



# JasperReports® Server

## Source Build Guide

Version 9.0.0 | January 2024

# Contents

---

<b>Contents</b> .....	<b>2</b>
<b>Introduction</b> .....	<b>5</b>
Supported Build Configurations .....	7
JasperReports Server Source Code Archives and Current Files .....	7
<b>Components Required for Source Build</b> .....	<b>8</b>
Check Your Java JDK .....	8
Check Your Maven Version .....	8
Check Your Application Server .....	9
Check Your Database Instance .....	9
Check Your Chrome/Chromium Installation .....	10
<b>Building JasperReports Server Source Code</b> .....	<b>11</b>
Introduction to Buildomatic Source Build Scripts .....	11
Downloading and Unpacking JasperReports Server Source Code .....	11
Downloading the Source Archive .....	11
Unpacking the Source Archive .....	12
Source Code Package Structure .....	12
Check Apache Ant .....	13
Configuring the Buildomatic Properties .....	13
PostgreSQL .....	14
MySQL .....	15
Additional Databases .....	17
Build Source Code .....	17
Set Java Options .....	19
Set Increased JAVA_OPTS Settings .....	19
Put jasperserver.license in Place .....	19

Starting JasperReports Server .....	20
Logging into JasperReports Server .....	20
JasperReports Server Log Files .....	21
<b>Create and Load Sample Data .....</b>	<b>22</b>
Load Sample Data .....	22
Generate Your Own Sample Resources .....	23
Generating Sample Resources for MySQL, PostgreSQL, Oracle, and SQL Server Databases .....	23
Generating Sample Resources for the DB2 Database .....	24
<b>Source Build Setup for Other Databases .....</b>	<b>25</b>
Get Your JDBC Driver .....	25
Download a JDBC Driver .....	25
Set Up Your Database .....	26
Oracle .....	26
SQL Server .....	28
DB2 .....	29
<b>Additional Buildomatic Information .....</b>	<b>32</b>
Detailed Description of the deploy-webapp-pro Target .....	32
Running Ant in Debug Mode .....	32
Regenerate Your Buildomatic Property Settings .....	33
Using Your Own Apache Ant: Get ant-contrib.jar .....	33
Generated Property Files .....	34
Existing and Generated Database SQL Files .....	34
Generated WAR File Location and deploy-webapp-pro Target .....	35
Details on Database Load Build Targets .....	36
create-load-js-db-pro .....	37
create-load-all-dbs-pro .....	37
General Fresh Database Schema File .....	38
Manual Creation of Databases .....	38

Manually Creating Databases: PostgreSQL .....	38
Additional Databases .....	39
<b>Jaspersoft Internal Developers and Advanced Developers .....</b>	<b>40</b>
Internal Developers and Advanced Developers .....	40
Additional Properties in default_master.properties .....	41
Integration Tests .....	43
Location of JavaScript files .....	43
<b>Java Options and JasperServer License Details .....</b>	<b>44</b>
Setting Java JVM Options .....	44
Tomcat and JBoss JVM Options .....	44
Configuring the JasperReports Server License File .....	46
Configuring the License for All Application Servers .....	46
Configure the License in the Tomcat Scripts .....	46
<b>Troubleshooting .....</b>	<b>48</b>
Build Troubleshooting .....	48
Name Undefined Error (Old Ant Version) .....	48
Database Troubleshooting .....	49
Maven Troubleshooting .....	49
Maven Error on Linux or Mac .....	49
Clear JasperReports Server Artifacts in Maven Local Repository .....	50
Clear Entire Local Repository .....	51
Maven Warnings .....	51
Old Maven Binary .....	51
Other Build Troubleshooting .....	52
Error When Building Database Scripts .....	52
<b>Jaspersoft Documentation and Support Services .....</b>	<b>53</b>
<b>Legal and Third-Party Notices .....</b>	<b>55</b>

# Introduction

---

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



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 are licensed to use, or to upgrade your license, contact Jaspersoft.

---

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

- Easily create 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 in PDF format on the [Product Documentation website](#). You can also access PDF and HTML versions of these guides 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.
- The [Jaspersoft Community site](#) covers topics for:
  - Developers
  - System administrators
  - Business users
  - Data integration users
- 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 <https://www.jaspersoft.com/support>.

JasperReports Server is a component of both a community project and commercial offerings. Each of these integrates the standard features such as security, scheduling, web services interface, and much more for running and sharing reports. Commercial editions provide additional features for hosting large BI deployments, including:

- Ad Hoc views and reports
- advanced charts
- dashboards
- domains
- auditing
- multi-organization architecture

This guide assists developers in obtaining, setting up, building, and running JasperReports Server from its source files.



This document describes how to build from a command line shell in Linux or Windows. It does not address the process of building within an IDE (Integrated Development Environment) such as Eclipse or IntelliJ.

## Supported Build Configurations

The following table lists the target configurations that can be built from the source:

Application Server	Database
Tomcat or JBoss	PostgreSQL
	MySQL
	Oracle
	SQL Server
	DB2

## JasperReports Server Source Code Archives and Current Files

The following table lists the source code archive files and current files for JasperReports Server:

File	Description	Documented In
js-jrs_9.0.0_src.zip	JasperReports Server source code	<a href="#">Building JasperReports Server Source Code</a>
js-jrs_9.0.0_repo.zip	JasperReports Server repository files	<a href="#">Building JasperReports Server Source Code</a>

# Components Required for Source Build

---

The components and versions listed in this section are required to build and run JasperReports Server:

- [Check Your Java JDK](#)
- [Check Your Maven Version](#)
- [Check Your Application Server](#)
- [Check Your Database Instance](#)
- [Check Your Chrome/Chromium Installation](#)

## Check Your Java JDK

Compile the JasperReports Server source code under Java 1.8 or Java 11. To check the version of your JDK (Java Development Kit), run the following command:

```
javac -version
```

To download the Java JDK, follow the instructions on the Java website:

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>.

The Oracle/Sun JDK is the certified Java platform for JasperReports Server. OpenJDK 1.8 and 11 are also supported.



IBM JDK 8 is not supported.

---

## Check Your Maven Version

We use Apache Maven 3.9 to compile, build, and package the JasperReports Server source code because of its ability to manage third-party dependencies via online repositories.

To download and install Maven, go to:

<http://maven.apache.org/download.html#installation>

---



To run `mvn` from the command line, put the maven binary (`mvn` or `mvn.exe`) in your environment `PATH`. To check your Maven version, run this command:

```
mvn -version
```

For information about Maven, see [Maven Troubleshooting](#).

## Check Your Application Server

To run JasperReports Server, you need an application server on the same computer as your source code. We support the following application servers:

Application Server	Comments
Apache Tomcat	Source build can automatically deploy to this application server.
JBoss EAP / Wildfly	Source build can automatically deploy to this application server.
WebSphere	JasperReports Server must be manually deployed.
WebLogic	JasperReports Server must be manually deployed.

## Check Your Database Instance

To run JasperReports Server, you need a database instance. We support the following:

Database	Comments
PostgreSQL	Source build automatically creates the <code>jasperserver</code> database.
MySQL	Source build automatically creates the <code>jasperserver</code> database.
Oracle	Source build automatically creates the <code>jasperserver</code> database.
SQL Server	Source build automatically creates the <code>jasperserver</code> database.

---

Database	Comments
DB2	The jasperserver database must be created by the DB administrator.

---

## Check Your Chrome/Chromium Installation

To build JasperReports Server, you need to install Chrome, Chromium, or any other browser based on Chromium like Microsoft Edge, and set the property `chrome.path` in `default_master.properties`. See the JasperReports Server Administrator Guide for more information about Chrome/Chromium.

# Building JasperReports Server Source Code

---



This document describes how to build from a command-line shell in Linux or Windows. It does not address the process of building within an IDE (Integrated Development Environment) such as Eclipse or IntelliJ.

---

## Introduction to Buildomatic Source Build Scripts

The JasperReports Server source code comes with a set of configuration and build scripts based on Apache Ant known as the buildomatic scripts. You will find these scripts in the following directory:

```
<js-src>/jasperserver/buildomatic
```

The buildomatic scripts automate most aspects of configuring, building, and deploying the source code. Apache Ant is bundled into the source code distribution to simplify the setup.

## Downloading and Unpacking JasperReports Server Source Code

### Downloading the Source Archive

Download the source code package zip for the commercial version of JasperReports Server from [Jaspersoft Technical Support](https://www.jaspersoft.com/support) (https://www.jaspersoft.com/support) . The download package is `js-jrs_9.0.0_src.zip`.

For access to the site, contact technical support or your sales representative.

## Unpacking the Source Archive

Unpack the `js-jrs_9.0.0_src.zip` file to a directory location, such as `C:\` or `/home/<user>`. The resulting location is referred to as `<js-src>` in this document.

---

Windows:      `<js-src>` example is `C:\JasperReports-Server-9.0.0-src`

---

Linux:         `<js-src>` example is `/home/<user>/JasperReports-Server-9.0.0-src`

---



The source build may use paths that exceed the 260-character limit on Windows. To extract the package, Enable NTFS long paths (Windows 10 only) or use a third-party file archiver such as 7-Zip.

---

## Source Code Package Structure

After you have unpacked the zip file, the folder directory has the following structure:

Directory or file	Description
<code>&lt;js-src&gt;/apache-ant</code>	Bundled version of Apache Ant build tool
<code>&lt;js-src&gt;/jasperserver</code>	JasperReports Server open source code for core functionality
<code>&lt;js-src&gt;/jasperserver-pro</code>	JasperReports Server source code for commercial functionality
<code>&lt;js-src&gt;/jasperserver-repo</code>	Dependent jar files (not readily available publicly)
<code>&lt;js-src&gt;/jasperserver-ui</code>	JasperReports Server source code for UI

---



The `repo-path` variable can be set to point to the location of the `js-jrs_9.0.0_repo.zip` / `./repository` directory.

---

## Check Apache Ant

The Apache Ant tool is bundled (pre-integrated) into the source code distribution package. So you do not need to download or install Ant to run the buildomatic scripts. For example:

```
cd <js-src>/jasperserver/buildomatic
```

```
js-ant help or
```

```
../js-ant help (Linux)
```

If you do not use the bundled version of Apache Ant, we recommend version 1.9.4 or later. Versions earlier than 1.8.1 are not compatible.

## Configuring the Buildomatic Properties

The buildomatic scripts are found at the following location:

```
<js-src>/jasperserver/buildomatic
```

Use the buildomatic scripts to build the source code and configure settings for a supported application server and database. The file for configuring these settings is `default_master.properties`. The source distribution includes a properties file for each type of database. Add your specific settings to this file and rename it to:

```
default_master.properties
```



When specifying paths with Apache Ant and Java in Windows, a single forward slash (/) normally works the same as “escaped” double backslashes (\\).

---

# PostgreSQL

## Procedure

1. Go to the buildomatic directory in the source distribution:

```
cd <js-src>/jasperserver/buildomatic
```

2. Copy the PostgreSQL specific file to the current directory and change its name to default\_master.properties as shown below:

---

Windows:      copy sample\_conf\postgresql\_master.properties default\_master.properties

---

Linux:         cp sample\_conf/postgresql\_master.properties default\_master.properties

---

3. Edit the new default\_master.properties file and set the following properties for your local environment:

Property	Examples
appServerType	appServerType=tomcat [jboss-eap-7, wildfly, skipAppServerCheck]
appServerDir	appServerDir = C:\\Program Files\\Apache Software Foundation\\Tomcat 9.0 appServerDir = /home/<user>/apache-tomcat-9.0
dbHost	dbHost=localhost
dbUsername	dbUsername=postgres
dbPassword	dbPassword=postgres
maven	maven = C:\\apache-maven-3.9\\bin\\mvn.cmd maven = /home/<user>/apache-maven-3.9/bin/mvn

Property	Examples
js-path	<pre>js-path = C:\\JasperReports-Server-9.0.0-src\\jasperserver</pre> <pre>js-path = /home/&lt;user&gt;/JasperReports-Server-9.0.0-src/jasperserver</pre>
js-pro-path	<pre>js-pro-path = C:\\JasperReports-Server-9.0.0-src\\jasperserver-pro</pre> <pre>js-pro-path = /home/&lt;user&gt;/JasperReports-Server-9.0.0-src/jasperserver-projs-pro-path =</pre>
maven.build.type	<pre>maven.build.type=repo</pre>
repo-path	<pre>repo-path = C:\\JasperReports-Server-9.0.0\\jasperserver-repo</pre> <pre>repo-path = /home/&lt;user&gt;/JasperReports-Server-9.0.0-src/jasperserver-repo</pre>
chrome.path	<pre>chrome.path = C:/Program Files (x86)/Google/Chrome/Application/chrome.exe</pre> <pre>chrome.path = /usr/bin/google-chrome</pre>

## MySQL

### Procedure

1. Go to the buildomatic directory in the source distribution:

```
cd <js-src>/jasperserver/buildomatic
```

2. Copy the MySQL specific file to the current directory and change its name to default\_master.properties:

```
Windows: copy sample_conf\mysql_master.properties default_master.properties
```

---

```
Linux:      cp sample_conf/mysql_master.properties default_
           master.properties
```

---

3. Edit the new `default_master.properties` file and set the following properties to your local environment:

Property	Examples
<code>appServerType</code>	<code>appServerType=tomcat [jboss-eap-7, wildfly, skipAppServerCheck]</code>
<code>appServerDir</code>	<code>appServerDir = C:\\Program Files\\Apache Software Foundation\\Tomcat 9.0</code> <code>appServerDir = /home/&lt;user&gt;/apache-tomcat-9.0</code>
<code>dbHost</code>	<code>dbHost = localhost</code>
<code>dbUsername</code>	<code>dbUsername = root</code>
<code>dbPassword</code>	<code>dbPassword = password</code>
<code>maven</code>	<code>maven = C:\\apache-maven-3.9\\bin\\mvn.cmd</code> <code>maven = /home/&lt;user&gt;/apache-3.9/bin/mvn</code>
<code>js-path</code>	<code>js-path = C:\\JasperReports-Server-9.0.0-src\\jasperserver</code> <code>js-path = /home/&lt;user&gt;/JasperReports-Server-9.0.0-src/jasperserver</code>
<code>js-pro-path</code>	<code>js-pro-path = C:\\JasperReports-Server-9.0.0-src\\jasperserver-pro</code> <code>js-pro-path = /home/&lt;user&gt;/JasperReports-Server-9.0.0-src/jasperserver-pro</code>
<code>maven.build.type</code>	<code>maven.build.type=repo</code>
<code>repo-path</code>	<code>repo-path = C:\\JasperReports-Server-9.0.0-src\\jasperserver-</code>



Property	Examples
	repo  repo-path = /home/<user>/JasperReports-Server-9.0.0-src/jasperserver-repo
chrome.path	chrome.path = C:/Program Files (x86)/Google/Chrome/Application/chrome.exe  chrome.path = /usr/bin/google-chrome

## Additional Databases

For `default_master.properties` configurations for other databases, please see [Source Build Setup for Other Databases](#).

## Build Source Code

Now that you have set up your `default_master.properties` file, you can build the source code.

To build JasperReports Server

### Procedure

1. Set up the `default_master.properties` file for your environment (as described above).
2. Start the database server.
3. Stop the application server.
4. Run the commands shown below:

After running each Ant target in [Commands for Building JasperReports Server](#), look for the message BUILD SUCCESSFUL.

### Commands for Building JasperReports Server

Commands	Description
<code>cd &lt;js-src&gt;/jasperserver/buildomatic</code>	
<code>js-ant clean-config</code>	(Optional) Clears the buildomatic/build_conf/default directory.
<code>js-ant gen-config</code>	(Optional) Rebuilds the buildomatic/build_conf/default directory.
<code>js-ant add-jdbc-driver</code>	Used for loading the databases
<code>js-ant build-ce</code>	Builds the community source code
<code>js-ant build-pro</code>	Builds the commercial source code
<code>js-ant create-load-js-db-pro</code>	(Optional) Creates and loads the jasperserver database, imports core bootstrap data
<code>js-ant deploy-webapp-pro</code>	(Optional) Deploys the jasperserver-pro war file to the application server
<code>js-ant deploy-jrws</code>	(Optional) Deploys only JasperReports Web Studio apps in the configured app path in default_master.properties.



Installing JasperReports Server automatically generates encryption keys that reside on the file system. These keys are stored in a dedicated Jaspersoft keystore. Make sure that this keystore is properly secured and backed up, as described in the *JasperReports Server Security Guide*.

## Set Java Options

JasperReports Server needs Java memory options that are larger than the standard defaults. For information about additional Java options, see [Setting Java JVM Options](#).

## Set Increased JAVA\_OPTS Settings

JasperReports Server needs greater heap settings for all functionality to operate. For testing your deployed JasperServer, set your JAVA\_OPTS to the same default values described in the JasperReports Server Installation Guide. The following shows the minimum recommended settings; you may need to increase these according to your usage.

### JVM Options on Linux and Mac OSX (64 bit)

Options for all app servers	<code>export JAVA_OPTS="\$JAVA_OPTS -Xms2048m -Xmx4096m -Xss2m"</code>
-----------------------------	--

### JVM Options on Windows (64 bit)

Options for all app servers	<code>set JAVA_OPTS=%JAVA_OPTS% -Xms2048m -Xmx4096m -Xss2m</code>
-----------------------------	---

Add these settings to your application server startup script:

Apache Tomcat:	<code>&lt;tomcat&gt;/bin/setclasspath.sh</code>	(.bat for Windows)
----------------	---	--------------------

JBoss:	<code>&lt;jboss&gt;/bin/standalone.conf</code>	(.bat for Windows)
--------	--	--------------------

For details on setting Java memory options, please see [Setting Java JVM Options](#).

## Put jasperserver.license in Place

JasperReports Server Commercial edition requires a license to run. An evaluation license is provided in the source code zip download package. You can use this evaluation license to

get started and then replace it with one you request [Jaspersoft Technical Support](https://www.jaspersoft.com/support) (<https://www.jaspersoft.com/support>) or from your sales representative.

JasperReports Server looks for the license file in the home directory of the user running the application server, so copy the license to that location. You will find the license in the root of the source package:

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

For more information on license configuration, please see [Configuring the JasperReports Server License File](#).

Copy `jasperserver.license` to the appropriate folder listed in the table below.

#### License Locations

Operating System	
Linux	/home/<user>/
Mac OSX	/Users/<user>/
Windows 7 using the bundled Tomcat	C:\Users\<user>\
Windows 7 using an existing Tomcat Windows service	C:\

## Starting JasperReports Server

You can now start your application server. Your database should already be running.

## Logging into JasperReports Server

You can now log into JasperReports Server through a web browser:

Enter the login URL with the default port number:

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

Log into JasperReports Server as superuser or jasperadmin:

User ID: superuser      Password: superuser

User ID: jasperadmin Password: jasperadmin

If you are unable to log in or have other problems, see [Troubleshooting](#), or the JasperReports Server Installation Guide.

## JasperReports Server Log Files

If you encounter any startup or runtime errors, you can check the application server log files. For Apache Tomcat, you will find the log file here:

```
<tomcat>/logs/catalina.out
```

Also check the `jasperserver.log` file. You can increase the debug output level by editing the `log4j.properties` file.

The JasperReports Server runtime log is here:

```
<tomcat>/webapps//WEB-INF/logs/jasperserver.log
```

The `log4j.properties` file is here:

```
<tomcat>/webapps//WEB-INF/log4j.properties
```

# Create and Load Sample Data

---

The procedure for [Building JasperReports Server Source Code](#) loads core data required to start the application, but it doesn't create sample data, such as sample reports to run and sample databases. Follow the steps below to create and load sample data.

## Load Sample Data

The buildomatic scripts can load sample resources and sample databases. Note: In the procedure below, your `jasperserver` database will be deleted and re-created unless you choose 'n' for No when prompted.

Your `default_master.properties` should already be created.

### Procedure

1. Start your database server.
2. Stop your application server.
3. Run the commands shown below:

Commands	Description
<pre>cd &lt;js- src&gt;/jasperserver/buildomatic</pre>	
<pre>js-ant create-load-all-dbs-pro</pre>	<p>Creates and loads the <code>jasperserver</code> database</p> <p>Imports core bootstrap resources</p> <p>Creates and loads sample databases</p> <p>Imports sample resources</p> <p>(Choose 'n' when prompted if you do not</p>

---

Commands	Description
	want to recreate your jasperserver database.)

## Generate Your Own Sample Resources

This procedure generates the same resources imported and used by the released version of JasperReports Server. Your jasperserver database will be deleted and re-created.

Your `default_master.properties` should have already been created.

## Generating Sample Resources for MySQL, PostgreSQL, Oracle, and SQL Server Databases

### Procedure

1. Make sure that the sample data has been loaded.
2. Start your database server.
3. Stop your application server.
4. Run the commands shown below:

Commands	Description
<code>cd &lt;js-src&gt;/jasperserver/buildomatic</code>	
<code>js-ant re-init-js-db-pro</code>	Drop, create, and initialize the jasperserver database.
<code>js-ant run-production-data-pro</code>	Generate sample resources using the processing logic from the production-test source code.

## Generating Sample Resources for the DB2 Database

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.

### Procedure

1. Start your database server.
2. Stop your application server.
3. If the database already exists, remove it before creating a new one.
4. Run the commands shown below:

Commands	Description
<code>db2 create database jsprsvr using codeset utf-8 territory us pagesize 16384</code>	Create and initialize the jasperserver database.
<code>db2 create database sugarcrm</code>	Optionally, create the sugarcrm database.
<code>db2 create database foodmart</code>	Optionally, create the foodmart database.



# Source Build Setup for Other Databases

---

You can use the example settings below for Oracle, SQL Server, and DB2.

## Get Your JDBC Driver

You can choose to use the provided JDBC driver or download a native JDBC Driver.

## Download a JDBC Driver

You can download a JDBC driver appropriate for your database. In this case, additional configurations are required. You can download a JDBC driver from one of these vendor sites:

- <http://www.oracle.com/technetwork/indexes/downloads> (Oracle)
- <https://www.microsoft.com/en-us/download/details.aspx?id=56615> (SQL Server)
- <http://www-01.ibm.com/software/data/db2/linux-unix-windows/downloads.html> (DB2)

Copy the downloaded JDBC jar to the following location:

- `<js-src>/buildomatic/conf_source/db/<dbType>/jdbc`

For example, for SQL Server the driver would go here:

- `<js-src>/buildomatic/conf_source/db/sqlserver/jdbc`

# Set Up Your Database

## Oracle

### Procedure

1. Go to the buildomatic directory in the source distribution:  

```
cd <js-src>/jasperserver/buildomatic
```
2. Copy the Oracle specific file to the current directory and change its name to `default_master.properties`:

---

Windows: `copy sample_conf\oracle_master.properties default_master.properties`

---

Linux: `cp sample_conf/oracle_master.properties default_master.properties`

---

3. Open the new `default_master.properties` file for editing.
4. Set the following properties for your local environment:

Property	Examples
<code>appServerType</code>	<code>appServerType=tomcat [jboss-eap-7, wildfly, skipAppServerCheck]</code>
<code>appServerDir</code>	<code>appServerDir = C:\\Program Files\\Apache Software Foundation\\Tomcat 9.0</code> <code>appServerDir = /home/&lt;user&gt;/apache-tomcat-9.0</code>
<code>sysUsername</code>	<code>sysUsername=system</code>
<code>sysPassword</code>	<code>sysPassword=password</code>
<code>dbUsername</code>	<code>dbUsername=jasperserver</code>

Property	Examples
dbPassword	dbPassword=password
dbHost	dbHost=localhost
maven	maven = C:\\apache-maven-3.9\\bin\\mvn.cmd maven = /home/<user>/apache-maven-3.9/bin/mvn
js-path	js-path = C:\\JasperReports-Server-9.0.0-src\\jasperserver js-path = /home/<user>/JasperReports-Server-9.0.0-src/jasperserver
js-pro-path	js-pro-path = C:\\JasperReports-Server-9.0.0-src\\jasperserver-pro js-pro-path = /home/<user>/JasperReports-Server-9.0.0-src/jasperserver-pro
repo-path	repo-path = C:\\JasperReports-Server-9.0.0-src\\jasperserver-repo repo-path = /home/<user>/JasperReports-Server-9.0.0-src/jasperserver-repo
chrome.path	chrome.path = C:/Program Files (x86)/Google/Chrome/Application/chrome.exe chrome.path = /usr/bin/google-chrome

5. Save the default\_master.properties file.

# SQL Server

## Procedure

1. Go to the buildomatic directory in the source distribution:  

```
cd <js-src>/jasperserver/buildomatic
```
2. Copy the SQL Server specific file to the current directory and change its name to `default_master.properties`:

---

Windows: `copy sample_conf\sqlserver_master.properties default_master.properties`

---

Linux: `cp sample_conf/sqlserver_master.properties default_master.properties`

---

3. Edit the new `default_master.properties` file and set the following properties for your local environment:

Property	Examples
<code>appServerType</code>	<code>appServerType=tomcat [jboss-eap-7, wildfly, skipAppServerCheck]</code>
<code>appServerDir</code>	<code>appServerDir = C:\\Program Files\\Apache Software Foundation\\Tomcat 9.0</code> <code>appServerDir = /home/&lt;user&gt;/apache-tomcat-9.0</code>
<code>dbUsername</code>	<code>dbUsername=sa</code>
<code>dbPassword</code>	<code>dbPassword=sa</code>
<code>dbHost</code>	<code>dbHost=localhost</code>
<code>js-path</code>	<code>js-path = C:\\JasperReports-Server-9.0.0-src\\jasperserver</code> <code>js-path = /home/&lt;user&gt;/JasperReports-Server-9.0.0-src/jasperserver</code>

Property	Examples
js-pro-path	<pre>js-pro-path = C:\\JasperReports-Server-9.0.0-src\\jasperserver-pro</pre> <pre>js-pro-path = /home/&lt;user&gt;/JasperReports-Server-9.0.0-src/jasperserver-pro</pre>
repo-path	<pre>repo-path = C:\\JasperReports-Server-9.0.0-src\\jasperserver-repo</pre> <pre>repo-path = /home/&lt;user&gt;/JasperReports-Server-9.0.0-src/jasperserver-repo</pre>
chrome.path	<pre>chrome.path = C:/Program Files (x86)/Google/Chrome/Application/chrome.exe</pre> <pre>chrome.path = /usr/bin/google-chrome</pre>

4. Save the default\_master.properties file.

## DB2

### Procedure

1. Go to the buildomatic directory in the source distribution:  

```
cd <js-src>/jasperserver/buildomatic
```
2. Copy the DB2 specific file to the current directory and change its name to default\_master.properties:

---

Windows: 

```
copy sample_conf\db2_master.properties default_master.properties
```

---

Linux: 

```
cp sample_conf/db2_master.properties ./default_master.properties
```

---

3. Edit the new default\_master.properties file and set the following properties for your local environment:

Property	Examples
appServerType	appServerType=tomcat [jboss-eap-7, wildfly, skipAppServerCheck]
appServerDir	appServerDir = C:\\Program Files\\Apache Software Foundation\\Tomcat 9.0 appServerDir = /home/<user>/apache-tomcat-9.0
dbUsername	dbUsername=db2admin
dbPassword	dbPassword=password
dbHost	dbHost=localhost
js-path	js-path = C:\\JasperReports-Server-9.0.0-src\\jasperserver js-path = /home/<user>/JasperReports-Server-9.0.0-src/jasperserver
js-pro-path	js-pro-path = C:\\JasperReports-Server-9.0.0-src\\jasperserver-pro js-pro-path = /home/<user>/JasperReports-Server-9.0.0-src/jasperserver-pro
repo-path	repo-path = C:\\JasperReports-Server-9.0.0-src\\jasperserver-repo repo-path = /home/<user>/JasperReports-Server-9.0.0-src/jasperserver-repo
chrome.path	chrome.path = C:/Program Files (x86)/Google/Chrome/Application/chrome.exe chrome.path = /usr/bin/google-chrome

4. 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
```

5. Save the default\_master.properties file.



For DB2, the database must be created manually, because it is not possible to use a JDBC call to automatically create a database on DB2.

---

## Additional Buildomatic Information

---

The Ant-based buildomatic scripts contain support files for the setup and configuration of a number of databases and application servers. This chapter gives the locations of many of these files.

### Detailed Description of the `deploy-webapp-pro` Target

The `deploy-webapp-pro` target performs the following actions in your application server environment:

- Deletes any existing `jasperserver-pro` WAR file.
- Copies the JDBC driver to the appropriate application server directory.
- Copies additional JDBC drivers to the application server to support data source creation in the UI.
- Adds a data source definition to the appropriate application server directory.
- Deploys the newly built `jasperserver-pro` WAR file.
- Deletes files within the application server work directory (to clear out compiled JSP files and other cached files).
- On Tomcat, if present, deletes the old version of `<tomcat>/conf/Catalina/Localhost/jasperserver-pro.xml`.

### Running Ant in Debug Mode

Ant can be run with a `-v` (verbose) or a `-d` (debug) option to help with troubleshooting, for example:

```
js-ant -v build-pro
```



## Regenerate Your Buildomatic Property Settings

If you change your `default_master.properties` file, buildomatic automatically cleans and regenerates all configuration settings. To explicitly clean and regenerate your settings manually, run the following commands:

Commands	Description
<code>js-ant clean-config</code>	Clears the <code>buildomatic/build_conf/default</code> directory.
<code>js-ant gen-config</code>	Rebuilds the <code>buildomatic/build_conf/default</code> directory.



Anytime you modify the `default_master.properties` file, the configuration settings are automatically regenerated into the `buildomatic/build_conf/default` folder.

## Using Your Own Apache Ant: Get ant-contrib.jar

If you prefer to use your own version of Apache Ant, get the file `ant-contrib-1.0b3.jar`. This JAR enables conditional logic in Ant scripts.

### Procedure

1. Make sure you are using Apache Ant 1.9.4 or higher.
2. Copy the file `ant-contrib-1.0b3.jar` from the `<js-src>/apache-ant/lib` folder to your `<ant-home>/lib` folder:

From:

`<js-src>/apache-ant/lib/ant-contrib.jar`

or

`<js-src>/jasperserver/buildomatic/install_resources/extra-jars/ant-contrib.jar`

To:

`<ant-home>/lib`

(General example)

---

C:\apache-ant-1.9.4\lib	(Windows example)
/usr/share/java/apache-ant/lib	(Linux example)
/usr/share/ant/lib	(Mac example)

---

## Generated Property Files

Once your database and application server property values are set, run buildomatic scripts to generate the database and application server configuration files to run JasperReports Server. Generated property files are in the following directory:

```
<js-src>/jasperserver/buildomatic/build_conf/default
```

Some of the key configuration files are:

```
js.jdbc.properties
```

```
maven_settings.xml - (This is the maven settings file used by the source build)
```

More generated property files are in the following directory:

```
<js-src>/jasperserver/buildomatic/build_conf/default/webapp
```

Some of the configuration files in this directory are:

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

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

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

Running `clean-config` removes these generated files. Running `gen-config` or any other target, regenerates these files.

## Existing and Generated Database SQL Files

Buildomatic files that support various databases are located in:

```
<js-src>/jasperserver/buildomatic/install_resources/sql/<db-type>
```

The source code build procedure creates the `jasperserver` repository database schema using these files:

- `js-pro-create.ddl`
- `js-pro-drop.ddl`

When you run the buildomatic target `build-js-ddl-pro`, these database files are freshly generated for your specified database platform. The files are generated to the following location:

```
<js-src>/jasperserver-pro/repository-hibernate/build-db/target/sql
```

Then the files are automatically copied into their buildomatic directory location:

```
<js-src>/jasperserver/buildomatic/install_resources/sql/<db-type>
```



These generated files also overwrite the ones already in the buildomatic directory location.

---

## Generated WAR File Location and deploy-webapp-pro Target

The JasperReports Server source code build creates a `jasperserver-pro` WAR file. The build assembles the WAR file into the following location:

```
<js-src>/jasperserver-pro/jasperserver-war/target
```

When the `build-pro` target is run, buildomatic assembles the `jasperserver-pro` WAR file, and copies the file to this location for use by subsequent buildomatic targets:

```
<js-src>/jasperserver/buildomatic/install_resources/war/jasperserver-pro
```

Later, when you run the buildomatic target `deploy-webapp-pro`, the following actions take place, for example on Tomcat:

---

Files:        `<js-src>/jasperserver/buildomatic/install_resources/war/jasperserver-pro/*`

---

Copied to:    `<tomcat>/webapps`

---

File:        `<js-src>/jasperserver/buildomatic/build_conf/default/webapp/META-`

---

	INF/context.xml
Copied to:	<tomcat>/webapps/jasperserver/jasperserver-pro/META-INF
Files:	<js-src>/jasperserver/buildomatic/build_conf/default/webapp/WEB-INF/classes/hibernate.properties
Copied to:	<tomcat>/webapps/jasperserver-pro/WEB-INF/classes
Files:	<js-src>/jasperserver/buildomatic/build_conf/default/webapp/WEB-INF/js.quartz.properties
Copied to:	<tomcat>/webapps/jasperserver-pro/WEB-INF/
File:	<js-src>/jasperserver/buildomatic/build_conf/db/postgresql/jdbc/postgresql-9.2-1002.jdbc4.jar
Copied to:	<tomcat>/lib
Files:	<js-src>/jasperserver/buildomatic/conf_source/db/app-srv-jdbc-drivers/*/jar
Copied to:	<tomcat>/lib

## Details on Database Load Build Targets

The buildomatic targets shown below are used in [Building JasperReports Server Source Code](#) to create and populate the databases used with JasperReports Server. These targets consolidate and simplify the handling of the jasperserver database and the optional sample databases:

- `create-load-js-db-pro`
- `create-load-all-dbs-pro`

## create-load-js-db-pro

This buildomatic target is a consolidation of the following targets:

- `drop-js-db` (if necessary)
- `create-js-db`
- `init-js-db-pro`
- `import-minimal-pro`

Additional functionality determines whether the `jasperserver` database exists. If so, a command-line prompt asks you if you want to delete and re-create the database.

## create-load-all-dbs-pro

This buildomatic target is a consolidation of the following targets:

- `drop-js-db` (if necessary)
- `create-js-db`
- `init-js-db-pro`
- `import-minimal-pro`
- `import-sample-data-pro`
- `(drop-foodmart-db, if necessary)`
- `create-foodmart-db`
- `load-foodmart-db`
- `(drop-sugarcrm-db, if necessary)`
- `create-sugarcrm-db`
- `load-sugarcrm-db`

Additional functionality determines whether the `jasperserver` database exists. If so, a command-line prompt asks you if you want to delete and re-create the database. The same logic applies for the sample databases: `foodmart` and `sugarcrm`.

## General Fresh Database Schema File

The consolidated database scripts do not regenerate the database schema file. Instead the existing default database schema files are used. To regenerate the database schema files, run the following target:

```
js-ant build-js-ddl-pro
```

The files are generated to the following location:

```
<js-src>/jasperserver-pro/repository-hibernate/build-db/target/sql
```

Then the files are automatically copied into their buildomatic directory location:

```
<js-src>/jasperserver/buildomatic/install_resources/sql/<db-type>
```

## Manual Creation of Databases

JasperReports Server runs with a repository database typically named `jasperserver`. The automated buildomatic steps create the `jasperserver` database and sample databases. But you can also create your databases manually.

## Manually Creating Databases: PostgreSQL

You can manually run the scripts buildomatic uses to create and populate databases. Here is an example for PostgreSQL:

### Procedure

1. To create the `jasperserver` database, use a client tool to log into PostgreSQL:

```
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. To create the sample databases, run these commands:

```
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-postgresql.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
```

## Additional Databases

For information on manual setup of databases other than PostgreSQL, refer to the *JasperReports Server Installation Guide*.

# Jaspersoft Internal Developers and Advanced Developers

---

This chapter is for Jaspersoft Internal Developers and for Advanced Developers who want to use some of the additional options available through the buildomatic property settings.

## Internal Developers and Advanced Developers

Jaspersoft provides a Maven repository using [Artifactory](#) to hold all third-party components required to build the server's source code. It also acts as a proxy for the standard public Maven repositories such as [repo1.maven.org](#).

This internal repository is convenient for internal Jaspersoft developers because the developer can point to one location to get all dependencies resolved.

In `default_master.properties`, internal developers should comment out `maven.build.type=repo` and `repo-path=<path>`:

```
# maven.build.type=repo
# repo-path=<path>
```

External developers (customers) who download the `js-jrs_9.0.0_src.zip` package from [jaspersoft.com](#) should set all the properties described in [Configuring the Buildomatic Properties](#) before building JasperReports Server.

Additional buildomatic property settings are available for advanced external developers. If you are an external developer working within an enterprise or on a project that has an internal Maven repository server, you can use the `mirror` value. The following property settings and values enable a local Maven repository:

```
maven.build.type=mirror
mvn-mirror=<repo-url>
```

If you are an external developer with other build configurations to add, you can do this with the `maven.build.type=custom` property setting. If you set this value, the following file will be used as the template to set up the JasperReports Server build configuration:

```
<js-path>/buildomatic/conf_source/templates/maven_settings_custom.xml
```



You can edit this file to add whatever configurations you want.

When buildomatic auto-setup is complete, you can see the final maven settings file used for the JasperReports Server here:

```
<js-path>/buildomatic/build_conf/default/maven_settings_custom.xml
```

## Additional Properties in default\_ master.properties

You can use the properties in the table below for various customizations of the JasperReports Server build:

Property Setting	Purpose
SKIP_TEST_ARG=skipTests	Enable this property to skip unit test execution. This speeds the source build.
VERBOSE_LOGGING=true	Enable this property to increase the INFO logging from the Maven package.
OFFLINE_ARG=-o	Enable this property if you want to build in “offline” mode. To run in offline mode, you need to have successfully built JasperServer at least once.
SKIP_EXPORT_FILES=true	Enable this property to skip the copying of files that set up the command-line import-export configuration. This saves time on file copying.
maven.build.type=repo	Use this setting for the build type if you have downloaded the source code zip package from the jaspersoft.com site, and you are building the source code as a customer (external developer) would build it. You will also need to set the repo-path property. maven.build.type=repo is the default value used in the sample <dbType>_master.properties files.
maven.build.type=community	Use this setting for the build type if you are building only

Property Setting	Purpose
	<p>the Community source code. This setting supports Community members who have checked out the JasperReports Server source code from the Community site: <a href="http://code.jaspersoft.com/svn/repos/jasperserver">code.jaspersoft.com/svn/repos/jasperserver</a></p>
<code>maven.build.type=mirror</code>	<p>If you are an external developer who has a central Maven style repository for your enterprise or project, you can use this setting to specify the local central repository. If you set this property value, you should also set the <code>mvn-mirror</code> property.</p>
<code>maven.build.type=custom</code>	<p>If you are an external developer whose build requires additional configurations, you can use this property value to support them. In this case, use the following template file:</p> <pre data-bbox="654 898 1247 968">&lt;js-path&gt;/buildomatic/conf_source/templates/maven_settings_custom.xml</pre> <p>You can manually edit this file to add more configurations. The file will be processed by buildomatic and copied to its final location after running a buildomatic command:</p> <pre data-bbox="654 1140 1414 1173">buildomatic/build_conf/default/maven_settings_custom.xml</pre>
<code>mvn-mirror=&lt;repo-url&gt;</code>	<p><code>mvn-mirror=http://mvnrepo.jaspersoft.com:8081/artifactory/repo</code></p> <p>The value shown is the default repo-url used by Jaspersoft internal development.</p>
<code>repo-path=&lt;path&gt;</code>	<p>Set a local path value for this property if you are using <code>maven.build.type=repo</code> (this is the default configuration from the source code zip download from <a href="http://jaspersoft.com">jaspersoft.com</a>).</p>

## Integration Tests

To run optional integration tests, make sure that Chrome, Chromium, or any other browser based on Chromium like Microsoft Edge is installed and executable, then use the following commands:

### Commands for Running Integration Tests

- `js-ant drop-js-db`
- `js-ant create-js-db`
- `js-ant init-js-db-pro`
- `js-ant run-integration-tests-pro`

## Location of JavaScript files

In the current version of JasperReports Server, the JavaScript-related files are in the following directory:

```
<js-pro-path>/jasperserver-war/src/main/webapp/scripts/runtime_dependencies/jrs-  
ui/src
```

This location may be different in earlier versions. See the JasperReports Server Source Build Guide for your version for more information.



If you customize JavaScript files, you need to optimize the JavaScript after your edits are complete. See the JasperReports Server Ultimate Guide for more information.

---

# Java Options and JasperServer License Details

---

## Setting Java JVM Options

To run properly, JasperReports Server needs more Java memory than the default settings. But for development work, the settings can be simpler than those recommended for production. For full information on recommended JAVA\_OPTS settings, see the JasperReports Server Installation Guide.

## Tomcat and JBoss JVM Options

Here are some typical settings for JVM options that affect JasperReports Server. For space reasons, some of the options are displayed on multiple lines; make sure you set all options. These are the minimum recommended options; you may need to increase the JVM memory assignment according to your usage.

### JVM Options on Windows

Options for all app servers	set JAVA_OPTS=%JAVA_OPTS% -Xms2048m -Xmx4096m set JAVA_OPTS=%JAVA_OPTS% -Xss2m -XX:+UseConcMarkSweepGC
-----------------------------	---

Java 11	set JAVA_OPTS=%JAVA_OPTS% -Djava.locale.providers=COMPAT
---------	--

For Oracle	set JAVA_OPTS=%JAVA_OPTS% -Doracle.jdbc.defaultNChar=true
------------	---

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

Options for all app servers	<pre>export JAVA_OPTS="\$JAVA_OPTS -Xms2048m -Xmx4096m" export JAVA_OPTS="\$JAVA_OPTS -Xss2m" export JAVA_OPTS="\$JAVA_OPTS -XX:+UseConcMarkSweepGC"</pre>
Java 11	<pre>export JAVA_OPTS="\$JAVA_OPTS -Djava.locale.providers=COMPAT"</pre>
For Oracle	<pre>export JAVA_OPTS="\$JAVA_OPTS -Doracle.jdbc.defaultNChar=true"</pre>

You can set JVM options in a number of ways. For example, you can add your `JAVA_OPTS` settings to these files:

File	Add JVM Options Below the Lines Shown Here:
<tomcat>\bin\setclasspath.bat	<pre>set JAVA_ENDORSED_ DIRS=%BASEDIR%\common\endorsed</pre>
<tomcat>/bin/setclasspath.sh	<pre>JAVA_ENDORSED_DIRS="\$BASEDIR"/common/endorsed</pre>
<tomcat>/bin/setenv.bat or <tomcat>/bin/setenv.sh	<p>JAVA_OPTS setting can go anywhere in this file.</p>
<jboss>\bin\standalone.conf.bat <jboss>/bin/standalone.conf	<pre>set JAVA_OPTS=%JAVA_OPTS% - Dprogram.name=%PROGNAME% or export JAVA_OPTS="\$JAVA_OPTS - Dprogram.name=\$PROGNAME"</pre>



For information on recommended `JAVA_OPTS` settings for all certified application servers, please refer to the JasperReports Server Installation Guide.

# Configuring the JasperReports Server License File

Commercial editions of JasperReports Server require a license file. The source code includes an evaluation license. You can use this license or replace it with the one you received from [Jaspersoft Technical Support](https://www.jaspersoft.com/support) (https://www.jaspersoft.com/support) or your sales representative.

## Configuring the License for All Application Servers

The main source build section describes placing a license in the home folder of the user running the application server. See [Put jasperserver.license in Place](#).

## Configure the License in the Tomcat Scripts

If you would like to locate your jasperserver.license in a specific folder, you can set a Java property in the shell script files used to control Tomcat.

JasperReports Server will look for a property named `js.license.directory` and use that folder as the location to find the `jasperserver.license` file.

For instance, if you want to point JasperReports Server to the license file in the root of the source package, update the following shell script:

---

Windows:     `<tomcat>\bin\setclasspath.bat`

---

Linux:        `<tomcat>/bin/setclasspath.sh`

---

And you could update the file with the following setting:

---

Windows:     `set JAVA_OPTS=%JAVA_OPTS% "-Djs.license.directory=<js-src>"`

---

Linux:        `export JAVA_OPTS=$JAVA_OPTS -Djs.license.directory="<js-src>"`

---

The `jasperserver.license` file can reside anywhere on the file system that's accessible from your application server. The `js.license.directory` setting should point to the folder containing the `jasperserver.license`.



You'll find more information on configuring the `jasperserver.license` for all certified application servers in the JasperReports Server Installation Guide.

---

# Troubleshooting

---

## Build Troubleshooting

### Name Undefined Error (Old Ant Version)

We recommend Apache Ant version 1.9.4 or higher.

If you are not using the version of Apache Ant included with the JasperReports Server source code package, you could get the following error when running the buildomatic scripts:

```
BUILD FAILED
c:\js-builds\jasperserver\buildomatic\install.xml:6: Problem: failed to
create task or type if
Cause: The name is undefined.
Action: Check the spelling.
Action: Check that any custom tasks/types have been declared.
Action: Check that any <presetdef>/<macrodef> declarations have taken
place.
```

---

### Solution

The buildomatic scripts require Ant version 1.8.1 or higher, and the ant-contrib.jar file needs to be included in your ant/lib directory. We recommend Ant version 1.9.4. If you are running with your own Ant version, you can copy this jar to your <ant-home>/lib directory:

From:

```
<js-src>/apache-ant/lib/ant-contrib.jar
```

or

```
<js-src>/jasperserver/buildomatic/extra-jars/ant-contrib.jar
```

To:



---

<code>&lt;ant-home&gt;/lib</code>	(General example)
<code>C:\apache-ant-1.9.4\lib</code>	(Windows example)
<code>/usr/share/java/apache-ant/lib</code>	(Linux example)
<code>/usr/share/ant/lib</code>	(Mac example)

---

## Database Troubleshooting

The most common errors encountered when building JasperReports Server involve the database connection. For information about database connection problems, see the Troubleshooting Appendix of the JasperReports Server Installation Guide.

## Maven Troubleshooting

### Maven Error on Linux or Mac

If Maven is installed on Linux via rpm, apt-get, or yum (or on Mac), the Maven binary and the Maven libraries are probably in separate locations. This can potentially cause a problem with the source build.

### `/usr/boot` Does Not Exist Error

When building under Linux or Mac, you may get an error similar to the following:

```
BUILD FAILED
/home/devuser/js-builds/jasperserver/buildomatic/bin/dev.xml:91:
/usr/boot does not exist
```

The Buildomatic scripts attempt to find the MAVEN\_HOME setting and can be unsuccessful when the maven binary is installed in the `/usr/bin/mvn` location. The workaround is to update your

`default_master.properties` file:

```
cd <js-src>/jasperserver/buildomatic
edit default_master.properties
```

Uncomment the `maven.home` line so that it looks like this:

```
maven.home = /usr/share/maven2           (Linux)
```

For Mac, the location of the Maven library files is slightly different:

```
maven.home = /usr/share/maven           (Mac)
```

## Clear JasperReports Server Artifacts in Maven Local Repository

If you add new code to an existing source build environment, such as a bug fix source patch update, you can clear the JasperReports Server artifacts in your Maven local repository to ensure that the newly built artifacts contain the necessary new content. Maven updates the artifacts automatically, but if you have trouble building or pulling in the modified code, you can try deleting these artifact trees.

To clear existing JasperReports Server artifacts

### Procedure

1. Go to the repository directory:

```
cd <home-dir-path>/m2/repository
```

2. Remove the old versions by deleting the following directories and their contents:

---

<code>com/jaspersoft:</code>	Community Project artifact tree
------------------------------	---------------------------------

---

<code>jaspersoft:</code>	Commercial version artifact tree
--------------------------	----------------------------------

---

## Clear Entire Local Repository

If you want to rebuild everything, remove all the cached jars in your Maven local repository. To do this, you can delete (or rename) the entire local repository.

Then when you build JasperServer, all dependencies are re-downloaded.

```
cd <home-dir-path>/m2
rm -rf repository
```



These steps must be performed if you are using the js-jrs\_9.0.0\_repo.zip package.

---

## Maven Warnings

Maven generates verbose warnings during the artifact validation process. For example, the following warning was generated, even though the required JAR file was downloaded successfully:

```
[WARNING] Unable to get resource from repository jasperServer
(file:///C:/svn/js-buildlds/jasperserver-repo
Downloading: http://repo1.maven.org/maven2/commons-logging/commons-
logging/1.0/commons-logging-1.0.pom
163b downloaded
```

## Old Maven Binary

In general, it is best to use the most current stable version of the Maven tool. We recommend the Apache Maven version 3.9.

# Other Build Troubleshooting

## Error When Building Database Scripts

When compiling in the `jasperserver-repository-hibernate/build-db` directory, you might see an error containing the following message:

```
[ERROR] BUILD ERROR
[INFO] -----
-----
[INFO] Error executing ant tasks
Embedded error: Source file does not exist!
```

The most likely problem is that your `.m2/settings.xml` file does not point to the correct source location, and the build step did not find the Quartz scripts. The `settings.xml` file should contain the path to the quartz script corresponding to your database, for example:

```
<js.quartz.script>/home/<user>/<js-src>/jasperserver/scripts/quartz/tables_
<database>.sql</js.quartz.script>
```

If you use the `buildomatic` script, you should not get this kind of error.

# Jaspersoft Documentation and Support Services

---

For information about this product, you can read the documentation, contact Support, and join Jaspersoft Community.

## How to Access Jaspersoft Documentation

Documentation for Jaspersoft products is available on the [Product Documentation website](#), mainly in HTML and PDF formats.

The [Product Documentation website](#) is updated frequently and is more current than any other documentation included with the product.

## Product-Specific Documentation

The documentation for this product is available on the [JasperReports® Server Product Documentation](#) page.

## How to Access Related Third-Party Documentation

When working with JasperReports® Server, you may find it useful to read the documentation of the following third-party products:

## How to Contact Support for Jaspersoft Products

You can contact the Support team in the following ways:

- To access the Support Knowledge Base and getting personalized content about products you are interested in, visit our [product Support website](#).
- To create a Support case, you must have a valid maintenance or support contract with a Cloud Software Group entity. You also need a username and password to log in to the [product Support website](#). If you do not have a username, you can request one by clicking **Register** on the website.

## How to Join Jaspersoft Community

Jaspersoft Community is the official channel for Jaspersoft customers, partners, and employee subject matter experts to share and access their collective experience. Jaspersoft Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from Jaspersoft products. In addition, users can submit and vote on feature requests from within the [Jaspersoft Ideas Portal](#). For a free registration, go to [Jaspersoft Community](#).

# Legal and Third-Party Notices

---

SOME CLOUD SOFTWARE GROUP, INC. (“CLOUD SG”) SOFTWARE AND CLOUD SERVICES EMBED, BUNDLE, OR OTHERWISE INCLUDE OTHER SOFTWARE, INCLUDING OTHER CLOUD SG SOFTWARE (COLLECTIVELY, “INCLUDED SOFTWARE”). USE OF INCLUDED SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED CLOUD SG SOFTWARE AND/OR CLOUD SERVICES. THE INCLUDED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER CLOUD SG SOFTWARE AND/OR CLOUD SERVICES OR FOR ANY OTHER PURPOSE.

USE OF CLOUD SG SOFTWARE AND CLOUD SERVICES IS SUBJECT TO THE TERMS AND CONDITIONS OF AN AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER AGREEMENT WHICH IS DISPLAYED WHEN ACCESSING, DOWNLOADING, OR INSTALLING THE SOFTWARE OR CLOUD SERVICES (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH LICENSE AGREEMENT OR CLICKWRAP END USER AGREEMENT, THE LICENSE(S) LOCATED IN THE “LICENSE” FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE SAME TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

This document is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of Cloud Software Group, Inc.

Jaspersoft, JasperReports, Visualize.js, and TIBCO are either registered trademarks or trademarks of Cloud Software Group, Inc. in the United States and/or other countries.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only. You acknowledge that all rights to these third party marks are the exclusive property of their respective owners. Please refer to Cloud SG’s Third Party Trademark Notices (<https://www.cloud.com/legal>) for more information.

This document includes fonts that are licensed under the SIL Open Font License, Version 1.1, which is available at: <https://scripts.sil.org/OFL>

Copyright (c) Paul D. Hunt, with Reserved Font Name Source Sans Pro and Source Code Pro.

Cloud SG software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. See the “readme” file for the availability of a specific version of Cloud SG software on a specific operating system platform.

THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. CLOUD SG MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S), THE PROGRAM(S), AND/OR THE SERVICES DESCRIBED IN THIS DOCUMENT AT ANY TIME WITHOUT NOTICE.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "README" FILES.

This and other products of Cloud SG may be covered by registered patents. For details, please refer to the Virtual Patent Marking document located at <https://www.tibco.com/patents>.

Copyright © 2005-2024. Cloud Software Group, Inc. All Rights Reserved.