



# **JasperServer Installation Guide**

**Version 2.0.1**

# 1 Introduction

## 1.1 About JasperServer

JasperServer builds on JasperReports as a comprehensive family of Business Intelligence (BI) products, providing robust static and interactive reporting, report server, and data analysis capabilities. For business intelligence users, JasperSoft offers JasperAnalysis, which runs seamlessly on JasperServer.

## 1.2 Conventions

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

	Convention	Meaning
<b>Paths and File Locations</b>	<js-install>	The root directory where JasperServer will be installed.
	<apache-tomcat>	The directory where Tomcat is installed. If you plan to use the instance of Tomcat that is included in the installer, Tomcat is installed under the <js-install> directory.
	<jboss>	The directory where JBoss is installed.
	<mysql>	The directory where MySQL is installed. If you plan to use the instance of MySQL that is included in the installer, MySQL is installed under the <js-install> directory.
<b>User Names and Passwords</b>	jasperadmin/ <jasperadmin>	The user name and password of the default JasperServer administrative user. Also refers to the RDBMS user created for use with JasperServer.
	sugarcrm/ <jasperadmin>	In Oracle, the user name and password of the account that owns the sample SugarCRM schema; only created if you install the sample data.
	foodmart/ <jasperadmin>	In Oracle, the user name and password of the account that owns the sample foodmart schema; only created if you install the sample data.

## 2 JasperServer Distributions

JasperServer can support many combinations of operating system, application server, and RDBMS. To simplify installation for the most common combinations, JasperSoft provides installers that are specific to each supported operating system.

If your operating system, application server, or RDBMS is not listed here, it may still be supported, but you may need to manually deploy the WAR file and configure the RDBMS. For more information, refer to Appendix A "Install Using War File Distribution Zip".

### 2.1 Third Party Server Software Supported by the Installers

The installers support these operating systems for the application server host:

Windows	Linux
Windows 2003	Red Hat Enterprise Linux
Windows XP	SUSE

The installers support these application servers:

- Tomcat

The installers support these RDBMS:

- MySQL

**Notes:**

- You should be able to use JasperServer in any compliant container. JasperSoft may offer assistance in non-certified environments but cannot guarantee success.
- This document may describe third-party software that this version of the distribution does not yet support. Subsequent distributions will support a wider range of platforms. See the Supported Platforms Guide on the Support Portal for guidance about when additional platform will be supported.

## 2.2 Included Components

The installers include these components:

- JasperServer
- JasperAnalysis (runs within JasperServer)
- Java Runtime
- MySQL (Optional)
- Tomcat Application Server (Optional)

## 3 Prerequisites for Installation

JasperServer relies on third-party products, such as application servers and RDBMS. Unless you use the ones included with JasperServer, these third party products must be installed and configured before installing JasperServer. Refer to the sections below that relate to your preferred application server and RDBMS.

### 3.1 System Requirements

The requirements in this table are deliberately loosely defined. The table contains recommendations for minimum and recommended configurations for a full installation, including MySQL and application server. The values are based on our own testing. You may find that JasperServer can run on systems with fewer resources or slower systems than stated in the **Recommended Minimum** column. At the same time, it is possible to run out of resources with the recommended configuration. The success of your deployment depends on the intended load of the system, the number of concurrent users, the data sets and whether the databases are installed on the same system as the JasperServer. The values in the table are guidelines to help you deploy.

Category	Installed Footprint	Recommended Minimum	Recommended
<b>Windows</b>		Windows Vista Windows XP Professional Windows Server 2003 Windows Server 2000 SP4	Windows Server 2003
Disk	~600MB	10 GB free	40 GB +
RAM		512 MB	1 GB +
CPU (class)		1 GHz (single Pentium)	1.5 GHz + (multi-core Pentium)
<b>MAC</b>		OSX	OSX
Disk	~600MB	10 GB free	40 GB +
RAM		512 MB	1 GB +
CPU (class)		1 GHz (single Pentium)	1.5 GHz + (multi-core

			Pentium)
<b>Linux</b>		RHEL AS/ES Novell SUSE Fedora Linux Enterprise Debian Linux	RHEL AS/ES
Disk	~600MB	10 GB free	40 GB +
RAM		512 MB	1 GB +
CPU (class)		1 GHz (single Pentium)	1.5 GHz + (multi-core Pentium)
<b>Solaris</b>			
Disk	~600 MB	10 GB free	40 GB +
RAM		512 MB	1 GB +
CPU (class)		UltraSparc II	
<b>AIX</b>			
Disk	~600 MB	10 GB free	40 GB +
RAM		512 MB	1 GB +
CPU (class)			
<b>HP-UX</b>			
Disk	~600 MB	10 GB free	40 GB +
RAM		512 MB	1 GB +
CPU (class)			

### 3.2 Included vs. Existing Software

To facilitate installation and maintenance, JasperSoft provides some of the third-party software that JasperServer supports:

- Tomcat. When you install JasperServer, you can optionally install Apache Tomcat.  
If you already have an instance of Tomcat, you can specify it during the installation of JasperServer. To obtain Tomcat separately, please visit <http://tomcat.apache.org>.
- MySQL. For each operating system, JasperSoft provides two different installers: one that includes a “bundled” MySQL and one that doesn’t. The installer that includes the “bundled” MySQL only handles a Tomcat-MySQL installer option.
- If you already have an instance of MySQL, you can specify it during the installation of JasperServer. To obtain MySQL separately, please visit <http://www.mysql.com>.

For information about requirements for the products, refer to the following sections below.

### 3.3 UTF-8 Configuration for Existing Software

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

---

## 4 Installing JasperServer

When you run the installation executable, you are prompted to specify information about the third party software JasperServer relies on. You are prompted for many of the same values, regardless of operating system, application server, or RDBMS. However, some prompts, such as those regarding the database, vary with the third party software you implement.

This section sets forth all the steps you might encounter when installing JasperServer.

**Note:** When you run the installer against existing instances of your application server and RDBMS, it connects to them to configure and deploy the application. Thus, your application server and RDBMS must be running when you install JasperServer. For startup instructions, see section 5 “Starting and Stopping JasperServer” on page 8.

### 4.1 Installation Steps

To begin, run the installer on the application server host.

**Note:** The name of the installer varies with the operating system and whether it includes MySQL.

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

```
jasperserver-<ver>-windows-installer.exe
```

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

```
./jasperserver-<ver>-linux-installer.bin
```

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

#### 4.1.1 Welcome

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

#### 4.1.2 Accept License Agreement

You are prompted to read and accept the license agreement. Read the agreement, agree to the terms by clicking **I accept the agreement**, and click **Next**.

**Command Line Note:** You must page through several screens of text to read the full agreement.

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

#### 4.1.3 Installation Directory Location

You are prompted for the directory where JasperServer will be installed (that is, the <js-install> directory). Accept the default or click **Browse** and select a different location, and click **Next**.

The default <js-install> directory varies with your operating system:

- In Windows, the default is C:\Program Files\JasperServer-<ver>
- In Linux, the default is <USER\_HOME>/JasperServer-<ver>

**Command Line Note:** Press the **Enter** key to accept the default. To choose a different directory location, enter that location at the prompt.

#### 4.1.4 Installation Mode

The installers include options for multiple configurations, such as the application server or RDBMS type you implement. You may also be prompted to indicate whether to use the MySQL and Tomcat products included in the JasperServer distribution. Choose the options that suit your needs.

The selections you make determine the other information you must supply. For example, if you plan to use an existing MySQL instance, you are prompted for the binary directory, IP address, and port number; if you choose to install the MySQL instance distributed with JasperServer, you are prompted for the user name and password to create in MySQL.

#### 4.1.5 Tomcat

If you plan to use Tomcat, you are prompted for its location; the default is <js-install>\apache-tomcat. Accept the default if you plan to use the Tomcat instance that can be installed with JasperServer, or click **Browse** to locate and select another location. Click **Next**.

#### 4.1.6 MySQL

If you are installing to an existing installation of MySQL, you must identify the database host:

<b>Binary Directory</b>	The directory where the mysql and mysqladmin binaries are located.
<b>IP or Host Name</b>	The IP address or the name of the computer hosting your existing MySQL instance.
<b>Port</b>	The port number that MySQL uses; the default is 3306. <b>Note:</b> If you install the instance of MySQL that is included in the installer, you are prompted only for the port number.

If you install the MySQL included in the JasperServer installer, you must identify the database user:

<b>User Name</b>	The user name for the MySQL root database account.
<b>Password</b>	The password for the MySQL root database account.
<b>Confirm Password</b>	Re-enter the password to confirm it.

Supply the requested details and click **Next**.

#### 4.1.7 Install Sample Data

JasperServer can be installed with sample data that can help you evaluate its features. Two data sets are included:

- The Sugar CRM data simulates three years of operations for a fictitious company that relies on the SugarCRM open source application
- The Foodmart data simulates three years of operations for a fictitious company. The data is specific to showing JasperAnalysis features.
- JasperSoft strongly recommends that you install this data, unless you are not interested in testing with the default sample data. Click **Yes** to install the sample data and click **Next**.

#### 4.1.8 Password Entry

The installer creates a user named jasperadmin in the JasperServer repository. You are prompted for a password for this account. Note that a user with the same name is also created in the RDBMS instance you specified previously. This is the user that owns the JasperServer metadata.

These instructions assume that the password to this account is <jasperadmin>.

Enter and confirm the password, then click **Next**.

#### 4.1.9 Scheduled Report Confirmation

When a scheduled report completes, JasperServer can send email notifications. This functionality requires a user account in a mail server. Enter information about the mail server JasperServer should use:

<b>Host Name</b>	The name of the computer hosting the mail server.
<b>Port</b>	The port number that the mail server uses. For SMTP, the default is typically 25.
<b>Protocol</b>	The protocol that the mail server uses. JasperServer only supports SMTP.

	<b>Note:</b> Your entry must be lower case. For example: <b>smtp</b>
<b>Email From Address</b>	The address that will appear in the <b>From</b> field on email notifications.
<b>Mail User name</b>	The name of the user in the mail server that JasperServer can use.
<b>Password</b>	The password of the mail server user.
<b>Confirm Password</b>	Re-enter the password to confirm it.

Supply the requested details and click **Next**.

**Note:** These fields are optional. They do not need to be filled out to run JasperServer.

#### 4.1.10 Install iReport

The iReport application is the leading GUI-based JasperReports creation tool. It has the capability of communicating directly with a JasperServer instance and can thus retrieve existing JasperReports from a JasperServer instance for editing, uploading or executing.

In the installer, iReport comes pre-configured with a plugin that allows it to communicate with JasperServer.

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

#### 4.1.11 Ready to Install

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

**Note:** The sample database scripts typically take a few minutes each to run.

#### 4.1.12 View Release Notes File

After the files have been installed, you are prompted to view the release notes file. If you choose to see the release notes file, it is displayed immediately.

#### 4.1.13 Register at JasperForge

If you check this box, the installer will launch a browser pointing to JasperForge.org. Registering on JasperForge gives you access to updates, community support, and documentation.

#### 4.1.14 Launch JasperServer Now

Depending on your installation configuration, you are prompted whether to start JasperServer immediately. For instance, if you installed to an existing application server, then you will need to start this application server using its own start script.

For other configurations, you must start JasperServer as described in section 5 “Starting and Stopping JasperServer” on page 12.

If you choose to launch JasperServer from the installer, the installer exits and the application server starts. It takes a few moments for the server to start up. When this is complete, the login page appears. For more information, see section 6 “5BLogging into JasperServer” on page 8.

#### 4.1.15 Installer Log File

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

Windows:

C:\Documents and Settings\username\Local Settings\Temp\bitrock\_installer\_<number>.log

Linux:

/tmp/bitrock\_installer.log or bitrock\_installer\_<number>.log

## 5 Starting and Stopping JasperServer

JasperServer runs in an application server. JasperServer also has database dependencies, so your database must be running, as well.

### 5.1 Using JasperServer Start and Stop scripts

If you used the installer to install JasperServer and you chose to use a “bundled” Tomcat and a “bundled” MySQL, the JasperServer start/stop scripts will allow you to start both applications with a single script.

To do this, do the following:

#### From Command Line:

```
cd <js-install>/bin
./allclt.bat start [stop]           (Windows)
cd <js-install>
./jasperctl.sh start [stop]       (Linux)
```

#### From Windows Start Menu:

Click **Start> All Programs> JasperServer > JasperServer Management> Start JasperServer**.

Click **Start> All Programs> JasperServer > JasperServer management> Stop JasperServer**.

### 5.2 Using Start/Stop Scripts without Bundled Installation

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

So, for instance, if you have an existing Tomcat and a “bundled” MySQL, the scripts and menus specified in the previous section would only start and stop the MySQL application.

## 6 Logging into JasperServer

This section assumes that JasperServer is running. If it isn't, start it as described in section 5, “Starting and Stopping JasperServer”, on page 12.

Log into JasperServer by entering the correct URL in your browser's address field and supplying the correct user name and password. JasperServer supports Mozilla Firefox and Internet Explorer. The URL varies with your application server:

Application Server	URL
Tomcat	http://<hostname>:8080/jasperserver

where:

- <hostname> is the name of the computer hosting JasperServer.
- <ip\_address> is the IP address of the computer hosting JasperServer (if not using hostname).
- 8080 is the default port number for the relevant application server. If you used a different port when installing your application server, specify its port number when you connect to JasperServer.

In Windows, you can also connect to JasperServer from the desktop of its host by clicking **Start> All Programs> JasperServer > JasperServer**.

If the login page appears, JasperServer has started properly. You may now login with the following username and password:

**Username:** jasperadmin  
**Password:** <password>

If you installed the sample data then an end-user named joeuser is created. This user does not have Administration privileges; it only has the role of ROLE\_USER. You may login as this sample user:

**Username:** joeuser

**Password:** joeuser

**Security Note:** Once you begin setting up your system for your own users you should remove the sample joeuser user.

## 7 Starting iReport

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

## 8 Log Files

Log files contain important information about how JasperServer is running. In addition to the JasperServer log, your application server and database server also log information about JasperServer. For information about the logs written by your application server and your database server, refer the associated documentation.

The JasperServer log file location is:

<application-server-path>/jasperserver/WEB-INF/logs/jasperserver.log

The log4j.properties file location is:

<application-server-path>/jasperserver/WEB-INF/log4j.properties

**Note:** By default, JasperServer only logs errors and warnings.

## 9 Uninstalling JasperServer

In Windows, click **Start > All Programs > JasperServer > Uninstall** to uninstall JasperServer.

In addition, in Windows, you can open the Control Panel and double-click the **Add or Remove Software** option. Locate JasperServer in the list of installed software and click **Change/Remove**. You are prompted to remove the software. Indicate **Yes** and follow the on-screen instructions.

In Linux and Solaris, the <js-install> directory includes an executable that removes JasperServer from its host. From the command line as the root user (or any user with sufficient privileges), enter:

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

You are prompted whether to remove JasperServer. Press **Y** and then press **Enter** to remove JasperServer from this computer.

---

## Appendix A Install Using War File Distribution Zip

### A.1 Introduction

In addition to the installer binaries, the JasperServer application is released as a stand-alone War File Distribution packaged as a zip file. This format is useful for customers who do not wish to use the installer or to support configuration options that are not currently available in the installers.

The jasperserver.war is designed to be dropped into an application server such as Apache Tomcat or JBoss. Scripts are included that help in the creation of the databases and the database configurations necessary to run JasperServer.

### A.2 Obtain the War File Distribution Zip

The war file distribution comes in a ZIP file format. To download the war distribution, login to the JasperForge site and go the JasperServer Downloads section. The JasperForge is at the following location:

<http://www.jasperforge.org>

### A.3 Unpack the War File Distribution Zip

Once you have downloaded the war file distribution, you will need to unpack it in order to access the contained files.

Choose a top level directory location to unpack the ZIP file to. The ZIP file will create a directory with the following naming structure:

```
jasperserver-<ver>-bin
```

Unpack to a directory location such as:

```
C:\jasperserver-<ver>-bin (Windows)
```

```
/home/<user>/jasperserver-<ver>-bin (Linux)
```

### A.4 Setup the JasperServer Database

To begin the database setup for JasperServer, your database must be installed and running.

The steps below describe setting up JasperServer with the minimal database setup. Later sections will describe the steps used for adding the JasperServer Sample Data to your new JasperServer instance.

#### A.4.1 MySQL Database Setup Steps

The MySQL client software, mysql.exe or mysql, will be used to create and populate the database.

This document will give example database setup commands that have been tested at JasperSoft. The precise commands to be used on your MySQL instance may be different.

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

##### Create and Populate the Database

Please check your database user manual for how to set up a database and how to create a database user. You will create the following (suggested) database instance, DB user and DB password.

First move to the scripts directory:

```
cd <unpacked-war-dir>/scripts/mysql
```

Enter the following commands:

```
mysql -u root -p
```

```
mysql>create database jasperserver character set utf8; (create database)
```

---

```
mysql>grant all on *.* to jasperadmin@localhost identified by 'password';
mysql>flush privileges;                                (reload privilege tables)
mysql>use jasperserver;
mysql>source jasperserverCreate-mysql.ddl              (create schema)
mysql>source jasperserverCreateDefaultSecurity-mysql.sql (populate minimal data)
mysql>quit
```

The MySQL database is now setup for use with JasperServer.

## A.4.2 PostgreSQL Database Setup Steps

The PostgreSQL client software psql will be used to create and populate the database.

### Create and Populate the Database

First move to the scripts/postgresql directory:

```
cd <unpacked-war-dir>/scripts/postgresql
```

Start psql using an admin account (such as “postgres”):

```
psql -U postgres
```

Enter the following commands:

```
create database jasperserver encoding='utf8';
\c jasperserver
\i jasperserverCreate.ddl
\i jasperserverCreateDefaultSecurity.sql
\q
```

The PostgreSQL database is now setup for use with JasperServer.

## A.5 Deploy the WAR to the Application Server

The process for deploying JasperServer into the application server involves “dropping” the WAR file into the proper directory.

First, you will need to locate your Tomcat or JBoss web applications directory:

```
<apache-tomcat>/webapps          (Tomcat)
<jboss>/server/default/deploy    (JBoss)
```

Then copy:

```
<unpacked-war-dir>/jasperserver.war
```

To the following directory:

```
<apache-tomcat>/webapps          (Tomcat)
<jboss>/server/default/deploy    (JBoss)
```

## A.6 Configure jasperserver.war for Database Connections

### A.6.1 Default Database Configuration Values

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

Setting	Default Config for MySQL	Default Config for PostgreSQL
Database Host	localhost	localhost
Database Name	jasperserver	Jasperserver
Database User	jasperadmin	postgres
Database Password	password	postgres
Database Port	3306	5432

Your instance of JasperServer should be modified to support the values in your own database environment.

## A.6.2 Dealing with the Archived WAR File

The jasperserver.war file is "archived" so you cannot directly edit files inside of it. The JasperServer database configuration file is inside the war file, so we must "un-archive" the war file first. JasperServer will be run using this "unarchived" war file.

### Un-archive the War File Using the Java "jar" Command

You can use the following steps to un-archive the war file:

#### Tomcat and JBoss

```
cd <apache-tomcat>/webapps (Tomcat)
cd <jboss>/server/default/deploy (JBoss)

mkdir jasperserver
cd jasperserver
jar xvf ../jasperserver.war (x - extract, v - verbose, f - is filename)
cd ..
rm -r jasperserver.war or del jasperserver.war (remove the original war file)
```

#### JBoss

```
mv jasperserver jasperserver.war (add ".war" to end of dir name)
copy jasperserver jasperserver.war (Windows)
```

## A.6.3 Configuration Steps for Tomcat

In order to connect to the database you will need to modify the following configuration file:

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

### MySQL Context.xml for Tomcat

Edit the items in bold and supply settings for your installation:

```
<Context path="/jasperserver"
  debug="5" reloadable="true" crossContext="true">

  <Resource name="jdbc/jasperserver" auth="Container" type="javax.sql.DataSource"
    maxActive="100" maxIdle="30" maxWait="10000"
    username="jasperadmin" password="password" driverClassName="com.mysql.jdbc.Driver"
    url="jdbc:mysql://localhost:3306/jasperserver ?useUnicode=true&characterEncoding=UTF-8"/>
```

```
<!--Configs below are for Sample Data -->
```

```
<Resource name="jdbc/sugarcrm" auth="Container" type="javax.sql.DataSource"
```

```

maxActive="100" maxIdle="30" maxWait="10000"
username="jasperadmin" password="password" driverClassName="com.mysql.jdbc.Driver"
url="jdbc:mysql://localhost:3306/sugarcrm"/>

<Resource name="jdbc/foodmart" auth="Container" type="javax.sql.DataSource"
maxActive="100" maxIdle="30" maxWait="10000"
username="jasperadmin" password="password" driverClassName="com.mysql.jdbc.Driver"
url="jdbc:mysql://localhost:3306/foodmart"/>
</Context>

```

## PostgreSQL Context.xml for Tomcat

Edit the items in bold and supply settings for your installation:

```

<Context path="/jasperserver"
debug="5" reloadable="true" crossContext="true">

<Resource name="jdbc/jasperserver" auth="Container" type="javax.sql.DataSource"
maxActive="100" maxIdle="30" maxWait="10000"
username="postgres" password="postgres" driverClassName="org.postgresql.Driver"
url="jdbc:postgresql://localhost:5432/jasperserver"
defaultAutoCommit="false"/>

<!-- Configs below are for Sample Data -->
<Resource name="jdbc/sugarcrm" auth="Container" type="javax.sql.DataSource"
maxActive="100" maxIdle="30" maxWait="10000"
username="postgres" password="postgres" driverClassName="org.postgresql.Driver"
url="jdbc:postgresql://localhost:5432/sugarcrm"/>

<Resource name="jdbc/foodmart" auth="Container" type="javax.sql.DataSource"
maxActive="100" maxIdle="30" maxWait="10000"
username="postgres" password="postgres" driverClassName="org.postgresql.Driver"
url="jdbc:postgresql://localhost:5432/foodmart"/>
</Context>

```

## A.6.4 Configuration Steps for JBoss

### Check for jboss-web.xml

The jboss-web.xml should already be in the following location:

```
<unarchived-jasperserver>/WEB-INF/jboss-web.xml
```

If you are installing from an old jasperserver.war file, it may not be there. If this is the case, you can copy the jboss-web.xml into the un-archived jasperserver war file:

Copy:

```
<unpacked-war-dir>/scripts/jboss/jboss-web.xml
```

To:

```
<unarchived-jasperserver>/WEB-INF
```

### Special log4j Step for JBoss

JBoss is normally distributed with the log4j facility enabled. JBoss initializes the log4j.jar at JBoss startup. JasperServer also includes and uses log4j. When JBoss loads the JasperServer war file, there is a JBoss logging error (log4j:ERROR "org.jboss.logging.util.OnlyOnceErrorHandler"). This error is not fatal to JasperServer, but can cause confusion when seen at JBoss startup.

To get rid of this error, we will delete the log4j.jar found in JasperServer.

Delete the following jar:

```
rm <unarchived-jasperserver>/WEB-INF/lib/log4j-<ver>.jar
```

## MySQL Datasource for JBoss

In the <unpacked-war-dir>/scripts/jboss directory, you will find a sample js-mysql-ds.xml file.

This datasource definition file should be copied to the following location:

```
<jboss>/server/default/deploy
```

You should edit this file and make sure that it has the proper settings for your database configuration.

Check that the following lines have the correct values:

```
<connection-url>jdbc:mysql://localhost:3306/jasperserver?useUnicode=true&characterEncoding=UTF-8</connection-url>
<connection-url>jdbc:mysql://localhost:3306/sugarcrm?useUnicode=true&characterEncoding=UTF-8</connection-url>
<connection-url>jdbc:mysql://localhost:3306/foodmart?useUnicode=true&characterEncoding=UTF-8</connection-url>
<user-name>jasperadmin</user-name>
<password>password</password>
```

## PostgreSQL Datasource for JBoss

In the <unpacked-war-dir>/scripts/jboss directory, you will find a sample js-oracle-ds.xml file.

This datasource definition file should be copied to the following location:

```
<jboss>/server/default/deploy
```

You should edit this file and make sure that it has the proper settings for your database configuration.

Check that the following lines have the correct values:

```
<connection-url>jdbc:postgresql://localhost:5432/jasperserver</connection-url>
<connection-url>jdbc:postgresql://localhost:5432/sugarcrm</connection-url>
<connection-url>jdbc:postgresql://localhost:5432/foodmart</connection-url>
<user-name>postgres</user-name>
<password>postgres</password>
```

### A.6.5 Check the Hibernate Properties Configuration

Make sure the hibernate.properties file has the right setting. The file is found at the following location:

```
<apache-tomcat>/webapps/jasperserver/WEB-INF/hibernate.properties
<jboss>/server/default/deploy/jasperserver/WEB-INF/hibernate.properties
```

#### MySQL

```
metadata.hibernate.dialect=org.hibernate.dialect.MySQLDialect
```

#### Oracle

```
metadata.hibernate.dialect=org.hibernate.dialect.PostgreSQLDialect
```

### A.6.6 Copy Database Driver File

Copy one of these drivers:

---

<unpacked-war-dir>/scripts/drivers/mysql-connector-java-<ver>-bin.jar	(MySQL)
<unpacked-war-dir>/scripts/drivers/postgresql-8.2-504.jdbc3.jar	(PostgreSQL)
To your application server:	
<apache-tomcat>/common/lib	(Tomcat)
<jboss>/server/default/lib	(JBoss)

## A.7 Start JasperServer

To run JasperServer you will start your application server.

Start the JasperServer application:

<apache-tomcat>/bin/startup.bat or startup.sh	(Tomcat)
<jboss>/bin/run.bat or run.sh	(JBoss)

The JasperServer log output goes to the following location:

<apache-tomcat>/webapps/jasperserver/WEB-INF/logs/jasperserver.log	(Tomcat)
<jboss>/server/default/deploy/jasperserver/WEB-INF/logs/jasperserver.log	(JBoss)

You can change the log4j logging level for the overall application or for particular classes by modifying the following property:

<apache-tomcat>/webapps/jasperserver/WEB-INF/ log4j.properties	(Tomcat)
<jboss>/server/default/deploy/jasperserver/WEB-INF/ log4j.properties	(JBoss)

The application server console log should also be checked for errors.

## A.8 Login to JasperServer

If JasperServer started up cleanly you should be able to login.

Login by going to the following URL:

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

Example:

<http://localhost:8080/jasperserver>

<http://my.server.com:8080/jasperserver>

The login page should appear (after some time to compile the necessary .jsp files).

Use the following ID to login to the system:

Username: jasperadmin	(Administrative user)
Password: password	
Username: joeuser	(Normal User)
Password: joeuser	

If you logged in successfully, you will be at the home page.

Please see the *JasperServer User Guide* to begin adding reports and other objects to JasperServer.

## A.9 Setup Sample Data

The War file distribution ZIP file comes with a set of scripts and a utility that allow for the setup of a full set of sample data. This includes sample reports, analysis views, and data sources.

To create the sample data, you should have already run the steps from section A.4, “Setup the JasperServer Database”. Then you can run the additional steps below.

## A.9.1 Create and Populate Sample Databases

These next commands will be entered at the Windows or Linux command line.

### MySQL

```
cd <unpacked-war-dir>/scripts/mysql
mysql -u root -p                               (login to mysql client)
mysql>create database sugarcrm;
mysql>create database foodmart;
mysql>use sugarcrm;
mysql>source sugarcrm-mysql.sql;
mysql>use foodmart;
mysql>source foodmart-mysql.sql;
mysql>exit
```

### PostgreSQL

```
cd <unpacked-war-dir>/scripts/postgresql
psql -U postgres                               (login to PostgreSQL client)
create database sugarcrm;
create database foodmart;
\c sugarcrm
\i sugarcrm-postgresql.sql
\c foodmart
\i foodmart-postgresql.sql
\q
```

## A.9.2 Update context.xml for Sample Databases

You should check your context.xml (Tomcat) or data source definition (JBoss) to make sure that you have the correct settings for the sample databases you just created above. Use settings similar to what you set in section A.6.3, "Configuration Steps for Tomcat", on page 13, or section A.6.4, "Configuration Steps for JBoss", on page 14.

Make sure that you have the correct settings for the sugarcrm and foodmart databases. Remove comment lines from the configuration files if necessary.

**Note:** In order for database configuration changes to take effect, you will need to restart JasperServer.

## A.9.3 Configure the Import Utility

The JasperServer import utility is used to load sample metadata into the JasperServer repository. The database that will be updated is the jasperserver database.

The import utility reads a "repository catalog" file that is a directory of XML and binary files.

The batch or shell script that runs the import operation is found in the scripts directory. You will first need to make sure that the import-export utility is properly configured for execution:

**Note:** To configure Import-Export, please see Appendix D, "Configuring the Import Export Utility".

The main file that needs configuring for your database settings is:

```
<unpacked-war-dir>/scripts/config/js.jdbc.properties
```

### MySQL

Please refer to section D.3.1, "Sample Settings for MySQL", on page 27.

## PostgreSQL

Please refer to section D.3.2, "Sample Settings for PostgreSQL", on page 28.

### A.9.4 Run the Import Utility

To run the import utility, enter the following commands:

#### MySQL and PostgreSQL

```
cd <unpacked-war-dir>/scripts
js-import.bat --input-dir js-catalog           (Windows)
js-import.sh --input-dir js-catalog           (Linux)
```

**Note:** The import command depends on the JAVA\_HOME variable being set in your environment.

**Note:** there are two dashes (--) in front of the command option.

## A.10 Report Scheduling Configuration

The scheduled reporting feature of JasperServer uses the Quartz scheduler (<http://www.opensymphony.com/quartz>).

The configuration file for scheduled reporting is the following:

```
jasperserver/WEB-INF/js.mail.properties
```

Please see section 4.1.9, "Scheduled Report Confirmation", on page 6, for information on scheduled reporting configuration options.

## A.11 Restart JasperServer

Since you have made database configuration changes to your JasperServer instance, you should restart your application server.

## A.12 Update XMLA Connection Definitions (Optional)

JasperServer is able to make XMLA connections over the Web Services interface. These http-based connections use a JasperServer user account for authentication. You may have different usernames and passwords than the defaults that get loaded from the sample data load in section A.10, "Setup Sample Data".

If you would like to run the sample reports and sample view that uses this XMLA interface, you should update your XMLA connection definition. For instance, an analysis view that uses this connection is:

```
/analysis/views/SugarCRM_xmla_sample
```

The XMLA connection definitions to update with your local user names and passwords are the following:

```
/analysis/connections/FoodmartXmlaConnection
/analysis/connections/SugarCRMXmlaConnection
```

The updates are made by editing repository objects in your running instance of JasperServer.

To make these updates, take these steps:

- Login to JasperServer as an Administrative user (such as jasperadmin).
- Navigate to the Repository Management page via the **Manage > Repository** menu item.
- Click the **analysis** folder, then the **connections** folder. Click the **Edit** link (right side of page) to edit the FoodmartXmlaConnection definition.
- Edit the information on this page for:

- URI (hostname and port)
- Login Username
- Login Password
- Click **Next**, then **Save**.
- Make the same updates for /analysis/connections/SugarCRMxmlaConnection

## **A.13 Troubleshooting Your JasperServer Configuration**

### **Startup Problems**

When trying to run a newly setup JasperServer instance, the most typical problems that users encounter are problems with database configuration.

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

For more information on resolving these types of errors, please **see section E.3 of the Troubleshooting Appendix**.

### **Error Running Report**

If you have trouble running reports in your new JasperServer Instance, please **see section E.4 of the Troubleshooting Appendix**.

## Appendix B Upgrade to 2.0.1 Using War File Distribution

The upgrade procedure in this section is known as the "Fast and Simple" upgrade procedure. It is assumed that you are upgrading from JasperServer 2.0.0.

If you are using JasperServer 1.2.1 or earlier there are additional DB scripts to be run. These are described at the end of this section.

### B.1 Fast and Simple Upgrade

In this upgrade procedure we will first upgrade Tomcat to JasperServer 2.0.1. Next we will upgrade the jasperserver database "in place" by running SQL based upgrade scripts.

#### B.1.1 Backup War File and jasperserver Database

First you will need to backup your JasperServer war file and your jasperserver database so that they can be restored in case there is a problem with the upgrade. This work will be done from a command shell under Windows or Linux.

Backup the jasperserver directory in Tomcat to a backup directory:

```
cd <apache-tomcat>
mkdir js-2.0-war-backup
copy <apache-tomcat>/webapps/ji-pro to <apache-tomcat>/js-2.0-war-backup
delete directory <apache-tomcat>/webapps/jasperserver
```

#### B.1.2 Backup the jasperserver Database

##### MySQL

```
cd <orig-2.0-install-dir> (That is, the location of your original 2.0 install.)
                        (or create a backup directory on disk.)
```

##### Windows

```
mysqldump --user=jasperadmin --password=<your-password> jasperserver > js-db-2.0-dump.sql
```

##### Linux

```
mysqldump --user=jasperadmin --password=< your-password > --host=127.0.0.1 jasperserver > js-db-2.0-dump.sql
```

You can use the jasperadmin user or the root user to carry out this operation. Please see section B.2.1 for more information on running the mysqldump command.

##### PostgreSQL

```
pg_dump jasperserver > js-db-dump.sql
```

Please refer to your PostgreSQL documentation for more details.

#### B.1.3 Unpack JasperServer 2.0.1 War File Distribution

It is assumed that you have already obtained the jasperserver-2.1-war.zip distribution from the JasperSoft Tech Support Portal.

Unpack the distribution to a directory on disk. This will create the location:

```
<unpacked-war-dir-2.0.1>
```

#### B.1.4 Upgrade the JasperServer War

You have already backed up and deleted your 2.0 war file. Now you can deploy the 2.0.1 war file into Tomcat:

---

```
copy file <unpacked-war-dir-2.0.1>/jasperserver.war to <apache-tomcat>/webapps
```

Under Tomcat, you will want to "un-archive" your jasperserver.war file in order to edit the database configuration files inside it. To handle the war file under Tomcat:

Please see section A.5, "Deploy the WAR to the Application Server", and section A.6.2, "Dealing with the Archived WAR File", for general information and troubleshooting when deploying a war to Tomcat.

In order to preserve your application server configurations for the database and other local settings, you should copy over the following configuration files from your 2.0 war to your 2.0.1 war.

```
Copy from <apache-tomcat>/js-2.0-war-backup>/jasperserver to <apache-tomcat>/webapps/jasperserver:
```

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

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

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

Also, copy or integrate any other custom modifications you have made to JasperServer 2.0.

**Note:** Please see section A.5, "Deploy the WAR to the Application Server", and section A.6.2, "Dealing with the Archived WAR File", for general information and troubleshooting when deploying a war to Tomcat.

## B.1.5 Upgrade the jasperserver Database

The jasperserver database will be upgraded in place by running SQL based scripts:

```
cd <unpacked-war-dir-2.0.1>/scripts/upgrade
```

### MySQL

Login to your MySQL client:

```
mysql -u root -p
```

Run the upgrade scripts:

```
mysql>use jasperserver;
```

```
mysql>source upgrade-mysql-2.0.0-2.0.1.sql;
```

### PostgreSQL

```
psql -U postgres (login to the PostgreSQL client)
```

```
\c jasperserver
```

```
\i upgrade-postgresql-1.2.1-2.0.0.sql;
```

## B.1.6 Start JasperServer 2.0.1

Before you actually start Tomcat with the new 2.0.1 war file, it is a good idea to clear out the Tomcat work directory. The work directory is where compiled JSPs and other temporary files are stored.

Clear Tomcat work directory:

```
cd <apache-tomcat>/work
```

Remove all files below "work" directory

Now you may start Tomcat. The MySQL database should already be running.

## B.1.7 Login to JasperServer 2.0.1

If Tomcat and JasperServer 2.0.1 were started correctly, you should be able to login.

Login using the following URL and identity:

`http://localhost:8080/jasperserver`

username: `jasperadmin`

password: `<your-password>`

## B.2 Additional Notes on JasperServer Upgrade

### B.2.1 Using mysqldump for Database Backup

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

The following command is an example of using `mysqldump` to back up the JasperServer database on a Windows system:

```
mysqldump --user=jasperadmin --password=<jasperadmin> jasperserver > js-db-2.0-dump.sql
```

The following command will perform the same task on a Linux or Solaris system:

```
mysqldump --user=jasperadmin --password=<jasperadmin> --host=127.0.0.1 jasperserver > js-db-2.0-dump.sql
```

Note:

- All the options in the commands above begin with two dashes.
- The JasperServer installation will erase your existing JasperServer database, so take steps to ensure that your `mysql dump` (or other means of backup) has backed up all the information in the database.
- If the repository contains file resources larger than one megabyte, you may encounter the following message when restoring the database: “ERROR 1153 (08S01): Got a packet bigger than 'max\_allowed\_packet' bytes”. This error requires adjustment of your MySQL server configuration. See the following URL for more information: <http://dev.mysql.com/doc/refman/5.0/en/packet-too-large.html>

### B.2.2 Additional Configuration Files

If you made modifications or customizations to your JasperServer 2.0 application, these configurations are typically found in the `WEB-INF/applicationContext-*.xml` set of files. Configuration modifications such as client specific security classes or LDAP server configurations, need to be hand copied from the older 2.0 environment to the new 2.0.1 environment.

### B.2.3 Clear the Tomcat/Work Directory Files

This directory path is where `.jsp` files are compiled. If you are doing an upgrade to an existing Tomcat installation (if the Tomcat used to hold an older version of JasperServer), you should make sure that the work directory is cleared.

Change directory to:

```
cd <tomcat-install>/work
```

Delete all files in this directory.

## B.3 Upgrade Notes for JasperServer 1.1 Users

JasperServer 1.1 users should run the additional scripts specified in this section before running the scripts specified in sections above:

MySQL

- `upgrade-mysql-1.1.0-1.2.0.sql`
- `upgrade-mysql-1.2.0-1.2.1.sql`
- `upgrade-mysql-1.2.1-2.0.0.sql`
- `upgrade-mysql-1.2.0-2.0.0.sql`
- `upgrade-mysql-sample-text-1.2.0-2.0.0.sql`;

Note: The last script is somewhat cosmetic in that it changes the /olap directory files to /analysis in the repository.

## **B.4 Upgrade Notes for JasperServer 1.2 Users**

JasperServer 1.2 users should run the additional scripts specified in this section before running the scripts specified in sections above:

### MySQL

- upgrade-mysql-1.2.1-2.0.0.sql
- upgrade-mysql-1.2.0-2.0.0.sql
- upgrade-mysql-sample-text-1.2.0-2.0.0.sql;

### PostgreSQL

- upgrade-postgresql-1.2.1-2.0.0.sql
- upgrade-postgresql-sample-text-1.2.1-2.0.0.sql

Note: The last script is somewhat cosmetic in that it changes the /olap directory files to /analysis in the repository.

---

## Appendix C Adding Password Encryption to JasperServer

### C.1 Introduction

JasperServer has the capability of running with encrypted passwords in the database. When this feature is enabled, passwords in the database will be stored as cipher text. Customers have the ability to choose the algorithm that JasperServer will use, as well as specify the salt key used to initiate the encryption algorithm.

By default password encryption is turned off.

This section will describe the procedure to enable password encryption. For more information on JasperServer encryption options please see the *JasperServer User Guide*.

#### C.1.1 General Notes

By default, JasperServer is released with encryption turned off. The steps described in this section are meant to be applied after you have installed JasperServer.

#### C.1.2 Backup your jasperserver database

As a precaution, you will need to backup your jasperserver database in case there is any problem with the encryption enabling process.

##### MySQL

Under Windows or Linux enter the following commands:

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

Windows

```
mysqldump --user=jasperadmin --password=<your-password> jasperserver > js-db-dump.sql
```

Linux

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

You can use the jasperadmin user or the root user to carry out this operation. Please see section B.2.1 for more information on running the mysqldump command.

##### PostgreSQL

```
pg_dump jasperserver > js-db-dump.sql
```

Please refer to your PostgreSQL documentation for more details.

#### C.1.3 Stop Application Server

You can now stop your application server since it will be re-started after configuration settings are changed in the following sections. You should leave MySQL or PostgreSQL running.

#### C.1.4 Run Repository Export Utility

The Repository Export utility will write out all of the JasperServer Repository objects to a set of XML format files. The output of the export operation is known as an export catalog file.

To create the export catalog, enter these commands:

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

```
js-export.bat --everything --output-dir js-backup-catalog (windows)
```

```
js-export.sh --everything --output-dir js-backup-catalog (linux)
```

**Note:** There are two dashes (--) in front of the command options.

## C.1.5 Specify Encryption Setting in JasperServer War Configuration Files

JasperServer uses Spring configuration and acegi security to enable the encryption configuration. There is more specific information in the *JasperServer User Guide* about the security algorithms and settings. These options can allow you to have a strong encryption setup. In this section, we will mostly focus on the minimal configuration necessary for enabling encryption.

The file to be edited is the following:

```
<apache-tomcat>/jasperserver/WEB-INF/ApplicationContext-security.xml
```

### Uncomment Reference to passwordEncoder Bean

In the definition of the daoAuthenticationProvider bean, there is a commented-out reference to the passwordEncoder bean. Look for the section of the XML file that starts with:

```
<bean id="daoAuthenticationProvider"
```

In this bean definition, uncomment the reference to passwordEncoder. This will cause the passwordEncoder logic to be used. After removing the commenting characters the line should look like the following:

```
<property name="passwordEncoder"><ref local="passwordEncoder"/></property>
```

### Enable Encryption in the “passwordEncoder” Bean

The property allowEncoding should be changed from false to true so that it looks like the following:

```
<property name="allowEncoding"><value>true</value></property>
```

### SecretKey Definition Properties

The next two properties will work with their default values. However, for better security, we recommend that they be changed.

If the default DESede algorithm is used, the secretKey (salt key) is expected to be 24 characters. The key can be in plain text to make it easier to enter. To hold the key as plain text, you will also set the keyInPlainText property to true.

Here is an example:

```
<property name="keyInPlainText"><value>true</value></property>
<property name="secretKey"><value> jaspersoftInSanFrancisco</value></property>
```

**Note:** The text “jaspersoftInSanFrancisco” is 24 characters long.

### Remaining Properties

The last two properties can be left unchanged. They are set to DESede by default. The default settings are the following:

```
<property name="secretKeyAlgorithm"><value>DESede</value></property>
<property name="cipherTransformation"><value>DESede/CBC/PKCS5Padding</value></property>
```

Note that, as described in the *JasperServer User Guide*, the secretKey, secretKeyAlgorithm, and cipherTransformation property settings must be consistent with each other. For instance, different algorithms expect different key lengths.

### Encryption Now Enabled

Once the changes described above are made, encryption is enabled for the JasperServer application upon the next restart.

## C.1.6 Specify Encryption Setting in Import Utility Configuration Files

Before starting JasperServer, we will need to convert the plain text passwords that are currently stored in the repository export catalog that you created in step E.1.4 (that is, in `scripts/js-backup-catalog`). These plain-text passwords need to be converted to cipher text and updated to the database in order to successfully login after server restart.

To do this, you will need to add the same encryption settings to the configuration file that are used by the import-export utility.

The configuration file is located here:

```
<js-install>/scripts/config/applicationContext-security.xml
```

In this file, which contains just the `passwordEncoder` bean definition, you will set the same configuration options that you set in the previous section.

## C.1.7 Drop and Recreate the jasperserver database

Next you will drop your existing jasperserver database and recreate an empty jasperserver database.

### MySQL

Change directory to:

```
cd <js-install>/scripts/mysql
```

Login to your MySQL client:

```
mysql -u root -p
```

Drop the jasperserver database, create a new one and load the jasperserver schema:

```
mysql>drop database jasperserver;
mysql>create database jasperserver character set utf8;
mysql>use jasperserver;
mysql>source jasperserverCreate-mysql.ddl;
```

### PostgreSQL

```
cd <unpacked-war-dir>/scripts/posgresql
psql -U postgres
drop database jasperserver;
create database jasperserver encoding='utf8';
\c jasperserver
\i jasperserverCreate.ddl
\i jasperserverCreateDefaultSecurity.sql
\q
```

## C.1.8 Run Repository Import Utility

The import utility will reload all of your repository data. As the data is being saved to the repository, the password fields that were plain text will be encrypted using the encryption settings you made in the sections above.

To import your backup catalog to the repository, change directory to:

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

Enter these commands:

```
js-import.bat --input-dir js-backup-catalog (windows)
```

```
js-import.sh --input-dir js-backup-catalog (linux)
```

**Note:** There are two dashes (--) in front of the command options.

### C.1.9 Start Application Server

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

### C.1.10 Login to JasperServer

You can now login to JasperServer.

You will enter your username and password in the same manner as you did before encryption was turned on.

You can check the contents of the JIUser table in the jasperserver database and examine the password column to see that the password is no longer stored in plain text.

## Appendix D Configuring the Import/Export Utility

### D.1 Introduction

The import and export utilities let you extract resources from or add resources to a JasperServer repository. The import utility is typically used at installation time in order to load the JasperServer sample data into the repository.

You may refer to the *JasperServer User Guide* for more information on command options for the import and export utility.

### D.2 Import/Export Configuration Files

In the scripts directory in your installation directory, you will find the following files that make up the main parts of the import-export utility. These are the files you will use or modify if you need to make configuration changes.

scripts/js-import.bat	Import batch script for Windows.
scripts/js-import.sh	Import shell script for Linux
scripts/config/js.jdbc.properties	Database and hibernate dialect settings file
scripts/config/applicationContext-*.xml	Spring configuration files
scripts/lib	All of the JasperServer jar files and JDBC drivers
New: build.xml	Ant script to run import-export

### D.3 Change Your Configuration Settings

When you install JasperServer from the installer binary, the import-export utility is automatically configured. However, if you are doing a manual installation from the War File Distribution you will need to modify the import-export configuration files for your database settings.

The file to be modified is the following:

```
<unpacked-war-dir>/scripts/config/js.jdbc.properties
```

#### D.3.1 Sample Settings for MySQL

Modify the items in **bold** to match your own environment setup:

```
metadata.hibernate.dialect=org.hibernate.dialect.MySQLDialect
metadata.jdbc.driverClassName=com.mysql.jdbc.Driver
```

```

metadata.jdbc.url=jdbc:mysql://localhost:3306/jasperserver?useUnicode=true&characterEncoding=UTF-8
metadata.jdbc.username=jasperadmin
metadata.jdbc.password=password

```

### D.3.2 Sample Settings for PostgreSQL

Modify the items in **bold** to match your own environment setup:

```

metadata.hibernate.dialect=org.hibernate.dialect.PostgreSQLDialect
metadata.jdbc.driverClassName=org.postgresql.Driver
metadata.jdbc.url=jdbc:postgresql://localhost:5432/jasperserver?defaultAutoCommit=false
metadata.jdbc.username=postgres
metadata.jdbc.password =postgres

```

## D.4 Extra Sample Settings Files

For database settings file samples look for the following directories in the scripts directory:

```
<unpacked-war-dir>/scripts/<your-db-flavor>
```

For instance, for PostgreSQL there is a sample import-export jdbc configuration file:

```
<unpacked-war-dir>/scripts/postgresql/js.jdbc-postgresql.properties
```

To be used, this file would be renamed to js.jdbc.properties and copied to the scripts/config directory.

## D.5 Database Driver Location

In order for the import-export utility to run, it will need the proper JDBC driver. This will allow a connection to be made to the database to support reading from or writing to the JasperServer repository database.

If JDBC driver jar files are put in the following directory, they will automatically be included on the classpath when the import or export command is run using the batch/shell script files:

```
<unpacked-war-dir>/scripts/lib
```

## D.6 Running Import or Export

To see that the import-export utility is properly configured, you can run the batch/shell scripts using the `--help` option which will display the command options:

```

js-import.bat --help           (Windows)
js-export.bat --help           (Windows)
js-import.sh --help           (Linux)
js-export.sh --help           (Linux)

```

**Note:** There are two dashes (`--`) in front of the command option.

### Running the new ant import-export script: build.xml

The ant binaries are included with the JasperServer 2.0.1. You will need to make sure your `ANT_HOME` variable is set up to point to the ant binary.

```
cd <unpacked-war-dir>/scripts
```

Sample Export:

```
ant -f build.xml export -Dexport-arg1="--output-dir=my-catalog --uris=/reports --roles=ROLE_USER"
```

Sample Import:

```
ant -f build.xml import -Dimport-arg1="--input-dir=my-catalog"
```

Run Help:

```
ant -f build.xml export -Dexport-arg1="--help"
```

```
ant -f build.xml import -Dimport-arg1="--help"
```

## D.7 Older Versions of Import-Export

The import-export utility has undergone modifications that make the configuration slightly different than earlier versions. The main differences are in configuration file naming. The basic functionality is the same, except that the 2.0 version handles the new features and objects that can be created in 2.0 (such as Report Unit Options).

Here are some of the differences:

Old	New	Comment
ji-import.bat/sh	Js-import.bat/sh	change to file name
ji-export.bat/sh	js-export.bat/sh	change to file name
scripts/ji-export-util	Scripts/config	change to directory name
ji-export-util/jdbc.properties	Config/js.jdbc.properties	change to file name
ji-export-util/applicationContext-*	Config/applicationContext-*	new context files added
N/A	build.xml	New ant import-export script

## Appendix E Troubleshooting

### E.1 On Windows Install, I Get an "Input Line Too Long" Error

**Note: This bug is fixed as of JasperServer installer versions 2.0.1 and higher.**

Under older versions of Windows, the OS may run out of environment variable space (for instance, environment variables supporting the java classpath). JasperServer does not support these older versions of Windows. If the installer is attempting to load sample data, it will use the JasperServer import-export utility which is a java utility that needs the full set of JasperServer jars on the classpath.

The error is similar to the following:



If this error occurs, you should validate that the version of Windows you are installing on is supported by JasperServer.

### E.2 Installer Freezes

If you run the JasperServer installer on any platform and the installer “freezes” or “hangs”, it is helpful to look at the log file created by the installer. This log file outputs status regarding the installer operations. If your installer has had an explicit error, there may be a specific error message in the log. At a minimum, the log file should help narrow where the error has occurred even if there is not a specific error message.

You can find the installer log in the following locations:

Windows:

C:\Documents and Settings\username\Local Settings\Temp\bitrock\_installer\_<number>.log

Linux:

/tmp/bitrock\_installer.log or bitrock\_installer\_<number>.log

If you have tried multiple installs, make sure you view the most recent install log file.

### E.3 Database Connectivity Errors

The most common problems encountered with a new JasperServer instance are database configuration problems. This section contains information that may help resolve such issues.

#### E.3.1 Login to Database to Test Connection

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

You can connect to your database using the database configuration settings that are found in JasperServer. This will validate the DB hostname, port, username, and password that are being used.

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

## MySQL

Start MySQL and try to log into MySQL directly using the jasperadmin user.

Log into MySQL from the command line. For example:

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

You are prompted for a password for the user you specified on the command line. Enter the <jasperadmin> password to login.

### E.3.2 Configuration File Locations

#### Tomcat File Locations

When using Tomcat, JasperServer configuration properties are found in the following files:

- <apache-tomcat>/webapps/jasperserver/META-INF/context.xml
- <apache-tomcat>/webapps/jasperserver/WEB-INF/hibernate.properties
- <apache-tomcat>/apache-tomcat/webapps/jasperserver-pro/WEB-INF/web.xml

#### JBoss File Locations

When using JBoss, JasperServer configuration properties are found in the following files:

- <jboss>/server/default/deploy/js-mysql-ds.xml *or*
- <jboss>/server/default/deploy/js-oracle-ds.xml
- <jboss>/server/default/deploy/jasperserver.war/WEB-INF/hibernate.properties
- <jboss>/server/default/deploy/ jasperserver.war/WEB-INF/web.xml
- <jboss>/server/default/deploy/ jasperserver.war/WEB-INF/jboss-web.xml

### E.3.3 Special Database Configuration if using Tomcat

If you installed JasperServer using the War File Distribution file and the manual installation procedures, Tomcat can have a special database configuration.

The special case occurs when you have deployed the jasperserver.war file into the Tomcat webapps directory. Valid JasperServer war deployments can be based on a single file (jasperserver.war) or an “unpacked” war file directory (jasperserver directory).

If you use a single war file for deployment under Tomcat, Tomcat will do the following steps:

- Unpack the jasperserver.war file into a new directory named jasperserver.
- Take the jasperserver-pro/META-INF/context.xml file and copy it to a new file:
  - <tomcat>/conf/Catalina/Localhost/jasperserver.xml

This database configuration in <tomcat>/conf tree will override the context.xml found in your jasperserver directory.

If you are having database trouble in this scenario, it is recommended that you keep things simple by:

- Deleting your <tomcat>/webapps/asperserver.war file
  - This will cause the jasperserver directory to be used
- Deleting your <tomcat>/Catalina/Localhost/jasperserver.xml
  - This will cause the META-INF/context.xml from your jasperserver directory.

## E.4 Error Running a Report

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

### Browse/Edit in JasperServer Repository

One common problem with individual reports is the data sources defined in JasperServer. You can troubleshoot these problems in the repository:

- Log into JasperServer as a user with administrative permissions and locate the report unit that returns errors.
- Click **edit** to identify the data source the report uses. The data source name is found on the fourth page.
- Locate this data source in the repository and click **edit**.
- Review the information specified for this data source. Often, the problem is a password that has changed, so verify the database user password.
- Click **Save**.
- Test your report. If it still returns errors, edit the data source again and try checking other values, such as the port used by the database.

### Validate the Data source Connection

If you chose to load the JasperServer sample data, you will have a number of reports and data sources that should be pre-configured for your database settings.

You can validate the data sources that are used by these sample reports by editing the data source. To do this:

- Log into JasperServer as a user with administrative permissions and locate the data sources.
- Data sources are located in the /datasources repository path.
- Click **edit** to view the details of the data source.
- On the Data Source edit page, click the **Test Connection** button to validate the database connectivity.

## E.5 Database Error after Changing MySQL's Port Number

MySQL's default port is 3306. If you entered a different port when you installed MySQL, the JasperServer installer configures them to communicate properly. If the MySQL port number has changed, or if you encounter a problem, check the configuration files to verify your port number. If it is incorrect, change it to the correct port number, save the file, and restart the application server. See section E.3.2 "Configuration File Locations" above for more information.

## E.6 Case Sensitivity for Table and Column Names

A database is case-sensitive with respect to table names if it considers "customer" and "Customer" to be two different tables. If JasperServer is using a case-sensitive database, it's important that the table names specified in query strings in the JRXML files match the actual table names found in the database. This type of problem may occur if you are transferring data from one database to another, which may result in the case of table names changing.

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

Under Linux MySQL, table and column names are case-sensitive. Linux MySQL can be configured to be non-case-sensitive by setting the configuration parameter `lower_case_table_names` to 1 in the `my.ini` or `my.cnf` file. In the MySQL manual see the section "Identifier Case Sensitivity" for more information.

Table and column names in Oracle are case-sensitive.

## E.7 Error when Running with Java 1.6

JasperServer is not certified to run with Java 1.6.

There is an error accessing XMLA connections when running under Java 1.6. The errors are similar to the following:

---

org.w3c.dom.DOMException: NAMESPACE\_ERR: An attempt is made to create or change an object in a way which is incorrect with regard to namespaces.

And

com.sun.xml.internal.messaging.saaj.SOAPEXceptionImpl: Unable to create envelope from given source:

If you encounter errors like this you should be sure you are using Java 1.5.

## E.8 Java Out of Memory Error - Runtime Java Options

If you encounter a Java out of memory error, it is suggested that you increase your Java heap size setting.

This java option is set within the application server, so you will need to set this and then restart your application server.

For Tomcat:

```
<apache-tomcat>/bin/setclasspath.bat
```

```
<apache-tomcat>/bin/setclasspath.sh
```

For JBoss:

```
<jboss>/bin/run.bat
```

```
<jboss>/bin/run.sh
```

Add to your JAVA\_OPTS setting:

```
-Xms512m -Xmx1024m
```

Another memory resource related error is running out of PermGen space. If you encounter an error relating to PermGen space you can make the following updates to your JAVA\_OPTS:

```
-XX:PermSize=128m -XX:MaxPermSize=512m
```

## E.9 Error Running a Scheduled Report

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

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

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

the problem may be the default size of the MySQL "blob" datatype. You can increase the size of this datatype by updating you're my.ini or my.cnf MySQL configuration file.

Add the following setting:

```
max_allowed_packet=32M
```

## E.10 Upgrade: Error during JasperServer 1.2 Export

Upgrading from 1.2 to 2.0 sometimes requires doing an export operation on your 1.2 database. If you get a null pointer exception like the following:

```
java.lang.NullPointerException
```

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

it may be due to an incorrect character in the following file:

```
scripts/ji-export-util/jdbc.properties.
```

The URL in this file should be checked. It should look like the following:

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

Note the ampersand "&" character. It is incorrect if it looks like the following: "&amp;"

The "&amp;" is only correct in HTML or XML context. It is incorrect in a properties file.

The error described in this section is known to happen if the user has i18n characters in their repository objects.