# JasperServer Open Source Installation Guide

# Version 2.1

**http://www.jaspersoft.com/**

# Table of Contents

# 1 Introduction

## 1.1 About JasperServer

JasperServer builds on JasperReports as a comprehensive family of Business Intelligence products, providing robust static and interactive reporting, report scheduling, and data analysis capabilities within an easy to use server environment.

## 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 |
|---|---|---|
| **Paths and File Locations** | <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. |
| **Additional Locations** | <unpacked-war-dir> | The directory where user unpacks the WAR file distribution ZIP. |
| **User Names and Passwords** | jasperadmin/<br><jasperadmin> | The user name and password of the default JasperServer administrative user. Also refers to the RDBMS user created for use with JasperServer. |

## 1.3 JasperServer Distributions

There are two main distribution packages for JasperServer. The first distribution set is the installers for the Windows and Linux platforms. The second distribution is the War File binary package.

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

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

| Distribution Package | Notes |
|---|---|
| Installers | Runs on Windows and Linux |
| War File Binary Distribution Zip | Used for manual installation |

## 1.4 Installer Distribution Support

### 1.4.1 Operating Systems Supported

The installers support the following operating systems:

| Windows | Linux |
|---|---|
| Windows 2003 | Red Hat Enterprise Linux |
| Windows XP | SUSE |

### 1.4.2 Components Included in the Installer

The installer is designed so that it is easy to get JasperServer up and running quickly. The installer comes with bundled versions of an application server, a database, and a Java runtime. This bundled software can optionally be used if you do not already have these components on your target computer. If, on the other hand, you have existing components, the installer can install to these components.

Here are the components included with the installer:

| Component | Notes |
|---|---|
| JasperServer Application | War file and support scripts |
| Java Runtime | Runs web application container (optional) |
| MySQL Database | Database Server (optional) |
| Apache Tomcat | Web application container (optional) |
| JasperServer Documentation | Found in the docs directory |

### 1.4.3 Applications Supported by the Installer

When you start the JasperServer installer, you are prompted to choose the components to use. For instance, if you already have Tomcat on your computer you can choose an "existing" Tomcat. If you would like the installer to install Tomcat for you, you can choose bundled Tomcat.

When you run the installer, you can choose between the following configurations:

| Application Server | Database |
|---|---|
| Apache Tomcat (existing or bundled) | MySQL (existing or bundled) |

### 1.4.4 Application Versions Supported

For information on the specific versions of 3rd party applications that are supported by the installer refer to the release notes for the distribution you are using. The release notes are found in the root of the installation directory.

## 1.5 War File Binary Distribution Support

The WAR file distribution is the distribution you would use to do a manual installation of the JasperServer application.

The WAR file supports many more applications than is supported by the installers.

Contents of the WAR file binary distribution:

| Content Item | Notes |
|---|---|
| JasperServer WAR file archive | This contains all of the JasperServer class files and dependent jars. |
| JasperServer Database Scripts | SQL scripts for each supported database. |
| JasperServer Standard Sample Data | Sample data that highlights JasperServer features |

| | |
|---|---|
| JasperServer Extra Samples | Web Service example applications and other sample files |
| JasperServer Documentation | Guides for end users, administrators, and developers |

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

## 1.6 Release Notes and Application Support

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

Not all applications are immediately supported when a new JasperServer version is released. For instance, some applications required additional testing beyond what is completed for the initial General Availability (GA) release.

To find out exactly what applications are supported with a particular distribution, refer to the release notes found in that distribution.

## 1.7 Prerequisites for Installation

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

### 1.7.1 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 an 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 below are guidelines to help with your deployment:

| Category | Installed Footprint | Recommended Minimum | Recommended |
|---|---|---|---|
| **Windows** | | | |
| Disk | ~600MB | 10 GB free | 40 GB + |
| RAM | | 512 MB | 1 GB + |
| CPU (class) | | 1 GHz (single Pentium) | 1.5 GHz + (multi-core Pentium) |
| **MAC** | | 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) |
| **Linux** | | | |
| Disk | ~600MB | 10 GB free | 40 GB + |

| | | | |
|---|---|---|---|
| RAM | | 512 MB | 1 GB + |
| CPU (class) | | 1 GHz (single Pentium) | 1.5 GHz + (multi-core Pentium) |
| **Solaris** | | | |
| Disk | ~600 MB | 10 GB free | 40 GB + |
| RAM | | 512 MB | 1 GB + |
| CPU (class) | | UltraSparc II | |
| **AIX** | | | |
| Disk | ~600 MB | 10 GB free | 40 GB + |
| RAM | | 512 MB | 1 GB + |
| CPU (class) | | | |
| **HP-UX** | | | |
| Disk | ~600 MB | 10 GB free | 40 GB + |
| RAM | | 512 MB | 1 GB + |
| CPU (class) | | | |

## 1.8 Support for Internationalization

JasperServer supports the full Unicode character set using UTF-8 encoding. JasperServer also depends on the underlying database and application server to complete 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 Administrator Guide* for instructions on how to configure software to support UTF-8.

# 2 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. However, some prompts may vary depending on the exact installation options you choose.

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

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

## 2.1 Installation Steps

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

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**.

### 2.1.1 Welcome

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

### 2.1.2 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.

### 2.1.3 Installation Directory Location

You are prompted for the directory where JasperServer is 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.

### 2.1.4 Installation Mode

The installers include options for multiple configurations of the application server and database. You are prompted to indicate whether to use the Tomcat and/or MySQL products bundled 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.

### 2.1.5 Tomcat

If you choose an existing instance of Tomcat, you are prompted for its location. Enter the correct location or click **Browse** to locate and select another location. Click **Next**. You are prompted for the server port and the shutdown port.

If you choose to install the bundled version of Tomcat, you are prompted for the server port and shutdown port. Take the defaults or enter alternate values. **Click Next**.

### 2.1.6 MySQL

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

| | |
|---|---|
| **Binary Directory** | The directory where the mysql and mysqladmin binaries are located. |
| **IP or Host Name** | The IP address or the name of the computer hosting your exiting MySQL instance. |
| **Port** | The port number that MySQL uses; the default is 3306.<br>**Note**: 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:

| | |
|---|---|
| **User Name** | The user name for the MySQL root database account (you should type "root") |
| **Password** | The password for the MySQL root database account. |
| **Confirm Password** | Re-enter the password to confirm it. |

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

**Note:** Regardless of whether you use an existing or included version of MySQL, the JasperServer installer creates a MySQL database user named jasperadmin. The application server database settings are configured to connect using the jasperadmin database user and the same password that you entered in the screens above.

### 2.1.7 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.
- 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**.

### 2.1.8 Login Password Entry

The installer creates an application user named jasperadmin. This is the administrative user that you can use to log into the JasperServer application after the installation is complete.

You are prompted to enter a password for this administrative user.

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

**Note**: The application user jasperadmin is distinct from the MySQL database user of the same name created in the previous set of screens. It is OK for them to have different passwords.

### 2.1.9 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:

| | |
|---|---|
| **Mail User name** | The name of the user in the mail server that JasperServer can use. |
| **Mail Password** | The password of the mail server user. |
| **Confirm Password** | Re-enter the password to confirm it. |
| **Host Name** | The name of the computer hosting the mail server. |
| **Port** | The port number that the mail server uses. For SMTP, the default is typically 25. |
| **Protocol** | The protocol that the mail server uses. JasperServer only supports SMTP.<br>**Note:** Your entry must be lower case. For example: **smtp** |
| **Email From Address** | The address that will appear in the **From** field on email notifications. |

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

### 2.1.10 Install iReport

iReport. 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 execution.

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

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

### 2.1.11 Ready to Install

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

### 2.1.12 View Release Notes

After the files have been installed, you are prompted to view the release notes text file. If you choose to see the release notes file, it is displayed immediately. You can also view it later from the root of the installation directory.

### 2.1.13 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.

### 2.1.14 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 must start this application server using its own start script.

For other configurations, you must start JasperServer as described in section 3, "Starting and Stopping JasperServer", on page 13.

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 JasperServer login page appears in your system default Browser. For more information, see section 3.3, "Logging into JasperServer", on page 13

### 2.1.15 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

## 2.2 Post-Installation Steps

### 2.2.1 Updates Made During Installation

**Updates to the application server:**

If you installed to an existing JBoss or Tomcat updates were made to the startup scripts for these applications.

Here is a list of the updates:

| Application | Script | Notes |
|---|---|---|
| Tomcat | bin/startup.bat, bin/startup.sh | No modifications. |

**Updates to the MySQL database:**

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

| MySQL Updates | Notes |
|---|---|
| JasperServer database | This holds the JasperServer Application metadata.<br>Also, holds customer business data such as reports, report scheduling, permissions, roles, and users. |
| Foodmart sample database | Database created if sample data install option was chosen |
| Sugarcrm sample database | Database created if sample data install option was chosen |
| Jasperadmin database user created | JasperServer connects to the database using this user. |

# 3 Starting and Stopping JasperServer

Before JasperServer is started, the database it depends on must be running. Then, JasperServer is started by starting the application server that JasperServer is deployed to.

## 3.1 Using JasperServer Start/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 Windows Start Menu:**

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

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

**From Command Line:**

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

## 3.2 Using Start/Stop Scripts Without Bundled Installation

If you used your own existing installation for one 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.

Note: If you used your existing versions of Tomcat or JBoss and MySQL then you should use the start/stop scripts provided by those applications.

## 3.3 Logging into JasperServer

This section assumes that JasperServer is running. If it isn’t, start it as described in the section above.

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 |
| JBoss | 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 launch the JasperServer login page 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 is a sample non-administrative 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.

## 3.4 Starting the Included 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.**

## 3.5 JasperServer Log Files

Log files contain important information about how JasperServer is running.

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.

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.

# 4 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.

Under Linux, 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.

# 5 Install Using the War File Distribution Zip

## 5.1 Introduction

In addition to the installer binaries, the JasperServer application is distributed as a stand-alone WAR file distribution. This distribution is packaged as a ZIP file.

Customers who do not wish to use the installer or who have target configurations other than those supported by the installer should use the WAR file distribution.

### 5.1.1 Applications Supported by the War File Distribution

The instructions in this section and the following sections support these configurations:

| Target Configuration | Instructions Location |
|---|---|
| Tomcat with MySQL | This section |
| JBoss with MySQL | This section |
| Tomcat with PostgreSQL | This section |
| JBoss with PostgreSQL | This section |

### 5.1.2 Application Versions Supported

For information on the specific versions of 3rd party applications that are supported by the WAR file distribution ZIP refer to the release notes for the distribution you are using. The release notes are found in the root of the unpacked distribution ZIP.

## 5.2 Obtain the War File Distribution

The WAR file distribution comes in a ZIP file format. To download the WAR file distribution, go to the downloads section of the JasperForge.org site.

## 5.3 Unpack the War File Distribution

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) |

## 5.4 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 describe the steps used for adding the JasperServer sample data to your new JasperServer instance.
- This document gives example database setup commands that have been tested at JasperSoft. The precise commands to be used with your database instance may be different.
- The database setup commands should be run from the Windows or Linux command line.

### 5.4.1 MySQL Database Setup Steps

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

This document gives 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 documentation for how to set up a database and how to create a database user. Create the following (suggested) database instance, database user and 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;
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.

### 5.4.2 PostgreSQL Database Setups Steps

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

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

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

**Create and Populate the Database**

First move to the scripts/postgresql directory:

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

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-postresql.ddl
\i jasperserverCreateDefaultSecurity-postresql.sql
\q
```

The PostgreSQL database is now setup for use with JasperServer.

## 5.5 Deploy JasperServer War to the Application Server

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

Copy the WAR file:

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

To the following directory:

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

## 5.6 Prepare for Database Configuration

### 5.6.1 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 |
| Hibernate Dialect | MySQLDialect | PostgreSQLDialect |

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

### 5.6.2 Expand 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 runs 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                       (same in Windows)
```

## 5.7 Configure Hibernate Properties File

By default, the hibernate properties are set to run with the MySQL database. If you are running another database then you must update your hibernate configuration properties file.

The properties file is found at the following location:

```
<apache-tomcat>/webapps/jasperserver /WEB-INF/hibernate.properties

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

Edit this file and set the dialect property to the appropriate value.

**MySQL**

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

**PostgreSQL**

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

## 5.8 Deploy JDBC Database Driver

The application server needs the appropriate JDBC driver in order for JasperServer to connect to the database. A set of drivers are included with the JasperServer distribution for your convenience. You may want to check with your database vendor for a more up to date version of the JDBC driver.

Copy one of these JDBC 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 lib location:

```
<apache-tomcat>/common/lib                                                     (Tomcat 5.5)
<apache-tomcat>/lib                                                            (Tomcat 6.x)
<jboss>/server/default/lib                                          (JBoss)
```

## 5.9 Database Connection Steps for Tomcat

By default, the jasperserver WAR file is pre-configured to run with Tomcat.

You must edit this file to specify your database configuration settings. Modify the following file:

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

### 5.9.1 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&amp;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>
```

### 5.9.2 PostgreSQL context.xml for Tomcat

You must update your Tomcat context.xml to configure it properly for PostgreSQL.

The <unpacked-war-dir>/scripts/postgresql directory contains a file named context- postgresql.xml that you can use.

Rename "context- postgresql.xml" to "context.xml" and copy it on top of the existing file in the JasperServer WAR file:

Copy (and rename):

```
   <unpacked-war-dir>/scripts/ postgresql /context- postgresql.xml
```

To:

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

Edit the context.xml, changing items in bold to match your settings:

```
….username="postgres" password="postgres"
   url="jdbc:postgresql://localhost:5432/jasperserver"
   url="jdbc:postgresql://localhost:5432/foodmart"
   url="jdbc:postgresql://localhost:5432/sugarcrm"
```

## 5.10 Database Connection Steps for JBoss

JBoss uses a data source definition file to setup the database connections for JasperServer.

Example data source definition files are found in the <unpacked-war-dir>/scripts/jboss directory for each of the supported database servers.

### 5.10.1 MySQL Datasource for JBoss

The <unpacked-war-dir>/scripts/jboss directory contains a sample js-mysql-ds.xml file.

This data source 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 appropriate values:

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

## 5.11 Extra Configuration Step for JBoss

### Delete log4j.jar

JBoss is normally distributed with the log4j facility enabled. JasperServer also uses log4j and includes the log4j.jar. This causes a (non-fatal) exception on JBoss startup. To clean up this error you should remove the log4j.jar from JasperServer.

Remove the following jar:

<jboss>/server/default/deploy/ jasperserver/WEB-INF/lib/log4j-<ver>.jar

## 5.12 Start JasperServer

To run JasperServer 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.

## 5.13 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 file.

Use the following credentials to login to the system:

Username: jasperadmin (Administrative user)
Password: password

If you logged in successfully, your JasperServer home page appears.

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

## 5.14 Setup and Load 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 described above and you should be able to start and login to JasperServer.

The commands that follow are intended to be run at the Windows or Linux command line.

### 5.14.1 MySQL Sample Database Create and Populate

```
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
```

### 5.14.2 PostgreSQL Sample Database Create and Populate

```
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
```

### 5.14.3 Check Database Config for Sample Data

You already configured and tested the Tomcat context.xml or the JBoss datasource file for connecting to the jasperserver database.

This would be a good time to check that you have the settings correct for the two databases you created above: foodmart and sugarcrm. These files were configured in section 5.9, "Database Connection Steps for Tomcat", or section 5.10 "Database Connection Steps for JBoss", above.

# 5.15 Import the JasperServer Sample Metadata

## 5.15.1 Prepare for Running the Import Utility

The JasperServer import utility is used to load metadata into the JasperServer repository. This metadata fully describes objects such as reports, report schedules, permissions, and datasources. The database that is updated is the jasperserver database.

The import utility reads a “repository catalog” file that is a set of XML and binary files organized in a directory style structure.

The shell script that runs the import operation is found in the scripts directory. First make sure that the import-export utility is properly configured for execution:

You must edit the following file to specify your database settings:

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

You should use the same settings that you did in earlier database configuration sections.

### MySQL Settings

Refer to Appendix A, "Configuring the Import-Export Utility", section A.3, "Sample Settings for MySQL".

### PostgreSQL Settings

Refer to Appendix A, "Configuring the Import-Export Utility", section A.3, "Sample Settings for PostgreSQL".

## 5.15.2 Run the Import Utility

To run the import utility, enter the following command:

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

**Notes:**

- The import command depends on the JAVA_HOME variable being set in your environment.
- There are two dashes (--) in front of the command option.

# 5.16 Report Scheduling Configuration (Optional)

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
```

Refer to section 2.1.9, "Scheduled Report Confirmation", for information on scheduled reporting configuration options.

# 5.17 Restart JasperServer

Since you may have made configuration changes to your JasperServer instance, you might want to restart JasperServer.

## 5.18 Update XMLA Connection Definitions (Optional)

JasperServer is able to make XMLA connections over the Web Services interface. These http-based connections use a JasperServer application user account for authentication. You may have different usernames and passwords than the defaults that get loaded from the sample data load in the sections above.

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 via the UI.

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.

## 5.19 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, refer to section B.2, "Database Connectivity Errors", in the Troubleshooting Appendix B.

### Error Running Report

If you have trouble running reports in your new JasperServer Instance, refer to section B3, "Error Running a Report" in the Troubleshooting Appendix B.

# 6 Upgrade from JasperServer 2.0.1 to JasperServer 2.1

This is the section to use if you have an existing JasperServer 2.0.1 application and you want to upgrade to JasperServer 2.1.

This procedure describes the exact steps to carry out this upgrade under a Tomcat application server and a MySQL database configuration. The intention is that the same steps can be applied to other combinations of application server and database - the upgrade script names vary slightly for databases other than MySQL.

## 6.1 Backup JasperServer War File and JasperServer Database

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

### 6.1.1 Backup the JasperServer War Archive

Backup the jasperserver directory in Tomcat to a backup directory:

```
cd <apache-tomcat>

mkdir js-2.0.1-war-backup

copy <apache-tomcat>/webapps/ jasperserver   to  <apache-tomcat>/js-2.0.1-war-backup

delete directory  <apache-tomcat>/webapps/jasperserver
```

### 6.1.2 Backup the JasperServer Database

```
cd <unpacked-war-dir-2.0.1>   (i.e., the location of your original unpacked 2.0.1 WAR file distribution)
```

Windows

```
mysqldump --user=jasperadmin --password=<your-password>  jasperserver > js-db-2.0.1-dump.sql
```

Linux

```
mysqldump --user=jasperadmin --password=< your-password > --host=127.0.0.1 jasperserver > js-db-2.0.1-dump.sql
```

You can use the jasperadmin user or the root user to carry out this operation. Refer to section 6.8.1, "Using mysqldump for Database Backup", for more information on running the mysqldump command.

## 6.2 Unpack JasperServer 2.1 War File Distribution

It is assumed that you have already obtained the jasperserver-2.1-bin.zip distribution.

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

```
<unpacked-war-dir-2.1>
```

## 6.3 Upgrade the JasperServer War

You have already backed up and removed your 2.0.1 WAR file. Now you can deploy the 2.1 WAR file into Tomcat:

### 6.3.1 Deploy the JasperServer WAR file

Deploy the WAR file:

copy <unpacked-war-dir-2.1>/jasperserver to <apache-tomcat>/webapps

### 6.3.2 Expand the Archived War File

The jasperserver.war file is “archived" so you cannot directly edit files inside of it. You must update the database configurations and other configurations inside the WAR, so you must "un-archive" the WAR file first.

Follow the steps in section 5.6.2 "Expand the Archived War File" to un-archive the jasperserver.war.

### 6.3.3 Check for Local Configurations

In order to preserve your application server configurations for the database and for other local settings, you should copy over the following configuration files from your 2.0.1 war to your 2.1 war.

Copy from <apache-tomcat>/js-2.0.1-war-backup>/jasperserver to <apache-tomcat>/webapps/jasperserver:

META-INF/context.xml

WEB-INF/hibernate.properties

WEB-INF/js.mail.properties

Also, copy or integrate any other custom modifications you have made to your JasperServer 2.0.1 instance.

**Note**: Refer to section 5.5 “Deploy JasperServer War to the Application Server”, and section 5.6.2 “Expand the Archived War File”, for general information and troubleshooting when deploying a WAR to Tomcat.

## 6.4 Upgrade the JasperServer Database

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

```
cd <unpacked-war-dir-2.1>/scripts/upgrade
```

Login to your mysql client:

```
mysql –u root –p
```

Run the upgrade scripts:

```
mysql>use jasperserver;

mysql>source upgrade-mysql-2.0.1-2.1.0.sql;
```

## 6.5 Start JasperServer 2.1

Before you actually start Tomcat with the new 2.1 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.

## 6.6 Login to JasperServer 2.1

If Tomcat and JasperServer 2.1 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>
```

Your JasperServer instance has now been upgraded to 2.1.

If there are problems on startup refer to section B.2, "Database Connectivity Errors" in the Troubleshooting Appendix B.

## 6.7 Sample Data Upgrade (Optional)

There are some updates to the sample data that you may want to look at. For instance, the sample foodmart database has an upgrade to set display dates to 2006-2007 (instead of the old 1997-1998).

### 6.7.1 Delete Repository Items

The sample data import run in the next section will recreate any reports/views that are removed.

Login to JasperServer as jasperadmin and remove the following folder:

> **(Warning:** Do not remove this folder if you have placed any of your Production objects in this folder. Instead, you can individually remove each of the sample objects. Or you can skip this optional step.)
>
> /analysis                                        (remove this folder)

### 6.7.2 Update the Foodmart Database

```
cd <unpacked-war-dir-2.1>/scripts/upgrade/sample                    (Note: be in the sample directory)
```

Login to your mysql client:

```
mysql –u root –p
```

Run the upgrade script:

```
mysql>use foodmart;
mysql>source upgrade-mysql-sample-foodmart-2.1.0.sql;
```

### 6.7.3 Import New Sample Data

The import operation does not overwrite existing repository objects. So, only sample data that is new (or has been deleted) will be created.

```
cd <unpacked-war-dir-2.1>/scripts
js-import.bat --input-dir js-catalog                              (Windows)
js-import.sh --input-dir js-catalog                               (Linux)
```

Refer to Appendix A, "Configuring the Import-Export Utility", for information on running the import utility.

## 6.8 Additional Notes on JasperServer Upgrade

### 6.8.1 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-<ver>-dump.sql
```

The following command performs the same task on a Linux system:

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

**Notes:**

- All the options in the commands above begin with two dashes (--).
- 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

### 6.8.2 Additional Configuration Files

If you made modifications or customizations to your JasperServer 2.0.1 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 2.0.1 environment and re-integrated into the new 2.1 environment.

### 6.8.3 Clear the Tomcat Work Directory

This directory path is where .jsp files are compiled and other objects are stored. These can potentially cause errors when updating to a new WAR file. It is a good practice to clear the work directory when doing an upgrade.

Change directory:

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

Delete all files under the work directory.

# 7 Upgrade Notes for JasperServer 1.2

If you are upgrading from an older JasperServer version such as 1.2 or 2.0 then you must run additional database upgrade scripts.

In the upgrade steps specified in the previous section, before you run the steps in section 6.4 "Upgrade the JasperServer Database", you should first run the steps below:

```
mysql>use jasperserver;

mysql>source upgrade-mysql-1.2.0-2.0.0.sql;

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

The same principle should be applied to JasperServer instances older than 1.2 or 2.0.

# 8 Adding Password Encryption to JasperServer

## 8.1 Introduction

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

By default password encryption turned off.

This section describes the procedure to enable password encryption. For more information on JasperServer encryption options refer to the *JasperServer Administrator Guide*.

## 8.2 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.

## 8.3 Backup your JasperServer Database

As a precaution, you must 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. Refer to section 6.8.1, "Using mysqldump for Database Backup", for more information on running the mysqldump command.

### PostgreSQL

Refer to your PosrgreSQL documentation for details.

## 8.4 Stop Application Server

You can now stop your application server. You should leave your database running.

## 8.5 Run Repository Export Utility

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

To create the export catalog, 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.

Refer to Appendix A, "Configuring the Import-Export Utility", for information on running the export utility.

## 8.6 Specify Encryption Settings in JasperServer War

JasperServer uses Spring configuration and acegi security to enable the encryption configuration. There is more specific information in the *JasperServer Administrator Guide* about the security algorithms and settings. These options can allow you to have a strong encryption setup. This section is focused 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 causes the passwordEncoder logic to be used. After removing the commenting characters the line should look like the following:

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

### 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 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 must 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: As described in the *JasperServer Administrator 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.

## 8.7 Specify Encryption Settings for Import Utility

Before starting JasperServer, you must convert the plain text passwords that are currently stored in the repository export catalog that you created in steps above (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 must 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, set the same configuration options that you set in the previous section.

## 8.8 Drop and Recreate the JasperServer Database

Next, 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

Change directory to:

```
….cd <js-install>/scripts/postgresql
```

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

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

```
   psql -U postgres
```

Enter the following commands:

```
   drop database jasperserver;

   create database jasperserver encoding='utf8';

   \c jasperserver

   \i jasperserverCreate-postgresql.ddl

   \i jasperserverCreateDefaultSecurity-postgresql.sql
```

## 8.9 Run Repository Import Utility

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

To import your backup catalog to the repository, 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.

Refer to Appendix A, "Configuring the Import-Export Utility", for information on running the import utility.

## 8.10 Start Application Server

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

## 8.11 Login to JasperServer

You can now login to JasperServer.

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 A Configuring the Import-Export Utility

## A.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 Administrator Guide* for more information on command options for the import and export utility.

## A.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 to use or modify 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 |

## A.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 must 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

### 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**

### 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

metadata.jdbc.username=**postgres**

metadata.jdbc.password=**postgres**

## A.4 Deploy Database Driver

In order for the import-export utility to run, it will need the proper JDBC driver. This allows a connection to be made to the JasperServer database.

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

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

Drivers can be found here:

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

## A.5 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 displays the command options:

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

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

## A.6 Home Directory Evaluation under Linux

In a Linux shell, you can use the ~/ syntax to refer to the home directory. However, because the import-export utility is a Java application, which does not necessarily handle shell specific syntax, this specific syntax is not correctly evaluated.

Therefore, if you would like to refer to the home directory when running the import-export command under Linux, you will have to give the absolute path to the home directory.

For instance:

```
./js-export.sh  --uris /reports  --output-dir /home/user/my-reports                (instead of ~/my-reports)
```

# Appendix B Troubleshooting

## B.1 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.

## B.2 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.

### B.2.1 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 validates the database 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.

#### PostgreSQL

Start psql using the same account credentials used in your JasperServer database config files:

```
psql -U postgres
```

### B.2.2 Check 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/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-postgresql-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

### B2.3 Special Configuration Case under 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 takes the following steps:

- Unpack the jasperserver.war file into a new directory named jasperserver.
- Take the jasperserver/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 overrides the context.xml found in your jasperserver directory.

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

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

## B.3 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 when you installed JasperServer, you have a number of reports and data sources that are 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.

## B.4 Database Error after Changing MySQL 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 database configuration files to verify your port number.

If it is incorrect, change it to the correct port number, save the file, and restart the application server. See the section above "Configuration File Locations" above for more information.

## B.5 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 configure 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 documentation see the section "Identifier Case Sensitivity" for more information.

Table and column names in PostgreSQL are case-sensitive.

## B.6 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/or

- 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.

## B.7 Java Out of Memory Error

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

This Java option is set within the application server, so you must 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=64m -XX:MaxPermSize=128m
```

## B.8 Error Running Scheduled Report

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

```
JDBC exception on Hibernate data access
org.hibernate.exception.GenericJDBCException: could not insert
```

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

Add the following setting:

```
max_allowed_packet=32M
```

## B 9 Upgrade: Error During JasperServer 1.2 Export

Upgrading from 1.2 to 2.0 typically 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.

## B.10 Memory Settings on JBoss Installation

The JBoss installation (using the installer) updates the <jboss>/bin/run.sh or run.bat with an additional settings on the JAVA_OPTS.

There are additional settings for the JVM permGen space. This is because JBoss has a larger memory footprint than Tomcat, and we have found that when running JasperServer, it is easy to run out of the permGen space. The Permanent Generation space is memory reserved for meta information on application classes and methods.

The permGen size maximum is typically set to 64Meg in a Java virtual machine.

After installing JasperServer under the JBoss application you should see permGen setting like the following:

```
-XX:PermSize=64m -XX:MaxPermSize=128m
```

## B.11 Log4j Error on JBoss Startup

Typical exception message:

```
log4j:ERROR "org.jboss.logging.util.OnlyOnceErrorHandler
```

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 the OnlyOnceErrorHandler logging error. This error is not fatal to JasperServer, but can cause confusion when seen at JBoss startup.

To get rid of this error, you should delete the log4j.jar found in JasperServer.

Delete the following jar:

```
<jboss>/server/default/deploy/jasperserver.war/WEB-INF/lib/log4j-<ver>.jar
```