settings

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

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

4.5.2 Database Settings for the Quartz Driver Delegate Class

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



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

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

Configuration File		
<app-server>/<deployment>/WEB-INF/js.quartz.properties		
Property	Database	Value
quartz.delegateClass	MySQL	org.quartz.impl.jdbcjobstore.StdJDBCDelegate
	PostgreSQL	org.quartz.impl.jdbcjobstore.PostgreSQLDelegate
	DB2	org.quartz.impl.jdbcjobstore.DB2v8Delegate
	Oracle	org.quartz.impl.jdbcjobstore.StdJDBCDelegate
	SQL Server ¹	org.quartz.impl.jdbcjobstore.StdJDBCDelegate
1. For SQL Server on WebSphere 8.5 use org.quartz.impl.jdbcjobstore.MSSQLDelegate		

4.5.3 Settings for the Report Scheduler Web URI

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

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

To set this value manually, edit this file:

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

Change the properties as shown in the following table.

Property	App Server	Example Value
report.scheduler.web.deployment.uri	Apache Tomcat	http://localhost:8080/jasperserver-pro
	JBoss	http://localhost:8080/jasperserver-pro
	GlassFish	http://localhost:8080/jasperserver-pro
	WebLogic	http://localhost:7001/jasperserver-pro
	WebSphere	http://localhost:9080/jasperserver-pro

4.5.4 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 to include the schema, if you use one. In the default configuration, only DB2 requires an explicit schema name.



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

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

Property	Description
quartz.tablePrefix	The prefix for the quartz table, including any schema name, for example JSPRSVR.QRTZ_ for DB2.

4.5.5 Settings for Import-Export

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



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

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

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

Change the following properties:

Property	Description
quartz.delegateClass	Set to the same value as described in 4.5.2, “Database Settings for the Quartz Driver Delegate Class,” on page 61.
quartz.tablePrefix	Set to the same value as described in 4.5.4, “Settings for the Quartz Table Prefix,” on page 62

4.5.6 Setting Properties in the default_master.properties File

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

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

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

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

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

4.5.6.1 Report Scheduler Email Properties

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

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

4.5.6.2 Diagnostic Properties

The following properties configure the Diagnostic functionality:

```
diagnostic.jmx.usePlatformServer = false
diagnostic.jmx.port = 10990
diagnostic.jmx.name = jasperserver
diagnostic.jmx.rmiHost = localhost
```

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

4.6 Updating XML/A Connection Definitions

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

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

The sample connections are:

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

To validate and update these resources:

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

CHAPTER 5 INSTALLING THE WAR FILE FOR WEBSPHERE

JasperReports Server supports deployment on the IBM WebSphere Application Server, but requires its own database to store information such as users, organizations, and the repository. WebSphere users must use the WAR file distribution to install JasperReports Server. Download the WAR file distribution from [TIBCO Jaspersoft Technical Support](http://support.tibco.com) (<http://support.tibco.com>) or contact your sales representative. The WAR file distribution comes in a file named `TIB_js-jrs_6.4.0_bin.zip`.

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

This chapter contains the following sections:

- **Procedure for Installing and Deploying the WAR File in WebSphere**
- **Logging into the Server**
- **Configuring Report Scheduling**
- **Updating XML/A Connection Definitions (Optional)**
- **Troubleshooting your Configuration**

5.1 Procedure for Installing and Deploying the WAR File in WebSphere

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

5.1.1 Installing WebSphere and a Database

To install WebSphere and a database:

1. Make sure you're using a supported version of WebSphere. See the *TIBCO JasperReports Server Supported Platform Datasheet* for more information.
2. Check that the WebSphere installation created a `JAVA_HOME` system environment variable. The variable needs to be set to the `JAVA` directory in the WebSphere installation.
3. Install the database (PostgreSQL, MySQL, Oracle, SQL Server, or DB2).



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

5.1.2 Preparing Server Files

To prepare JasperReports Server files:

1. Unpack TIB_js-jrs_6.4.0_bin.zip to a top-level directory. Unpacking the ZIP file creates the directory TIB_js-jrs_6.4.0_bin.
2. (Required) Manually create and load the JasperReports Server database.
3. (Optional) Manually create and load the sample databases. See [Appendix B, “Manually Creating the JasperReports Server Database,” on page 125](#) for instructions.
4. (Required) Manually import the default users and organization.
 - a. Copy the <dbType>_master.properties file for your database from sample_conf and paste it to buildomatic:
 - `cd <js-install>/buildomatic`
 - Copy from — `<js-install>/buildomatic/sample_conf`
 - Paste to — `<js-install>/buildomatic`
 For example, copy `sample_conf/postgresql_master.properties` to `buildomatic`.
 - b. Rename the file you copied to `default_master.properties`.
 - c. Edit the `default_master.properties` file:
 - Set `appServerType` to `skipAppServerCheck`.
 - Change `dbUsername`, `dbPassword`, and `dbHost` to the appropriate settings for your database.
 - If you're using a port other than the default for your database, or if you've installed the database on a remote machine, change the `dbPort` field under Custom Properties to the appropriate settings.
 Each `sample_conf/<dbType>_master.properties` file contains appropriate sample values.
 - d. Start your database server.
 - e. Open a Command Prompt as Administrator and run these commands:

Table 5-1 Buildomatic Targets to Execute

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

5. Set up your license file. For information, refer to 4.2, “Setting Up the JasperReports Server License,” on page 48.

5.1.3 Configuring CSRFGuard, Hibernate, and Quartz Settings

Before deploying the JasperReports Server WAR file, update the CSRFGuard, Hibernate, Quartz, and settings as described here.

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

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

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

This creates the `WEB-INF/csrf` folder in the current location and places the extracted file there.

- b. Rename the file from `Websphere.jrs.csrfguard.properties` to `jrs.csrfguard.properties` using the following command:

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

2. Extract the `web.xml` file using the commands below:

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

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

Open the `WEB-INF/web.xml` file for editing and replace every occurrence of

```
<res-type>javax.sql.ConnectionPoolDataSource</res-type>
```

with

```
<res-type>javax.sql.DataSource</res-type>
```

For example, change the following:

```
<resource-ref>
  <description>JasperServer Metadata repository</description>
  <res-ref-name>jdbc/jasperserver</res-ref-name>
  <res-type>javax.sql.ConnectionPoolDataSource</res-type>
  <res-auth>Container</res-auth>
</resource-ref>
```

to:

```
<resource-ref>
  <description>JasperServer Metadata repository</description>
  <res-ref-name>jdbc/jasperserver</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <res-auth>Container</res-auth>
</resource-ref>
```

Do the same for the Supermart and Foodmart databases.

3. Copy the already configured files for `hibernate.properties` and `js.quartz.properties` to the `WEB-INF` folder.
(Buildomatic configured these files for your database type in the steps above.)
From:

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

To:

```
<js-install>/WEB-INF
```
4. Edit the scheduler URI port value for WebSphere in the `js.quartz.properties`:
Edit `js.quartz.properties`:
Set :

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

To:

```
report.scheduler.web.deployment.uri=http://localhost:9080/jasperserver-pro
```
5. 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 for your mail server.
6. Now that you have modified/updated the individual configuration files, you must replace them in the WAR file archive using the following commands.

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

7. If you have modified the `web.xml` file, replace that file in the WAR file archive using the following additional commands.

```
cd <js-install>  
"%JAVA_HOME%\bin\jar" uf jasperserver-pro.war WEB-INF/web.xml
```

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

```
cd <js-install>  
zip -d jasperserver-pro.war WEB-INF/lib/stax-api-1.0.2.jar  
zip -d jasperserver-pro.war WEB-INF/lib/jta-1.1.jar
```

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

5.1.4 Configuring a JDBC Provider in WebSphere

To configure a JDBC Provider in WebSphere:

1. Launch the WebSphere Administrative Console and navigate to **Resources > JDBC > JDBC Providers**.
2. On the JDBC providers page, click the **Guided Activity** link at the top of the **JDBC Providers** page and follow the Integrated Solutions Console instructions:
 - a. Configure credentials for a secure database:
 - Use the J2C authentication aliases panel to create a new authenticated user.

- In Global Security, click **New** and enter the user alias, user ID, and password. The following table shows the credentials that WebSphere uses to access the database.

Table 5-2 J2C Authentication Alias Settings

	Alias	Example User ID	Example Password
PostgreSQL	postgresql_jasperdb	postgres	postgres
MySQL	mysql_jasperdb	root	password
Oracle	jasperserver_user	jasperserver	password
DB2	db2admin_user	db2admin	password
SQL Server	jasperserver_user	sa	sa

- Connect to a database panel. From the Scope drop-down, choose **Node:<node_name>,Server=<server_name>**
- Click the **New** button to create a new JDBC Provider.
- Select your database type:
 - If you're using PostgreSQL, MySQL, or a TIBCO JDBC driver — select User-defined.
 - If you want to use the JDBC driver built and distributed by the MySQL project, see [4.3.1.2, “MySQL Example,” on page 51](#)
 - If you're using a vendor-supplied driver for DB2, Oracle, or SQL Server — select your database.
- Select or enter these options. Your options depend on your database type.

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

Database type	Implementation class name or type	Name
Oracle	Connection pool data source	Oracle JDBC Driver
SQL Server	Connection pool data source	Microsoft SQL Server JDBC Driver

3. Click **Next** and enter the database classpath information for the JDBC provider.
 - a. For TIBCO JDBC drivers for Oracle, SQL Server, or DB2, locate the correct JAR in `<js-install>\buildomatic\conf_source\db\<your_database>\jdbc`:
 - Oracle — `<js-install>\buildomatic\conf_source\db\oracle\jdbc\Tloracle-X.X.jar`
 - SQL Server — `<js-install>\buildomatic\conf_source\db\sqlserver\jdbc\TIsqlserver-X.X.jar`
 - DB2 — `<js-install>\buildomatic\conf_source\db\db2\jdbc\TIdb2-X.X.jar`
 Copy the JAR file to a location in your WebSphere deployment and specify that location for the JDBC driver path.



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

- b. For PostgreSQL, MySQL, and vendor drivers for Oracle, SQL Server, and DB2, enter the following:
 `<js-install>\buildomatic\conf_source\db\<your_database>\jdbc\`

For example, enter:

`C:\TIB_js-jrs_6.4.0_bin\buildomatic\conf_source\db\postgresql\jdbc\postgresql-9.2-1002.jdbc4.jar`

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



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

4. To ensure you have full support for import/export from the command line, copy your JDBC driver to the following location. If you are not using the command line for import/export, you can skip this step:

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

from: or

`<js-install>\buildomatic\conf_source\db\<your_database>\native.jdbc\`

to: `<js-install>\buildomatic\conf_source\iePro\lib`

5. Click **Next** to proceed to the next step.
6. Review the JDBC provider information you entered and click **Finish**.

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

1. Click the name of the JDBC provider that you just created. For example, for PostgreSQL, click **PostgreSQL JDBC Provider**.



To use a database other than PostgreSQL, configure the database connections and custom properties as described in [5.2, “Configuring Other Database Connections,” on page 74](#).

2. Click **Data sources** in the Additional Properties of the JDBC provider details panel.
3. To create a new data source, click **New**. The new data source wizard appears.

4. Enter the data source name: `jasperserver`
5. Enter the JNDI name: `jdbc/jasperserver`
6. Click **Next**, choose **Select an existing JDBC provider**, then select **PostgreSQL JDBC Provider** from the drop-down list.
7. Click **Next** and accept the default helper class (`com.ibm.websphere.rsadapter.GenericDataStoreHelper`). Select the check box to use this data source in container managed persistence (CMP).
8. Click **Next** and select the Setup security aliases, as shown in the following table.

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

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

To set the connection pool size:

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

To define custom properties:

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

In the Messages area a success or failure message appears. The failure message gives you information about which custom properties you need to define. For example, in the case of PostgreSQL 9.0, a message indicates that the error is related to a null database name.



If you are using the TIBCO JDBC driver with SQL Server, and you see an error such as the following:

```
Could not find stored procedure 'master..xp_jdbc_open2' DSRA0010E: SQL
State = HY000, Error Code=2,812
```

Add the following variable to **Custom properties**:

```
enable2Phase = false
```

2. Navigate to the **jasperserver** data sources General Properties page.
3. In Additional Properties on the right side of the General Properties page, click **Custom properties**.
4. Scroll down the list of properties and select **databaseName**. Set the value to `jasperserver`.
5. Set **serverName** to the correct value for your server.

To create optional sugarcrm and foodmart data sources:

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

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

- Click **Save directly to the master configuration**.

Next, deploy the WAR file in WebSphere as described in [5.1.5, “Deploying the WAR File in WebSphere,” on page 72](#).

5.1.5 Deploying the WAR File in WebSphere

To deploy the JasperReports Server WAR file in WebSphere:

- In the Administrative Console, navigate to **Applications > New Application** and select **New Enterprise Application**.
JasperReports Server is a modern application, based on Java Servlet version 2.4, so you do not select the older, WebSphere V4-compliant application type.
- Browse to `<js-install>/jasperserver-pro.war` on the local file system. Keep the default setting (**Fast path**) selected and click **Next**.
- On the Select installation options page, accept all the default settings and 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. Click **Next**.
- On the Map modules to servers page, select **jasperserver**. Click **Next**.
- On the Map resource references to resources page, map the resources you want:
 - First, select the **Browse** button under the `jdbc/jasperserver` resource. In the page that opens, select the `jdbc/jasperserver` radio button, and click **Apply**. Then select the check box next to the `jdbc/jasperserver` resource.
 - If you plan to run the sample reports, follow the same steps for `jdbc/sugarcrm` and `jdbc/foodmart`, making sure to select the correct radio button for each one.
 - When you have mapped all resources, select the check boxes next to every resource have mapped.
 - Click **Next**.
- On the Map virtual hosts page, choose the **JasperServer UI application module**. Click **Next**.
- In the Map context roots for Web modules, enter `jasperserver-pro`.
- Click **Next**, review the summary information and start the installation process. (The installation process may take a while.)
- Click **Save directly to the master configuration**.

5.1.6 Setting JVM Options

To set the Java JVM Options:

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

To configure your Java JVM options:

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

Generic JVM Options on Windows	
Options for all databases	<pre>-Dclient.encoding.override=UTF-8 -Xms1024m -Xmx2048m -Xss2m -XX:+UseConcMarkSweepGC -XX:+CMSClassUnloadingEnabled -Djavax.xml.soap.MessageFactory=org.apache.axis.soap.MessageFactoryImpl -Djavax.xml.soap.SOAPConnectionFactory=org.apache.axis.soap.SOAPConnectionFactoryImpl -Djavax.xml.soap.SOAPFactory=org.apache.axis.soap.SOAPFactoryImpl -Djavax.xml.transform.TransformerFactory=org.apache.xalan.processor.TransformerFactoryImpl</pre>
Additional for JDK 1.7	<pre>set JAVA_OPTS=%JAVA_OPTS% -XX:PermSize=32m -XX:MaxPermSize=512m</pre>
Additional for Oracle	<pre>set JAVA_OPTS=%JAVA_OPTS% -Doracle.jdbc.defaultNChar=true</pre>

Generic JVM Options on Linux	
Options for all databases	<pre>-Dclient.encoding.override=UTF-8 -Xms1024m -Xmx2048m -Xss2m -XX:+UseConcMarkSweepGC -XX:+CMSClassUnloadingEnabled -Djavax.xml.soap.MessageFactory=org.apache.axis.soap.MessageFactoryImpl -Djavax.xml.soap.SOAPConnectionFactory=org.apache.axis.soap.SOAPConnectionFactoryImpl -Djavax.xml.soap.SOAPFactory=org.apache.axis.soap.SOAPFactoryImpl -Djavax.xml.transform.TransformerFactory=org.apache.xalan.processor.TransformerFactoryImpl</pre>
Additional for JDK 1.7	<pre>export JAVA_OPTS="-XX:PermSize=32m -XX:MaxPermSize=512m"</pre>
Additional for Oracle	<pre>export JAVA_OPTS="\$JAVA_OPTS -Doracle.jdbc.defaultNChar=true"</pre>



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

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

To configure class loading:

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

- Restart WebSphere.

Next, start JasperReports Server as described in 5.3, “Starting and Restarting JasperReports Server,” on page 81.

5.2 Configuring Other Database Connections

5.2.1 Defining a JNDI Name and Sample Data Sources for MySQL

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

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

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

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

To set the connection pool size:

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

To define custom properties:

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

In the Messages area a success or failure message appears. The failure message gives you information about which custom properties you need to define. For example, in the case of PostgreSQL 9.0 and MySQL, a

message indicates that the error is related to a null database name.

2. Navigate to the jasperserver data sources General Properties page.
3. In Additional Properties on the right side of the General Properties page, click **Custom properties**.
4. Scroll down the list of properties and select **databaseName**. Set the value to `jasperserver`.
5. Create a new property called **url**. Enter the following value and save the change:
`jdbc:mysql://localhost/jasperserver?useUnicode=true&characterEncoding=UTF-8`
6. Click **Save directly to the master configuration**.

To create optional sugarcrm and foodmart data sources:

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

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

2. Click **Save directly to the master configuration**.
3. Set the connection pool size as described in [“To set the connection pool size” on page 74](#).

Next, deploy the WAR file in WebSphere as described in [5.1.5, “Deploying the WAR File in WebSphere,” on page 72](#).

5.2.2 Defining a JNDI Name and Sample Data Sources for DB2

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

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

Field Name	Value
Driver type	4
Database name	jsprsvr

Field Name	Value
Server name	localhost
Port number	50000

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

To set the connection pool size:

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

To define custom properties:

1. In the list of JDBC data sources, select the check box for the newly created **jasperserver** data source, and click **Test Connection**.
2. In the Messages area a success or failure message appears. The failure message gives you information about which custom properties you need to define.
3. Edit the following properties, adding any that are missing, then save the changes:

Table 5-3 Properties for TIBCO JDBC Driver for DB2

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



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

Table 5-4 Properties for Vendor JDBC Driver for DB2

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

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

To create optional sugarcrm and foodmart data sources:

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

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

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

Table 5-6 Custom Properties for TIBCO JDBC Driver for DB2 with WebSphere

Property Name	Value	
alternateID	FOODMART	SUGARCRM
batchPerformanceWorkaround	true	

Property Name	Value	
databaseName	FOODMART	SUGARCRM
serverName	localhost	

Table 5-7 Custom Properties for Vendor's Driver for DB2 with WebSphere

Property Name	Value	
currentSchema	FOODMART	SUGARCRM
resultSetHoldability	1	

- Click **Save directly to the master configuration**.
- Set the connection pool size as described in [“To set the connection pool size” on page 76](#).

Next, deploy the WAR file in WebSphere as described in [5.1.5, “Deploying the WAR File in WebSphere,” on page 72](#).

5.2.3 Defining a JNDI Name and Sample Data Sources for Oracle

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

- Click the name of the JDBC provider you just created. For example, **TIBCO JDBC Provider - Oracle** or **Oracle JDBC Driver**.
- Click Data sources in the Additional Properties of the JDBC provider details panel.
- To create a new data source, click New. The new data source wizard appears.
- Enter the data source name: `jasperserver`
- Enter the JNDI name: `jdbc/jasperserver`
- Click **Next**.
- (If you're using the TIBCO JDBC Provider, skip to [step 9](#).) If you are using the vendor's Oracle driver, choose **Select an existing JDBC provider**, then select **Oracle JDBC Driver** from the drop-down list.
- Click **Next** and enter the following values:

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

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

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

To set the connection pool size:

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

To define custom properties:

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

Table 5-8 Custom Properties for TIBCO JDBC Driver for Oracle

Property Name	Value
serverName	localhost
SID	ORCL
portNumber	1521
CatalogOptions	0
batchPerformanceWorkaround	true



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

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

To create optional sugarcrm and foodmart data sources:

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

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

Field Name	Value	
Component-managed authentication alias	<node>/foodmart_user	<node>/sugarcrm_user

Property Name	Value
portNumber	1521
serverName	localhost

- Click **Save directly to the master configuration**.
- Set the connection pool size as described in [“To set the connection pool size” on page 79](#).

Next, deploy the WAR file in WebSphere as described in [5.1.5, “Deploying the WAR File in WebSphere,” on page 72](#).

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

To define a JDBC provider:

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

To set the connection pool size:

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

To define custom properties:

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

2. Navigate to the jasperserver data sources **General Properties** page.
3. In **Additional Properties** on the right side of the page, click **Custom properties** and define properties. The following table shows how to set properties if you are using the TIBCO JDBC driver for SQL Server

Table 5-9 Custom Properties for TIBCO JDBC Driver for SQL Server

Property Name	Value
batchPerformanceWorkaround	true
databaseName	jasperserver
serverName	localhost
portNumber	1433



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

To create optional sugarcrm and foodmart data sources:

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

1. In the list of JDBC data sources, click the link for the newly created **jasperserver** data source.
2. Click **Save directly to the master configuration**.
3. Set the connection pool size as described in “**To set the connection pool size**” on page 80.

Next, deploy the WAR file in WebSphere as described in 5.1.5, “**Deploying the WAR File in WebSphere,**” on page 72.

5.3 Starting and Restarting JasperReports Server

To start the jasperserver-pro application:

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

5.4 Logging into the Server

1. Go to the following URL to log in:
`http://<hostname>:9080/jasperserver-pro`

Where <hostname> is localhost, a machine name, or an IP address. The login page should appear after some time to compile the necessary JSP files.

2. Log in with administrative credentials:

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

If you have trouble logging in and get the following error message, you may be running at a WebSphere patch level that needs further configuration:

Page cannot be found, HTTP 404 error

Refer to the troubleshooting section [A.9.8, “WebSphere Modifications,” on page 119](#).



The first time you log into JasperReports Server, you'll be prompted to opt-in to the JasperReports Server Heartbeat. For more information, refer to [3.6.1, “JasperReports Server Heartbeat,” on page 37](#).

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

5.5 Configuring Report Scheduling

Scheduled reporting in JasperReports Server allows you to run reports at specified times and gives you the option to send an email notification to users when a new report is available.

5.5.1 Additional Fix for Scheduled Report with JNDI Data Source

On the WebSphere application server, reports using a JNDI data source require additional configuration to correctly resolve the JNDI lookup.

The jasperserver-pro.war archive includes WebSphere-specific configuration files. These files are generated during the installation process. To enable this fix, you'll need to remove the `webSphere` prefix from the names of these files. 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)



A work manager class is used to run scheduled report jobs on WebSphere. The JNDI name of the work manager and its default value (`wm/default`) are configured in `js.scheduling.properties`. The number of threads that run report jobs is provided by the work manager configuration.

5.5.2 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.

For more information about setting up report scheduling, refer to [4.5, “Configuring Report Scheduling,” on page 59](#).

5.6 Updating XML/A Connection Definitions (Optional)

If you loaded the sample data, and you want to run the Analysis XML/A examples, you'll 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 [4.6, “Updating XML/A Connection Definitions,” on page 63](#).

5.7 Troubleshooting your Configuration

5.7.1 Startup Problems

The most common problems are errors in the database configuration. These are typically errors in the database configuration files or in the application server configuration files. For information about resolving these errors, refer to troubleshooting section [Appendix A, “Troubleshooting,” on page 103](#).

5.7.2 Error Running Report

If you have trouble running reports in your new JasperReports Server instance, refer to troubleshooting section [A.8.12, “Error Running a Report,” on page 113](#). If you're having trouble running the MDX example Topic or SugarCRM OLAP view, you need to update the port for XML/A connections. See [5.6, “Updating XML/A Connection Definitions \(Optional\),” on page 83](#).

5.7.3 Filter Error Using MySQL

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

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

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

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

5.7.4 Error Creating Internationalized Name

If you encounter errors when creating resources with internationalized names, and you have an Oracle database, configure your Oracle JDBC driver. Set the Oracle-specific option listed in the tables of [5.1.6, “Setting JVM Options,” on page 72](#).

5.7.5 Xerces Error

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

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

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

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

- Remove the xercesImpl library from the following location:


```
<websphere>\profiles\AppSrv<NN>\installedApps\<node>\jasperserver-pro_war.ear\
jasperserver-pro.war\WEB-INF\lib
```
- Update the xercesImpl library to a new version (if it's an old version).

5.7.6 OLAP View Fails With Exception

The following error may occur because AspectJ needs class loaders to be tried out in a specific order:

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

Change the default class loader policy:

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

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

- Click **OK**.

4. Save the master configuration.
5. Restart the WebSphere server.

CHAPTER 6 INSTALLING THE WAR FILE FOR WEBLOGIC

JasperReports Server supports deployment on the WebLogic Application Server, but requires its own database to store information such as users, organizations, and the repository. WebLogic users need the WAR file distribution to install JasperReports Server. Download the WAR file distribution from [TIBCO Jaspersoft Technical Support](http://support.tibco.com) (<http://support.tibco.com>) or contact your sales representative. The WAR file distribution comes in a file named TIB_js-jrs_6.4.0_bin.zip.

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

This chapter contains the following sections:

- **Procedure for Installing the WAR File for WebLogic**
- **Setting Java Properties**
- **Configuring Other Database Connections**
- **Starting the Server**
- **Logging into the Server**
- **Configuring Report Scheduling**
- **Restarting the Server**
- **Updating XML/A Connection Definitions (Optional)**
- **Troubleshooting Your JasperReports Server Configuration**

6.1 Procedure for Installing the WAR File for WebLogic

To meet prerequisites for installing the WAR file for WebLogic:

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



The target database can be on a remote server.

To install the WAR file for WebLogic:

1. Extract all files in TIB_js-jrs_6.4.0_bin.zip into a top-level directory, such as C:\Jaspersoft on Windows or /home/<user> on Linux.

Unpacking the ZIP file creates the directory TIB_js-jrs_6.4.0_bin.

2. Check that WebLogic is installed in the default location on your local machine.

If WebLogic is not installed in the default location, or if you encounter problems using the buildomatic scripts, set up the database manually as described in [Appendix B, “Manually Creating the JasperReports Server Database,” on page 125](#). After setting up the database manually, skip [step 6](#) through [step 9](#), and proceed to [step 10](#).

3. (If you're using MySQL, you can skip this step.) Copy your JDBC driver to WebLogic.

- a. PostgreSQL example: Copy the JDBC jar from

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

to

`<weblogic_home>/server/lib`

- b. TIBCO JDBC Oracle driver example: Copy the following two jars

`<js-install>/buildomatic/conf_source/db/oracle/jdbc/TIoracle-X.X.jar`

and

`<js-install>/buildomatic/install_resources/extra-jars/jswlstc-1.0.jar`

to

`<weblogic_home>/server/lib/`

Note that the MySQL JDBC driver is included in recent versions of WebLogic.

4. To ensure you have full support for import/export from the command line, copy your JDBC driver to the following location. If you are not using the command line for import/export, you can skip this step:

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

from: or

`<js-install>\buildomatic\conf_source\db\<your_database>\native.jdbc\`

to: `<js-install>\buildomatic\conf_source\iePro\lib`

5. (If you're using PostgreSQL or MySQL, you can skip this step). If you are using the TIBCO JDBC driver for Oracle, SQL Server, or DB2, you need to add the driver to the WebLogic jdbcdrivers.xml file. To do this:

- a. Open the file `<weblogic_home>/server/lib/jdbcdrivers.xml` in a text editor.

- b. Add the correct settings for your database. The following example shows the settings for the TIBCO JDBC driver for Oracle. For DB2 or SQL Server, see [6.3.1, “Configuring TIBCO JDBC Driver Connections,” on page 96](#).


```

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

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

```

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



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

Table 6-1 Sample Values for the default_master.properties File

Database	Sample Property Values
PostgreSQL	<pre> appServerType=skipAppServerCheck appServerDir=[path to WebLogic application server] dbUsername=postgres dbPassword=postgres dbHost=localhost </pre>
DB2	<pre> appServerType=skipAppServerCheck appServerDir=[path to WebLogic application server] dbUsername=db2admin dbPassword=password dbHost=localhost </pre> <p>For DB2 8.x, change your deployed JDBC driver as described in 4.4, “Locating and Changing Buildomatic Configuration Files,” on page 56.</p>

Database	Sample Property Values
MySQL	<pre>appServerType=skipAppServerCheck appServerDir=[path to WebLogic application server] dbUsername=root dbPassword=password dbHost=localhost</pre>
Oracle	<pre>appServerType=skipAppServerCheck appServerDir=[path to WebLogic application server] sysUsername=system sysPassword=password dbUsername=jasperserver dbPassword=password dbHost=hostname</pre> <p>Note that dbUsername must be the same as the Oracle user name.</p>
SQL Server	<pre>appServerType=skipAppServerCheck appServerDir=[path to WebLogic application server] dbUsername=sa dbPassword=sa dbHost=localhost</pre> <p>If your application server runs on Java 1.5, change your deployed JDBC driver as described in 4.4, “Locating and Changing Buildomatic Configuration Files,” on page 56.</p>

9. Set up the database and optional sample databases using the buildomatic Ant scripts. Enter commands in the table below to call buildomatic Ant scripts:



Exception: For DB2, skip this step and perform [step 1](#) to [step 3](#) in [B.4, “DB2,” on page 129](#), then go to the next step ([step 10](#)) of this procedure.

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

Windows – `js-ant <target-name>`

Linux – `./js-ant <target-name>`

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

Commands	Description
<code>js-ant import-sample-data-pro</code>	(Optional) Loads the demos that use the sample data



On non-Linux Unix platforms, the `js-ant` commands may not be compatible with all shells. If you have errors, use the `bash` shell explicitly. For more information, see [A.4, “Bash Shell for Solaris, IBM AIX, HP UX and FreeBSD,” on page 105](#).

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



To use a database other than PostgreSQL, configure the database connections using settings shown in [6.3.1, “Configuring TIBCO JDBC Driver Connections,” on page 96](#) or [6.3.2, “Configuring Databases Using the Vendor's Driver,” on page 99](#).

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

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

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

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

Parameter Name	JasperReports Server	Foodmart Example	Sugar CRM Example
Database User Name	postgres		
Password	postgres		
Confirm Password	postgres		

14. Test the database connection:
 - a. For SugarCRM and Foodmart, use the default connections:
 jdbc:postgresql://localhost:5432/sugarcrm
 jdbc:postgresql://localhost:5432/foodmart
 - b. Change the URL for the `jasperserver` database to:
 jdbc:postgresql://localhost:5432/jasperserver
15. Select targets and ensure that **AdminServer** is set for all data sources.
16. In WebLogic, open an Administrative Console window and navigate to Services > Data Sources or Domain Configurations > Services > Data Sources
17. Select each created data source (JasperServerDataBase, FoodmartDataBase, SugarcrmDataBase)
18. Select the Connection Pool tab and increase the **Maximum Capacity** setting, depending on load. For most installations, a **Maximum Capacity** in the range 50 – 100 should be sufficient. If you receive connection pool errors, increase this setting; see the documentation for WebLogic for more information.
19. Click **Save**.
20. Use the Java jar tool or an unzip tool to unpack the `jasperserver-pro.war` file. For example, using the Java jar tool, enter these commands to unpack the `jasperserver-pro.war` file to a folder:

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

21. (If you're using WebLogic 12c, skip this step.) Search for conflicting JARs and delete them from the WAR file. If the following JARs are present in your WebLogic installation, you need to delete them from your JasperReports Server installation to avoid conflicts. To do this:
 - a. Search your WebLogic installation for these files:
 - jaxb-api-<ver>.jar
 - jaxb-impl-<ver>.jar
 - serializer-<ver>.jar
 - stax-api-<ver>.jar
 - xalan-<ver>.jar
 - xercesImpl-<ver>.jar
 - xml-apis-<ver>.jar
 - b. Change to the JasperReports Server WEB-INF/lib directory:


```
cd <js-install>/jasperserver-pro/WEB-INF/lib
```
 - c. Delete any conflicting JARs.

22. Update your Hibernate, Quartz, and Mail Server configuration:
- The buildomatic logic has already configured the `hibernate.properties` and `js.quartz.properties` files for your database type. So you can copy these files to the `jasperserver-pro` file as shown below.
Copy from:

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

To:

```
jasperserver-pro/WEB-INF
```
 - Edit the scheduler URI port value for WebLogic in the `js.quartz.properties`:
Edit `js.quartz.properties`:
Set :

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

To:

```
report.scheduler.web.deployment.uri=http://localhost:7001/jasperserver-pro
```
 - 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.
23. If your mail server requires authentication, edit the `applicationContext-report-scheduling.xml` file:
- Open the `jasperserver-pro/WEB-INF/applicationContext-report-scheduling.xml` file for editing and locate the `reportSchedulerMailSender` bean.
 - Set the `javaMailProperties` key="mail.smtp.auth" value to `true`.
24. Now you can change to the `jasperserver-pro` folder and re-archive the `jasperserver-pro.war` file, using commands such as the following:

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



You now have a `jasperserver-pro.war` file you can use for deploying to WebLogic.

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



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

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

```
<server>
...
    <stuck-thread-max-time>1200</stuck-thread-max-time>
    <listen-address></listen-address>
</server>
<security-configuration>
...
    <enforce-valid-basic-auth-credentials>false</enforce-valid-basic-auth-credentials>
</security-configuration>
```

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



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

26. Set JVM options as described in [6.2, “Setting Java Properties,” on page 95](#).

Deploy JasperReports Server to WebLogic:

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

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

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

6.2 Setting Java Properties

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

WebLogic Startup Settings on Windows	
Filename	<wl-domain>\bin\startWebLogic.cmd
Settings	<pre> set JAVA_OPTIONS=%JAVA_OPTIONS% -Djs.license.directory=C:\<js-install>\ -Dfile.encoding=UTF-8 -Dcom.sun.xml.namespace.QName.useCompatibleSerialVersionUID=1.0 -Djavax.xml.soap.SOAPConnectionFactory=org.apache.axis.soap.SOAPConnectionFactoryImpl -Djavax.xml.soap.MessageFactory=org.apache.axis.soap.MessageFactoryImpl -Djavax.xml.soap.SOAPFactory=org.apache.axis.soap.SOAPFactoryImpl -Xms1024m -Xmx2048m -Xss2m </pre>
Additional for JDK 1.7	<pre> set JAVA_OPTIONS=%JAVA_OPTIONS% -XX:PermSize=128m -XX:MaxPermSize=512m </pre>
Additional for Oracle	<pre> set JAVA_OPTIONS=%JAVA_OPTIONS% -Doracle.jdbc.defaultNChar=true </pre>



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

WebLogic Startup Settings on Linux	
Filename	<wl-domain>/bin/startWebLogic.sh
Settings	<pre> export JAVA_OPTIONS="\$JAVA_OPTIONS -Djs.license.directory=/home/<user>/weblogic/jasperlicense/ -Dfile.encoding=UTF-8 -Dcom.sun.xml.namespace.QName.useCompatibleSerialVersionUID=1.0 -Djavax.xml.soap.SOAPConnectionFactory=org.apache.axis.soap.SOAPConnectionFactoryImpl -Djavax.xml.soap.MessageFactory=org.apache.axis.soap.MessageFactoryImpl -Djavax.xml.soap.SOAPFactory=org.apache.axis.soap.SOAPFactoryImpl -Xms1024m -Xmx2048m -Xss2m" </pre>
Additional for JDK 1.7	export JAVA_OPTIONS="\$JAVA_OPTIONS -XX:PermSize=128m -XX:MaxPermSize=512m "
For Oracle	export JAVA_OPTIONS="\$JAVA_OPTIONS -Doracle.jdbc.defaultNChar=true"



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

6.3 Configuring Other Database Connections

6.3.1 Configuring TIBCO JDBC Driver Connections

6.3.1.1 Configuring a TIBCO JDBC Oracle Connection

If you're using the TIBCO JDBC driver for Oracle, add the driver to the WebLogic jdbcdrivers.xml file as described in [step 5](#) in [6.1, “Procedure for Installing the WAR File for WebLogic,” on page 87](#)

To use the driver, select DataDirect's Oracle Driver (Type4) from the menu after [step 11](#) in [6.1, “Procedure for Installing the WAR File for WebLogic,” on page 87](#) and enter the following properties:

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

Parameter Name	JasperReports Server	Foodmart Example	Sugar CRM Example
Properties	SID=ORCL user=jasperserver CatalogOptions=0	SID=ORCL user=foodmart CatalogOptions=0	SID=ORCL user=sugarcrm CatalogOptions=0

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

6.3.1.2 Configuring a TIBCO JDBC DB2 Connection

If you're using the TIBCO JDBC driver for DB2, add the following information to the WebLogic `jdbcdrivers.xml` file, as described in [step 5](#) in [6.1, “Procedure for Installing the WAR File for WebLogic,” on page 87](#)

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

To use the driver, select the TIBCO driver for DB2 from the menu after [step 11](#) in [6.1, “Procedure for Installing the WAR File for WebLogic,” on page 87](#) and enter the following properties. For the URL, enter the value of the `databaseName` lower in the table. For example, for the `jasperserver` database, the URL is `jdbc:tibcosoftware:db2://localhost:50000;databaseName=jsprsrvr;AlternateID=jsprsrvr`

Parameter Name	JasperReports Server	Foodmart Example	Sugar CRM Example
Name	JasperServerDataBase	FoodmartDataBase	SugarcrmDataBase
JNDI Name	JasperServerDataBase	FoodmartDataBase	SugarcrmDataBase
URL	jdbc:tibcosoftware:db2://localhost:50000;databaseName=<databaseName>;AlternateID=<databaseName>		
Driver Class Name	tibcosoftware.jdbc.db2.DB2Driver		
Properties	user=db2admin alternateID=JSPRSRVR	user=db2admin alternateID=FOODMART	user=db2admin alternateID=SUGARCRM

Parameter Name	JasperReports Server	Foodmart Example	Sugar CRM Example
portNumber	50000		
databaseName	jsprsvr	foodmart	sugarcrm
alternateId	jsprsvr	foodmart	sugarcrm
serverName	localhost		
batchPerformance Workaround	true		

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

6.3.1.3 Configuring a TIBCO JDBC SQL Server Connection

If you're using the TIBCO JDBC driver for SQL Server, add the following information to the WebLogic `jdbcdrivers.xml` file, as described in [step 5](#) in [6.1, “Procedure for Installing the WAR File for WebLogic,” on page 87](#)

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

To use the driver, select the TIBCO driver for SQL Server from the menu after [step 11](#) in [6.1, “Procedure for Installing the WAR File for WebLogic,” on page 87](#) and enter the following properties. For the URL, enter the value of the `databaseName` lower in the table. For example, for the `jasperserver` database, the URL is `jdbc:tibcosoftware:sqlserver://localhost:1433;databaseName=jasperserver`

Parameter Name	JasperReports Server	Foodmart Example	Sugar CRM Example
Name	JasperServerDataBase	FoodmartDataBase	SugarcrmDataBase
JNDI Name	JasperServerDataBase	FoodmartDataBase	SugarcrmDataBase
URL	jdbc:tibcosoftware:sqlserver://localhost:1433;databaseName=<databaseName>		

Parameter Name	JasperReports Server	Foodmart Example	Sugar CRM Example
Driver Class Name	tibcosoftware.jdbc.sqlserver.SQLServerDriver		
Properties	user=sa		
databaseName	foodmart	jasperserver	sugarcrm

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

6.3.2 Configuring Databases Using the Vendor's Driver

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

6.3.2.1 Configuring a MySQL Connection

Database Setting	Value
Host	localhost
Name or SID	jasperserver
User	root
Password	password
Port	3306
characterEncoding	UTF-8
autoReconnect	true
tinyInt1isBit	false
autoReconnectForPools	true
Hibernate Dialect	MySQLInnoDBDialect
Quartz Driver Delegate	StdJDBCDelegate

6.3.2.2 Configuring an Oracle Connection with the Vendor's Driver

To use the vendor's driver for Oracle, enter the following properties:

Database Setting	Value
Host	localhost
Name or SID	Orcl
User	jasperserver
Password	password
Port	1521
Hibernate Dialect	OracleJICustomDialect
Quartz Driver Delegate	StdJDBCDelegate

6.3.2.3 Configuring a DB2 Connection with the Vendor's Driver

To use the vendor's driver for DB2, enter the following properties:

Database Setting	JasperReports Server	Foodmart Example	Sugar CRM Example
Host	localhost		
Name or SID	jsprsrvr	foodmart	sugarcrm
currentSchema	JSPRSRVR	FOODMART	SUGARCRM
User	db2admin		
Password	password		
Port	50000		
Hibernate Dialect	DB2JICustomDialect		
Quartz Driver Delegate	DB2v8Delegate		

6.3.2.4 Configuring a SQL Server Connection with the Vendor's Driver

To use the vendor's driver for SQL Server, enter the following properties:

Database Setting	SQL Server
Host	localhost
Name or SID	jasperserver

Database Setting	SQL Server
User	sa
Password	sa
Port	1433
Hibernate Dialect	SQLServerJICustomDialect
Quartz Driver Delegate	StdJDBCDelegate

6.4 Starting the Server

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

6.5 Logging into the Server

1. To log in, go to this URL:
`http://<hostname>:7001/jasperserver-pro`
 Where <hostname> could be localhost, a machine name, or an IP address. The login page appears when the necessary JSP files are compiled.
2. Enter the following credentials:

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



The first time you log in, you'll be prompted to opt-in to the JasperReports Server Heartbeat. For more information, refer to [3.6.1, “JasperReports Server Heartbeat,” on page 37](#).

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

6.6 Configuring Report Scheduling

Scheduled reporting allows you to run reports at specified times. And you have the option to send an email notification to users when a new report is available.

For more information about setting up report scheduling, refer to [4.5, “Configuring Report Scheduling,” on page 59](#).

6.7 Restarting the Server

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

6.8 Updating XML/A Connection Definitions (Optional)

If you loaded the sample data and would like to run the XML/A examples, update the XML/A connection resources to use the correct web port. The typical port used by WebLogic is 7001. Follow the procedure in [4.6](#), “Updating XML/A Connection Definitions,” on page 63.

6.9 Troubleshooting Your JasperReports Server Configuration

If you have problems running the buildomatic scripts that set up the database, set up the database manually. For more information, see [4.1](#), “Setting JVM Options for Application Servers,” on page 43.

6.9.1 Startup Problems

Problems starting a new JasperReports Server instance are usually errors in the database or application server configuration files. To resolve these errors, see [Appendix A](#), “Troubleshooting,” on page 103.

6.9.2 Error Running Report

If you have trouble running reports in your new JasperReports Server instance, see [A.8.12](#), “Error Running a Report,” on page 113.

APPENDIX A TROUBLESHOOTING

This appendix contains the following sections:

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

A.1 Binary Installer Freezes

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

A.1.1 Installer Log Files

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

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

Windows: <js-install>/installation.log

Linux: <js-install>/installation.log

Mac <js-install>/installation.log

If you've tried multiple installs, make sure you view the most recent install log. Then you can submit the installation.log to [TIBCO Jaspersoft Technical Support](http://support.tibco.com) (<http://support.tibco.com>).

A.1.2 Installer DebugTrace Mode

You can also run the installer a second time using the `--debugtrace` option. This creates a binary output file with precise details about the execution of the installer and any problems encountered. [TIBCO Jaspersoft Technical Support](http://support.tibco.com) (<http://support.tibco.com>) can analyze this file.

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

```
jasperreports-server-_installer-linux-x64.run --debugtrace install-trace-out.bin
```

When you run the installer in `--debugtrace` mode, the installer takes extra time to write the binary output file. The final size of the output file is approximately 10 mg. Contact [TIBCO Jaspersoft Technical Support](http://support.tibco.com) (<http://support.tibco.com>) to hand off the binary file for analysis.

A.2 Error Running Buildomatic Scripts

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

A.2.1 Missing Java JDK

If you have the Java JRE (Java Runtime Environment) instead of the JDK, you won't have all the required utilities. In particular, you may see an error referring to the `tools.jar`, as in the following message:

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

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

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

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

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

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

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

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

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

A.2.4 Older Apache Ant Version

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

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

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

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

```
ant -version
```

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

```
echo %CLASSPATH%
```

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

A.3 Unable to Edit Files on Windows 7

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

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

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

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

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

The bash shell is required to execute the js-install shell scripts described in [Chapter 3, "Installing the WAR File Distribution," on page 31](#). The following js-install and js-upgrade scripts are in the buildomatic folder:

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

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

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

Alternatively, you can manually run the same “buildomatic” Ant targets that are run by the `js-install` script. These Ant targets are listed in [“Troubleshooting Your Server Configuration” on page 38](#).

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

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

For more information see [A.2.2, “Forgot to Copy the File ant-contrib.jar,” on page 104](#).

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

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

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

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

A.5 Linux Installer Issue with Unknown Host Error

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

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

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

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

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

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

To fix the `/etc/hosts` file:

1. Include entries that look like these:

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

For instance:

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

2. You can also double check the file `/etc/sysconfig/network` (if it exists). In this file it would be similar to the following:
`HOSTNAME=myhost`
3. After fixing the `/etc/hosts` file, reinstall JasperReports Server.

A.6 Installation Error with Windows Path

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

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

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

A.7 Mac OSX Issues

A.7.1 Problem Starting JasperReports Server on Mac

We have seen some issues caused by the improper shutdown of the Tomcat included with JasperReports Server. This may be caused by shutting the machine down while Tomcat is running.

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

You may see this when you double-click the `jasperServerStart.app` startup. JasperReports Server seems to be starting up, but it never actually does.

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

Delete `catalina.pid` using Finder:

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

Delete the catalina.pid file using Terminal shell:

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

```
rm catalina.pid
```

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

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

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

```
./ctlscript stop  
or  
./ctlscript stop tomcat  
./ctlscript stop postgresql
```

A.8 Database-related Problems

A.8.1 Database Privileges Required By JasperReports Server

Install/upgrade process permissions:

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

Database Component	Permissions Required
databases	CREATE DROP
tables indexes constraints	CREATE ALTER DROP
data records	INSERT UPDATE DELETE

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

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

A.8.2 Database Connectivity Errors

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

A.8.2.1 Testing the Database Connection

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

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

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

A.8.2.2 Logging into PostgreSQL

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

```
psql -U postgres jasperserver
```

A.8.2.3 Logging into MySQL

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

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

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

A.8.2.4 Logging into Oracle

Start SQL*Plus and try logging into Oracle directly. Use the password specified during installation to log in as each of these users:

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

A.8.2.5 Logging into Microsoft SQL Server

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

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

A.8.2.6 Connectivity Errors with Vendor's Driver for SQL Server

If you're using the vendor's driver for SQL Server and have configured default_master.properties as described in 4.3.2.3, “SQL Server Example,” on page 53, you'll see connection errors if you uncommented the following line:

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

Make sure this line is commented.

A.8.3 Case-sensitive Collation in SQL Server

Microsoft SQL Server doesn't support standalone case-sensitive collation. When collation is case-sensitive SQL Server also treats column and table names as case-sensitive. This can happen when setting a locale that includes case-sensitive collation. In this case you may see an error such as the following.

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

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

A.8.4 Configuring the TIBCO Oracle or SQL Server Driver for NTLM Authentication

To avoid storage of the user and password values for the database, you can configure the TIBCO Oracle or SQL Server driver to use Windows authentication. To do this, copy the NTLM authentication DLLs for your database from the <js-install>\jasperserver-war\shared-config\tools\TibcoJDBCdriverFile directory to a location in your Windows system path (defined by the PATH environment variable) and configure them as described in the instructions for the Progress DataDirect NTLM authentication configuration for your database:

- [Oracle Driver > Authentication > Configuring NTLM Authentication](#)
- [Microsoft SQL Server Driver > Authentication > Configuring NTLM Authentication](#)

A.8.5 Maximum Packet Size in MySQL

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

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

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

```
max_allowed_packet = 16M
```

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

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

A.8.5.1 Connection reset by peer MySQL Error

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

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

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

A.8.6 Case Sensitivity for Table and Column Names

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

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

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

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

Table and column names in Oracle and PostgreSQL are case-sensitive.

A.8.7 PostgreSQL: Job Scheduling Error

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

The errors will look like this:

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

If you see this error, check your Quartz properties file in the following location:

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

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

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

A.8.8 Invalid SQL statement Error with TIBCO JDBC Driver Under WebLogic

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

If you don't delete this when setting up the driver, you'll get errors like:

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

If you get this error, you need to use the WebLogic console to edit or recreate the driver for each database that failed. See [step 11 in 6.1, “Procedure for Installing the WAR File for WebLogic,” on page 87](#) as well as [6.3, “Configuring Other Database Connections,” on page 96](#).

A.8.9 Performance Issues with Oracle JDBC Queries

Setting the Oracle database localization option `defaultNChar` to `true` can substantially impact the performance of JDBC queries. When `defaultNChar` is set to `true`, the database implicitly converts all `CHAR` data to `NCHAR` when you access `CHAR` columns. If you don't need to support UTF-8 for your Oracle database, you can omit this setting.

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

To change this setting on Windows, enter a command like this at the command line:

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

To change this setting on Linux, enter a command like this at the command line:

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

A.8.10 Using an Oracle Service Name

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

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

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

```
serviceName=ORCL
```

When you're using an Oracle service name, make sure you don't set the `SID` or `dbPort` in the `default_master.properties` file.

A.8.11 Error Running a Scheduled Report

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

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

You can increase the size of this datatype by updating your `my.ini` or `my.cnf` MySQL configuration file with the following setting:

```
max_allowed_packet=32M
```


A.8.12 Error Running a Report

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

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

1. Log into JasperReports Server as a user with administrative permissions and locate the report unit that returns errors.
2. Select the report and click the **Edit** button in the toolbar and identify the data source on the fourth edit page.
3. Edit the data source in the repository and check its settings.
4. Click the **Test Connection** button.

If the connection fails, perhaps the application server can't find the driver for the data source. For example, in a default installation of JasperReports Server, Tomcat looks for data source drivers in `<js-install>/apache-tomcat/lib`.

5. Test your report. If it still returns errors, edit the data source again and try checking other values, like the port used by the database.

A.8.13 Save Error with DB2 Database

When the DB2 database is your repository database, you may get errors when saving longer strings (over 50 characters) to data entry fields in the UI. For example, saving a resource with a name over 50 characters may cause an error like this:

```
Expected status code is 200, but was 400. Response body contained:
An unexpected exception has occurred
```

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

A.8.14 BeanDefinitionStoreException with DB2 with Vendor's Driver

If you're using the vendor's driver for DB2, you need to manually add properties to `default_master.properties`, or you'll get an error like the following.

```
[java] Resource name: applicationContext-virtual-data-source.xml
[java] org.springframework.beans.factory.BeanDefinitionStoreException:
Invalid bean definition with name 'dataSource' defined in file
[/opt/JasperReports-Server-6.4.0-src/jasperserver/buildomatic/conf_source/iePro/applicationContext-
export-config.xml]:
Could not resolve placeholder 'dbPort' in string value
```

Add the following properties to your `default_master.properties`, setting the correct values for your installation:

```
db2.driverType=4
db2.fullyMaterializeLobData=true
db2.fullyMaterializeInputStreams=true
db2.progressiveStreaming=2
```

```
db2.progressiveLocators=2
dbPort=50000
js.dbName=JSPRSVR
sugarcrm.dbName=SUGARCRM
foodmart.dbName=FOODMART
```

A.8.15 JDBC Driver Loading Error on Import/Export from WebLogic or WebSphere

If you are using WebLogic or WebSphere and want to run import/export from the command line, you need to manually copy the JDBC driver to the same location as the import/export scripts. If you have not copied these files, you may encounter the following error:

```
Cannot load JDBC driver class 'tibcosoftware.jdbc.sqlserver.SQLServerDriver'
```

To fix this error, copy your database driver to the correct location:

```
from:    <js-install>\buildomatic\conf_source\db\<your_database>\jdbc\
         or
         <js-install>\buildomatic\conf_source\db\<your_database>\native.jdbc\
to:      <js-install>\buildomatic\conf_source\iePro\lib
```

A.9 Application Server-related Problems

A.9.1 Memory Issues Running Under Tomcat

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

1. Set the following parameter in the global \$CATALINA_BASE/conf/web.xml:
enablepooling = false
2. Restart Tomcat.

A.9.2 Java Out of Memory Error

If you encounter a Java out of memory error, try increasing your Java heap size setting. See [4.1, “Setting JVM Options for Application Servers,” on page 43](#). As a minimum, add `-Xms1024m -Xmx2048m` to your `JAVA_OPTS` setting.

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

A.9.3 Configuration File Locations

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

```

Tomcat:  <tomcat>/webapps/jasperserver-pro/META-INF/context.xml
        <tomcat>/webapps/jasperserver-pro/WEB-INF/hibernate.properties
        <tomcat>/apache-tomcat/webapps/jasperserver-pro/WEB-INF/web.xml      (JNDI config)
        <tomcat>/apache-tomcat/config/Catalina/localhost/jasperserver-pro.xml  (delete: see below)
JBoss:   <jboss>/standalone/deployments/jasperserver-pro.war/WEB-INF/js-jboss7-ds.xml
        <jboss>/standalone/deployments/jasperserver-pro.war/WEB-INF/hibernate.properties
        <jboss>/standalone/deployments/jasperserver-pro.war/WEB-INF/web.xml
        <jboss>/standalone/deployments/jasperserver-pro.war/WEB-INF/jboss-web.xml
GlassFish: <glassfish>/domains/domain1/autodeploy/jasperserver-pro.war/WEB-INF/hibernate.properties
        <glassfish>/domains/domain1/autodeploy/jasperserver-pro.war/WEB-INF/js.quartz.properties
        <glassfish>/domains/domain1/config/domain.xml

```

A.9.4 Context.xml on Tomcat: Special Case

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

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

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

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

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

Remember to delete the `jasperserver-pro.xml` file:

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

A.9.5 Tomcat Installed Using apt-get/yum

A.9.5.1 Setting CATALINA_HOME

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

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

```

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

```

and change it to this:

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

Note that you must set both CATALINA_HOME and CATALINA_BASE.

A.9.5.2 Database Driver Location

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

A.9.6 GlassFish Modifications

A.9.6.1 Using a Custom Domain

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

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

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

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

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

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

To fix this issue, run this command:

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

A.9.6.3 BufferOverflowException When Working With Input Controls

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

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

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

A.9.7 JBoss Modifications

A.9.7.1 JBoss 7 Startup Error

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

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

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

1. Change to the JBoss standalone configuration directory.

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

2. Open the `standalone.xml` file.

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

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

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

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

5. Save the file.

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

A.9.7.2 JBoss 7 ReservedCodeCacheSize Error

If you get a fatal error like this:

```
"out of space in CodeCache for adapters"
```

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

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

A.9.7.2.1 Linux

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

A.9.7.2.2 Windows

```
set JAVA_OPTS=%JAVA_OPTS% -DReservedCodeCacheSize=128m
```

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

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

To silence this INFO message:

1. Open the JBoss log4j configuration file for editing:
`<jboss>/server/default/conf/jboss-log4j.xml`
2. Set the logging level for the `CachedConnectionManager` class to this value:

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

A.9.7.4 Using a Non-default JBoss Profile

If JBoss is your application server, and you're using a profile other than the default, you need to set the `jboss.profile` property before running the `js-install` script in [3.2, "Installing the WAR File Using js-install Scripts," on page 32](#):

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

A.9.7.5 Using JBoss with Non-Latin Characters

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

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

```
<extensions>
.....
</extensions>

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

A.9.7.6 Maximum Post Size in Wildfly

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

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

A.9.7.7 Performance Issues with JBoss EAP 6.1

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

A.9.8 WebSphere Modifications

A.9.8.1 Page Not Found Error on Login

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

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

To solve the Page Not Found Error on Login:

1. Log in into WebSphere Administrative Console.
2. Navigate to **Application Servers > <server> > Web Container Settings > Web Container > Custom Properties**.
3. Create a new property with the following attributes:
 name: `com.ibm.ws.webcontainer.invokefilterscompatibility`
 value: `true`
4. Save the master configuration.
5. **Restart** the WebSphere server.

A.9.9 WebLogic Modifications

A.9.9.1 Schema Validation Error

You may see an error like the following on some WebLogic installations:

```
<Critical> <WebLogicServer> <BEA-000362> <Server failed. Reason:
[Management:141245]Schema Validation Error in /u01/app/oracle/WLS/user_
projects/domains/x2o_uat_01/config/config.xml
```

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

A.9.10 Disabling User Session Persistence in Application Servers

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

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

The errors appear in the JasperReports Server log when users log in after the application server has been restarted. Users don't see the errors, and they have no impact on JasperReports Server operations.

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

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

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

A.9.11 Session Error Using JasperReports Server and Tomcat 7

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

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

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

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

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

A.9.11.1 Disabling httpOnly for Cookies in Tomcat

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

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

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


A.9.11.2 Allowing Requests from Other Domains in DWR

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

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

A.10 License-related Errors

A.10.1 License Not Found Errors

Normally, the JasperReports Server installer includes an evaluation license file that you replace with a commercial license file, as described in [2.10.4, “Installing a New License File,” on page 24](#). If JasperReports Server returns an error after you replace the license file, the most likely causes are:

- You didn't clear your application server's work directory, as explained in [2.10.4](#). Delete the work directory, restart the application server, and try logging into JasperReports Server again.
- The `Djs.license.directory` property in your application server startup environment is incorrectly set: For example, in Windows the correct setting looks like this:

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

In Linux, the correct setting looks like this:

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

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

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

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

A.10.2 Failure to Unlock TIBCO JDBC Driver Error

When you use one of the TIBCO JDBC drivers, the server checks for a valid JasperReports Server license in the correct location. If no license is found, you'll see an error like this:

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

This is essentially a "License not found" error. Make sure the jar file for the TIBCO JDBC driver jar is located in JasperReports Server's `jasperserver-pro/web-inf/lib` directory, for example, `<tomcat_home>/tomcat/jasperserver-pro/web-inf/lib`. Do not place the driver jar outside the `jasperserver-pro` directory, for example, in the `<tomcat_`

home>/lib directory. The TIBCO driver jar files are Jaspersoft-specific and will not work with other applications.

If moving the jar file doesn't resolve this error, use the methods in [A.10.1, “License Not Found Errors,”](#) on [page 121](#).

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

If you have Tomcat running as a service in Windows, the installer attempts to make the proper updates so the server license file is found at when you start the application server. If the installer is unsuccessful, make sure you took the steps described in [2.10.5, “License File for Existing Tomcat as Windows Service,”](#) on [page 24](#).

A.11 Problems Importing and Exporting Data from the Repository

A.11.1 Exporting a Repository That Contains UTF-8

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

A.11.1.1 Error During Export

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

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

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

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

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

A.11.1.2 Error During Export from Repository on Oracle

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

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

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

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

A.12 Problems with Upgrade

A.12.1 Oracle Error on Upgrade when PL/SQL Not Enabled

If you're upgrading to JasperReports Server version 6.4 or later using the `js-upgrade-samedb.sh/bat` script, you may encounter an error if Oracle's Procedural Language (PL/SQL) is not enabled.

The upgrade script for upgrading to a 4.7 database from a 4.5 database requires enabling the PL/SQL language. The script is located here:

```
builddomatic/install_resources/sql/oracle/upgrade-oracle-4.5.0-4.7.0-pro.plsql
```

The error you encounter might look something like this:

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

To enable your PL/SQL language, consult the documentation for your Oracle database to enable PL/SQL.

A.12.2 DB2 Script Error on Upgrade

If you're upgrading from 4.7 to 6.4, and you use the `js-upgrade-samedb.bat/sh` script, you may get an error related to an inadequate PAGESIZE setting. The recommended minimal PAGESIZE setting for both 4.7 and 5.0 is 16384.

If the PAGESIZE setting is less than this you may get an error like this:

```
com.ibm.db2.jcc.am.SqlException:
DB2 SQL Error: SQLCODE=-670,SQLSTATE=54010, SQLERRMC=4005;
or
com.ibm.db2.jcc.am.SqlSyntaxErrorException:
DB2 SQL Error: SQLCODE=-286, SQLSTATE=42727, SQLERRMC=8192;DB2ADMIN,
DRIVER=4.11.77
```

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

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

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

A.12.3 Include Audit Events on Upgrade

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

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

Then run one of the following commands:

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

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

or

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

These commands reimport all resources from the specified export file, add the audit event, and do not overwrite existing resources.



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

A.12.4 Overlay Upgrade Permissions Error with Bundled Installation

If you're using the overlay upgrade procedure with a PostgreSQL database, and you installed an earlier version of JasperReports Server, you might see an HTTP Status 404 error.

The overlay installer is not supported with the bundled installation and may only be used with a war file installation of JRS.

A.12.5 Overlay Upgrade Domain Issue with MySQL and MariaDB JDBC Driver

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

Because JasperReports Server 5.5, 5.6, 6.0, 6.1 and the Overlay packages are distributed with the MariaDB 1.1.2 JDBC driver, this error can also occur even without an upgrade operation.

The issue may occur when a boolean filter is created on a domain. Because of a bug in older versions of the MariaDB driver, the boolean filter is evaluated as a numeric filter.

<https://mariadb.atlassian.net/browse/CONJ-72>

APPENDIX B MANUALLY CREATING THE JASPERREPORTS SERVER DATABASE

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

- [PostgreSQL](#)
- [MySQL](#)
- [Oracle](#)
- [DB2](#)
- [SQL Server](#)

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

B.1 PostgreSQL

To manually create the JasperReports Server database in PostgreSQL:

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

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

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

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

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

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

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

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

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

For more information about executing the Ant scripts, see [3.8, “Installing the WAR File Manually,” on page 39](#).

4. Set Java JVM Options (required), as described in [4.1, “Setting JVM Options for Application Servers,” on page 43](#).
5. Set up the JasperReports Server License (required) as described in [4.2, “Setting Up the JasperReports Server License,” on page 48](#).

B.2 MySQL

To manually create the JasperReports Server database in MySQL:

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



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

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

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

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

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

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

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

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

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

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

For more information about executing the Ant scripts, see [3.8, “Installing the WAR File Manually,” on page 39](#).

4. Set Java JVM Options (required), as described in [4.1, “Setting JVM Options for Application Servers,” on page 43](#).
5. Set up the JasperReports Server License (required) as described in [4.2, “Setting Up the JasperReports Server License,” on page 48](#).

B.3 Oracle

To manually create the JasperReports Server database in Oracle:

You can use the Oracle client software, `sqlplus.exe` or `sqlplus`, to interact with Oracle.



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

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

```
cd <js-install>/buildomatic/install_resources/sql/oracle
sqlplus /nolog (start sqlplus client)
SQL> connect system/password (use your sysUsername and password)
(or SQL>connect sys/password as sysdba
SQL> create user jasperserver identified by password;
SQL> grant connect, resource to jasperserver;
SQL> connect jasperserver/password@ORCL (use your password, your SID)
SQL> @js-pro-create.ddl
SQL> @quartz.ddl
SQL> exit
```

2. (Optional) Special edit to the `sugarcrm.sql` script that creates the `sugarcrm` sample database. The `sqlplus` command line tool interprets SQL statements differently than a JDBC call (that is, the way `buildomatic` executes SQL scripts). Because of this, the `sugarcrm.sql` file must be edited in order to execute using `sqlplus`. To make these edits do the following:
 - Unzip the `sugarcrm.zip` file to get the `sugarcrm.sql` file. Open `sugarcrm.sql` for editing:
 - Uncomment the `-- set define off` line to look like this `set define off` (Line 7)

- Uncomment the "--/" line that follows the CREATE TRIGGER statements (there are 12 of these toward the very end of the file on line 71,282. Just before the CREATE INDEX statements). Change to be just "/". (This terminates the trigger procedure definition in sqlplus.)
- Save the file.



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

If you load the sample databases using buildomatic, you won't need to set any variables or make any script edits.

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

Windows: `set NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P1`

Linux: `export NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P1`

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

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

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

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

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

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

For more information about executing the Ant scripts, see [3.8, “Installing the WAR File Manually,” on page 39](#).

6. Set Java JVM Options (required), as described in [4.1, “Setting JVM Options for Application Servers,” on page 43](#).
7. Set up the JasperReports Server License (required) as described in [4.2, “Setting Up the JasperReports Server License,” on page 48](#).

B.4 DB2

To manually create the JasperReports Server database in DB2:

Use the DB2 client software, `db2` or `db2cmd`, to interact with DB2.



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

1. Change to the following directory:

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

2. Enter these commands in the DB2 command window to create and initialize the repository database called `jsprsrvr` in DB2 to conform to the 8-character limitation:

```
db2 create database jsprsrvr using codeset utf-8 territory us pagesize 16384
db2 connect to jsprsrvr
db2 -tf js-pro-create.ddl
db2 -tf quartz.ddl
```

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

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

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

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

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

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

For more information about executing the Ant scripts, see [3.8, “Installing the WAR File Manually,” on page 39](#).

5. Set Java JVM Options (required), as described in [4.1, “Setting JVM Options for Application Servers,” on page 43](#).
6. Set up the JasperReports Server License (required) as described in [4.2, “Setting Up the JasperReports Server License,” on page 48](#).

Further considerations:

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

B.5 SQL Server

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



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

To manually create the JasperReports Server database in SQL Server:

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

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

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

To manually create the optional sample databases in SQL Server:

5. Extract the files in the `sugarcrm.zip` file to the level above your current directory, placing the `sugarcrm.sql` file in this directory:

```
<js-install>\jasperserver\buildomatic\install_resources\sql\sqlserver
```

6. Enter these commands to create and initialize the `sugarcrm` database:

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

7. You cannot initialize the foodmart database manually. Instead, change to the buildomatic directory and use the following buildomatic commands to create and initialize it from the command line:

```
js-ant create-foodmart-db  
js-ant load-foodmart-db
```

Alternatively, you can replace the first command and create the database manually using the following SQL Server commands, but you still have to use the buildomatic command `js-ant load-foodmart-db` to load the data:

```
1> CREATE DATABASE [foodmart]  
2> GO  
1> USE [foodmart]  
2> GO
```

To complete the manual installation of databases in SQL Server:

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

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

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

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

For more information about executing the Ant scripts, see [3.8, “Installing the WAR File Manually,” on page 39](#).

9. Set Java JVM Options (required), as described in [4.1, “Setting JVM Options for Application Servers,” on page 43](#).
10. Set up the JasperReports Server License (required) as described in [4.2, “Setting Up the JasperReports Server License,” on page 48](#).

