TIBCO™ Jaspersoft®

TIBCO JasperReports® Server

Upgrade Guide

Release 6.0

http://www.jaspersoft.com

This is version 1114-JSP60-03 of the *JasperReports Server Upgrade Guide*.

TABLE OF CONTENTS

Chapter 1 Introduction

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

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

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

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

For business intelligence users, Jaspersoft offers TIBCO™ Jaspersoft® OLAP, which runs on the server.

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

You can use the following sources of information to extend your knowledge of JasperReports Server:

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

- Our Online Learning Portal lets you learn at your own pace, and covers topics for developers, system administrators, business users, and data integration users. The Portal is available online from Professional Services section of our website.
- Our free samples, which are installed with JasperReports, Jaspersoft Studio, and JasperReports Server, are documented online.

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

This chapter contains the following sections:

- **JasperReports Server Upgrade Distributions**

1.1 JasperReports Server Upgrade Distributions

There are two main distribution packages for JasperReports Server.

Distribution Package	Description
Overlay Upgrade zip	Supports upgrade starting with version 4.0. Supports only the Apache Tomcat application server. Supports all certified repository databases. Supports upgrade and rollback of upgrade changes. Supports assistance with identifying customized files in your environment. Supports Windows, Linux, Mac, and other platforms. Only available with the Commercial version of JasperReports Server. File name is: **jasperreports-server-overlay-6.0.zip**
WAR File Distribution Zip	Supports upgrade starting with version 3.7. Supports all certified application servers. Supports all certified repository databases. Supports Windows, Linux, Mac, and other platforms. File name is: **jasperreports-server-6.0-bin.zip**

1.1.1 About Bundled Apache Ant

The Overlay Upgrade ZIP and the War File Distribution ZIP come with a bundled version of Apache Ant so you do not need to download or install Ant. The Ant scripts used for upgrade come with Windows and Linux batch scripts that are pre-configured to use the bundled version of Apache Ant.

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

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

On Linux and Solaris, the Ant commands may not be compatible with all shells. If you have errors, use the `bash` shell explicitly. For more information, see the information on the bash shell in the Troubleshooting appendix of the *JasperReports Server Installation Guide*.

CHAPTER 2 OVERLAY UPGRADE

This chapter describes how to upgrade to JasperReports Server 6.0 using the overlay upgrade package. This chapter contains the following sections:

2.1 Introduction to the Overlay Upgrade

JasperReports Server now has an overlay upgrade procedure. For now, this is only available with JasperReports Server Commercial edition installed with the WAR file and only with the Apache Tomcat application server.

- The **overlay upgrade only supports Apache Tomcat as the application server**.
- The **overlay upgrade only supports JasperReports Server installed using the WAR file**. The binary installer is not supported.
- All certified databases are supported.

The overlay upgrade supports upgrading from the following JasperReports Server versions:

4.0, 4.0.1, 4.1, 4.2.1, 4.5.1, 4.5.2, 4.7, 4.7.1, 5.0, 5.0.1, 5.1, 5.2, 5.5, 5.6, 5.6.1

to JasperReports Server 6.0.

Although the overlay upgrade does offer a rollback feature, you should always back up your database and application before upgrading.

This section will use a 5.0 to 6.0 upgrade as an example.

2.2 Upgrade Steps Overview

These are the general steps used in this section:

1. Back up your current JasperReports Server instance.

   (The overlay tool will additionally back up your war file automatically and ask you if you have backed up your database.)
2. Download and unpack the new JasperReports Server overlay upgrade 6.0 package zip file.
3. Run the upgrade steps.

The overlay upgrade procedure will assist you in handling any custom modifications or extensions that you have made to your JasperReports Server instance.

Best practice is to always back up your application and database before upgrading.

2.3 Back Up Your JasperReports Server Instance

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

The following backup example is for Tomcat with the PostgreSQL database. For other databases please consult your DB administration documentation for back up information.

Back up your JasperReports Server War File :

1. Create a folder location where you can save your `jasperserver-pro` war file. For example, `C:\JS_50_BACKUP` or `/opt/JS_50_BACKUP`.
2. Copy `<tomcat>/webapps/jasperserver-pro` to `<path>/JS_50_BACKUP`

Back up your jasperserver Database:

1. Create a folder location (if you did not do so in the step above) where you can save your `jasperserver` database, For example, `C:\JS_50_BACKUP` or `/opt/JS_50_BACKUP`.
2. Run the following commands for PostgreSQL:

```
cd <path>/JS_50_BACKUP
pg_dump --username=postgres  jasperserver  >  js-db-50-dump.sql
```

2.4 Unpack the Overlay Upgrade Package

The overlay upgrade package comes in a file named: `jasperreports-server-overlay-6.0.zip`.

1. Download the overlay upgrade package from Jaspersoft technical support (http://support.jaspersoft.com) or contact your sales representative.
2. Extract all files from `jasperreports-server-overlay-6.0.zip`. Create or choose a destination folder, such as `C:\JS_OVERLAY` on Windows, `/home/<user>/JS_OVERLAY` on Linux, or `/Users/<user>/JS_OVERLAY` on Mac.
3. The overlay upgrade package will unpack into a folder named:

```
overlay
```

This folder location will be known as:

```
<overlay-folder>
```

2.5 Check for JDBC Driver (Oracle, SQL Server, DB2)

As of version 5.6.1, JasperReports Server uses the TIBCO JDBC drivers for the following commercial databases: Oracle, SQL Server, or DB2. If you want to use a different JDBC driver, you need to copy it to the correct location. Oracle and DB2 users must also use the current version of the db.template.properties file.

Oracle example:

The alternate JDBC jar should be copied into the following location:

```
<overlay-folder>/buildomatic/conf_source/db/oracle/jdbc
```

You also need to copy your existing version of the db.template.properties file to the following location:

<overlay-folder>/buildomatic/conf_source/db/oracle/db.template.properties

DB2 example:

For DB2, the alternate JDBC jar should be copied into the following location:

```
<overlay-folder>/buildomatic/conf_source/db/db2/jdbc
```

You also need to copy your existing version of the db.template.properties file to the following location:

<overlay-folder>/buildomatic/conf_source/db/db2/db.template.properties

SQL Server example:

For SQL Server, the alternate JDBC jar should be copied into the following location:

```
<overlay-folder>/buildomatic/conf_source/db/sqlserver/jdbc
```

In addition, for SQL Server, modify the default_master.properties file in your existing installation to comment out the maven version:

1. Open the default_master.properties in your existing installation in a text editor.
2. Locate the line:

   ```
   maven.jdbc.version=1.6
   ```

   and change it to:

   ```
   # maven.jdbc.version=1.6
   ```

3. Make sure to back up the modified file.

For MS SQL Server, comment out the maven version information in your existing copy of default_master.properties.

2.6 Run the Overlay Upgrade

The overlay upgrade only works with the Tomcat application server. All certified JasperReports Server databases are supported. You can perform the overlay upgrade whether or not you have local customizations.

1. Stop the Tomcat application server
2. Your database should be running

3. Run the following commands:

   ```
   cd <overlay-folder>
   overlay install
   ```

- You are prompted to specify a path to a working folder:

  You can accept the default value or specify an alternate location

  Press `enter` to accept the default "`../overlayWorkspace`"
- You are prompted to back up your `jasperserver` database:

  You can back up your database now if you have not already done so

  Choose "y" for `yes` to continue
- You are prompted to shutdown your Tomcat instance:

  You can stop Tomcat now if you have not already done so

  Choose "y" for `yes` to continue
- You are prompted to specify a path to your `master.properties` file:

  For a 5.0 instance it might be similar to:

  ```
  C:\Jaspersoft\jasperreports-server-5.0\buildomatic\default_master.properties
  /opt/jasperreports-server-5.0/buildomatic/default_master.properties
  ```

  Enter the full path and file name for your `default_master.properties` file
- For final verification, the overlay prompts you for the path to your application server:

  If you haven't moved it, it will be located at path to: `<tomcat>`

  Press `enter` to accept the default if it is correct
- The overlay will now begin making updates to your system:

  Your `jasperserver-pro` war file will be automatically backed up

  Potential customizations in your environment will be analyzed

  - You are prompted to review the report on customizations if you choose to:

    Choose "y" for `yes` to continue with the upgrade

    The `jasperserver` database will be upgraded

    The `jasperserver-pro` war file will be upgraded

    The core data resources will be upgraded in the `jasperserver` repository database
- When the overlay execution has completed, start Tomcat, and log in to test the upgraded JasperReports Server.

  If execution was successful, you will see BUILD SUCCESSFUL on the command line

2.7 Rerun the Overlay Upgrade

If you exited out of the `overlay install` for any reason, you can resume execution without re-answering the prompted questions that you already answered. This execution behavior is the default when re-running the `overlay`.

You can re-run the overlay by simply running the same command:

```
overlay install
```

By default, under the covers, the overlay will run in a mode of: `resumeMode=true`

This means that any previous prompts you answered will be remembered.

If you would like to re-run the `overlay` “from scratch”, you can run the following command:

```
overlay install -DresumeMode=false
```

To see more information on the `overlay` options you can run:

```
overlay help
```

2.8 Rollback Procedure

In the case of an error with the overlay upgrade, use the following rollback procedure:

1. Stop Tomcat.
2. Run the following command:

   ```
   overlay rollback
   ```
3. Specify the path to the working folder:

   The default is “`../overlayWorkspace`”
4. The tool will ask if you have rolled back your JasperReports Server database:

   The default is `no`

It is required that you restore your database manually.

5. When the tool has finished, restore your database (see below), start Tomcat, and test JasperReports Server.

To restore your JasperReports Server Database:

1. Go to the directory location where you saved the backup of your `jasperserver` database

   For example, `C:\JS_50_BACKUP` or `/opt/JS_50_BACKUP`.
2. Run the following commands for PostgreSQL:

   ```
   cd /opt/JS_50_BACKUP
   pg_restore --username=postgres  jasperserver  <  js-db-50-dump.sql
   ```

2.9 Starting and Logging into JasperReports Server 6.0

Start your application server. Your database should already be running.

2.9.1 Clearing Your Browser Cache

Before you log in, make sure you and your end-users clear the browser cache. JavaScript files, which enable UI elements of JasperReports Server, are typically cached by the browser. Clear the cache to ensure that the newer files are used.

If the UI for JasperReports Server looks broken or incorrect, it is often because of needing to clear the browser cache.

2.9.2 Logging into JasperReports Server

Log in using the following URL, user IDs, and passwords:

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

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

Your JasperReports Server instance has now been upgraded to 6.0. In the event of startup or login problems, refer to the Troubleshooting appendix of the *JasperReports Server Installation Guide*.

2.10 Additional Tasks to Complete the Upgrade

For additional information about optional steps after the main upgrade steps are complete, see **4.9, "Additional Tasks to Complete the Upgrade," on page 33**.

2.10.1 Clearing the Application Server Temp Folder

JasperReports Server uses caching to speed operations within the application. In the application server, caching files are created and stored for this caching functionality. Typically, these cached files are stored in a `temp` folder. Clear this `temp` folder to avoid any conflicts after the upgrade is complete. For Apache Tomcat the `temp` folder is `<tomcat>/temp`. (In general, the `temp` folder used by an Application Server corresponds to the path pointed at by the `java.io.tmpdir` Java system property.)

To clear the temp folder in Apache Tomcat:

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

2.11 Running Overlay Upgrade a Second Time

If you run the overlay upgrade a second time, the overlay logic will ask if you want to resume the last run of the overlay, so that your previous answers to questions are remembered and reused.

The overlay procedure will ask:

"We have detected that overlay install was already run. Do you want to resume last run? Default is 'y' ([y], n):"

Choose "y" for `yes` if you do not want to change any information previously given to the overlay

Choose "n" for `no` if you would like to enter new or different information

One reason for entering "n" for `no` would be if you did not give a valid path to your `default_master.properties` the first time you executed the overlay.

CHAPTER 3 UPGRADING FROM 5.6 TO 6.0

This chapter describes the recommended procedure for upgrading from JasperReports Server:

5.6 or 5.6.1

to JasperReports Server 6.0.

The examples describe how to use the js-upgrade shell scripts to carry out the upgrade operation.

This chapter contains the following sections:

3.1 Upgrade Steps Overview

These are the general steps used in this section:

1. Back up your current JasperReports Server instance.
2. Download and set up the new 6.0 JasperReports Server WAR file distribution zip.
3. Run the js-upgrade script as described in **3.5, "Upgrading to JasperReports Server 6.0," on page 21**.

If your current instance of JasperReports Server has any custom modifications or extensions, keep track of these and re-integrate them into your 6.0 instance after upgrading.

3.2 Back Up Your JasperReports Server Instance

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

The following backup example is for Tomcat with the PostgreSQL or MySQL database. For other databases, consult your DB administration documentation for back up information.

Back up your JasperReports Server WAR File:

1. Create a folder location where you can save your `jasperserver-pro` war file. For example, `C:\JS_55_BACKUP` or `/opt/JS_55_BACKUP`.
2. Copy `<tomcat>/webapps/jasperserver-pro` to `<path>/JS_55_BACKUP`

Back up your Jasperserver Database:

1. Create a folder location (if you did not do so in the step above) where you can save you `jasperserver` database, For example, `C:\JS_55_BACKUP` or `/opt/JS_55_BACKUP`.
2. Run the following commands for PostgreSQL or MySQL:
   - PostgreSQL

     ```
     cd <path>/JS_55_BACKUP
     pg_dump --username=postgres  jasperserver  >  js-db-5.5-dump.sql
     ```
   - MySQL on Windows

     ```
     cd <path>/JS_55_BACKUP
     mysqldump --user=root --password=<password> jasperserver > js-db-5.5-dump.sql
     ```
   - MySQL on Linux

     ```
     cd <path>/JS_55_BACKUP
     mysqldump --user=root --password=<password> --host=127.0.0.1 jasperserver > js-db-5.5-dump.sql
     ```

For MySQL, If you receive an error about packet size, see the Troubleshooting appendix of the *JasperReports Server Installation Guide*.

3.3 Preparing the JasperReports Server 6.0 WAR File Distribution

Use the buildomatic `js-upgrade` scripts included in the 6.0 WAR file distribution ZIP release package to carry out the upgrade. Follow these steps to obtain and unpack the WAR file distribution ZIP file:

1. The WAR file distribution comes in a file named `jasperreports-server-6.0-bin.zip` in the compressed ZIP format. Download the WAR file distribution from Jaspersoft technical support (http://support.jaspersoft.com) or contact your sales representative. .
2. Extract all files from `jasperreports-server-6.0-bin.zip`. Choose a destination, such as a `C:\Jaspersoft` folder on Windows, `/home/<user>` on Linux, or `/Users/<user>` on Mac.

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

```
<js-install-6.0>
```

3.4 Configuring Buildomatic for Your Database and Application Server

This upgrade procedure uses the `js-upgrade-samedb` shell script.

For Unix, the bash shell is required for the `js-upgrade` scripts. If you are installing to a non-Linux Unix platform such as HP-UX, IBM AIX, FreeBSD or Solaris, you need to download and install the bash shell. See the Troubleshooting appendix of the *JasperReports Server Installation Guide* for more information.

This section shows example configurations for the PostgreSQL, MySQL, and Oracle databases. Other databases are similar.

3.4.1 Example Buildomatic Configuration

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

3.4.1.1 PostgreSQL Example

This example shows how to configure `default_master.properties` for PostgreSQL.

1. Locate the `postgresql_master.properties` sample configuration file:

Database	Master Properties File
PostgreSQL	<js-install-6.0>/buildomatic/sample_conf/postgresql_master.properties

2. Copy the file to `<js-install-6.0>/buildomatic`
3. Rename the file to `default_master.properties`
4. Edit `default_master.properties` for your database and application server:

Database	Sample Property Values

PostgreSQL	`appServerType=tomcat6 [tomcat7, tomcat6, jboss, glassfish2, glassfish3]`
	`appServerDir=c:\\apache-tomcat-6.0.26 (for example)`
	`dbUsername=postgres`
	`dbPassword=postgres`
	`dbHost=localhost`

3.4.1.2 MySQL Example

This example shows how to configure `default_master.properties` for MySQL.

1. Locate the `mysql_master.properties` sample configuration file:

Database	Master Properties File
MySQL	<js-install-5.5>/buildomatic/sample_conf/mysql_master.properties

2. Copy the file to `<js-install-6.0>/buildomatic`
3. Rename the file to `default_master.properties`
4. Edit `default_master.properties` for your database and application server:

Database	Sample Property Values

MySQL	`appServerType=tomcat6 [tomcat7, tomcat6, jboss, glassfish2, glassfish3]`
	`appServerDir=c:\\Apache Software Foundation\\Tomcat-6 (for example)`
	`dbUsername=root`
	`dbPassword=password`
	`dbHost=localhost`

3.4.1.3 Oracle Example

This example shows how to configure `default_master.properties` for Oracle.

1. Locate the `oracle_master.properties` sample configuration file:

Database	Master Properties File
Oracle	<js-install-6.0>/buildomatic/sample_conf/oracle_master.properties

2. Copy the file to `<js-install-6.0>/buildomatic`
3. Rename the file to `default_master.properties`
4. Edit `default_master.properties` for your database and application server:

Database	Sample Property Values

Oracle	`appServerType=tomcat6 [tomcat7, tomcat6, jboss, glassfish2, glassfish3]`
	`appServerDir=c:\\Apache Software Foundation\\Tomcat-6 (for example)`
	`dbUsername=jasperserver`
	`dbPassword=password`
	`sysUsername=system`
	`sysPassword=password`
	`dbHost=localhost`

5. As of 5.6.1, JasperReports Server includes the TIBCO JDBC drivers for the following commercial databases: Oracle, SQL Server, or DB2. If you want to use a different JDBC driver, you need to copy it to the correct location and edit default_master.properties before running the upgrade steps.

   With Oracle, for example, the alternate JDBC jar should be copied into the following location:

   ```
   <js-install-6.0>/buildomatic/conf_source/db/oracle/jdbc
   ```

   For SQL Server or DB2, the driver would get copied to the following locations respectively:

   ```
   <js-install-6.0>/buildomatic/conf_source/db/sqlserver/jdbc
   <js-install-6.0>/buildomatic/conf_source/db/db2/jdbc
   ```

   Once you have copied the driver, in your default_master.properties file, go to the Setup Standard JDBC Driver section and edit it according to the instructions in that section.

3.4.2 Additional Step when Using JBoss 7 (and Oracle, SQL Server, or DB2)

If you are using JBoss 7 as your application server and Oracle, SQL Server, or DB2 as your database, and you are using a driver other than the TIBCO JDBC driver, there is an additional set of required steps to handle the JDBC driver.

You will need to make an explicit reference to your JDBC driver file name so that JBoss 7 will know the exact file name.

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

   Edit: `<js-install-6.0>/buildomatic/default_master.properties`

   Look for the section "Setup JDBC Driver"

   Uncomment and edit the two lines shown below:

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

So that they look like this:

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

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

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

   Edit: `<js-install-6.0>/buildomatic/install_resources/jboss/jboss-deployment-structure.xml`

   Look for the section "Setup JDBC Driver"

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

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

   So that it looks like this:

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

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

3.5 Upgrading to JasperReports Server 6.0

Now that your buildomatic scripts have been configured, you can complete the upgrade.

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

Make sure you have backed up your old JasperReports Server WAR file before proceeding.

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

Commands	Description
`cd <js-install-6.0>/buildomatic`	
`js-upgrade-samedb.bat`	(Windows) Upgrade jasperserver-pro war file, upgrade jasperserver database to 6.0, add 6.0 repository resources into the database
`./js-upgrade-samedb.sh`	(Linux) Upgrade jasperserver-pro war file, upgrade jasperserver database to 6.0, add 6.0 repository resources into the database

3.5.1 js-upgrade Test

You can run the `js-upgrade` scripts in test mode using the `test` option. For example, in Windows enter:

```
cd <js-install-6.0>/buildomatic
```

```
js-upgrade-samedb.bat test
```

In `test` mode, the js-upgrade scripts check your default_master.properties settings. The application server location is validated and the capability to connect to the specified database is validated. Using `test` mode can help debug issues, such as an incorrect database password. Your system is not altered when executing the script in `test` mode.

3.5.2 Output Log Location

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

The output log file is located here:

```
<js-install-6.0>/buildomatic/logs/js-upgrade-<date>-<number>.log
```

3.5.3 Errors

If you encounter errors during the `js-upgrade` script execution, first look at the output log to see if you can spot any errors. Additionally, you should refer to the Troubleshooting appendix of the *JasperReports Server Installation Guide*. The information in this appendix applies to both `js-upgrade` scripts and the `js-install` scripts.

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

3.6 Starting and Logging into JasperReports Server 6.0

Start your application server. Your database should already be running.

3.6.1 Clearing Your Browser Cache

Before you log in, make sure you and your end users clear the browser cache. JavaScript files, which enable the UI elements of JasperReports Server, are typically cached by the browser. Clear the cache to ensure that the newer files are used.

3.6.2 Logging into JasperReports Server

Log in using the following URL, user IDs, and passwords:

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

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

Your JasperReports Server instance has now been upgraded to 6.0. In the event of startup or login problems, refer to the Troubleshooting appendix of the *JasperReports Server Installation Guide*.

3.7 Additional Tasks to Complete the Upgrade

The tasks described below should be done when the application server is shutdown.

3.7.1 Clearing the Application Server Work Folder

Application servers have work folders where JasperReports Server files are compiled and cached and other objects are stored. When you update the WAR file or license, the buildomatic `deploy-webapp-pro` target should automatically clear the application server's `work` directory, but it's a good practice to double-check. A permission problem, or some other problem, could prevent the clearing of the work folder.

To clear the work folder in Tomcat:

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

3.7.2 Clearing the Application Server Temp Folder

JasperReports Server uses caching to speed operations within the application. In the application server, caching files are created and stored for this caching functionality. Typically, these cached files are stored in a `temp` folder. Clear this `temp` folder to avoid any conflicts after the upgrade is complete. For Apache Tomcat the `temp` folder is `<tomcat>/temp`. (In general, the `temp` folder used by an Application Server corresponds to the path pointed at by the `java.io.tmpdir` Java system property.)

To clear the temp folder in Apache Tomcat:

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

3.7.3 Clearing the Repository Cache Database Table

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

To manually clear the repository cache database table, run a SQL command similar to one shown below:

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

3.8 Old Manual Upgrade Steps: 5.6 to 6.0

This section has the older, manual upgrade steps that were in place before the `js-upgrade` shell scripts were implemented in the 4.0 release. These are provided in the following table as a reference, mainly for internal use. The `js-upgrade` shell scripts execute these buildomatic targets “behind the scenes.” Jaspersoft recommends using the `js-upgrade` scripts described in the beginning of this upgrade chapter instead of these manual steps.

Older buildomatic upgrade steps for this chapter are the following:

Commands	Description
`cd <js-install-6.0>/buildomatic`	
`js-ant upgrade-5.6-6.0-pro`	Execute SQL script to upgrade database to 6.0. Executes script buildomatic/install_resources/sql/<dbType>/upgrade-<dbType>-5.6.0-6.0.0-pro.sql
`js-ant import-minimal-for-upgrade-pro`	Loads themes and other core resources for 6.0. Note: "import-minimal-for-upgrade" will import core resources in an "update" mode so that the older 5.6 core resources will be overwritten. Additionally, the "skip-user-update" option will be applied so that the superuser and jasperadmin users will not have their passwords modified.
`js-ant import-sample-data-upgrade-pro`	(Optional) This step is optional. Loads the 6.0 sample data.
`js-ant deploy-webapp-pro`	Delete old 5.6 war file, deploy the 6.0 war file.

CHAPTER 4 UPGRADING FROM 3.7 - 5.5 TO 6.0

This chapter describes the recommended procedure for upgrading from JasperReports Server:

3.7, 3.7.1, 4.0, 4.0.1, 4.1, 4.2, 4.2.1, 4.5, 4.5.1, 4.7, 4.7.1, 5.0, 5.1, 5.2, or 5.5

to JasperReports Server 6.0.

The upgrade procedures described in this chapter use the JasperReports Server WAR File Distribution ZIP release package and the included buildomatic scripts.

The procedure in this chapter can also be used to upgrade JasperReports Server 5.6 or 5.6.1 to 6.0. However, we recommend you use the procedure in **Chapter 3, "Upgrading from 5.6 to 6.0," on page 17**.

In this chapter the examples shown will use JasperReports Server 5.5 as the version being upgraded from.

This chapter contains the following sections:

- **Upgrade Steps Overview**
- **Planning Your Upgrade**
- **Back Up Your JasperReports Server Instance**
- **Exporting Current Repository Data**
- **Preparing the JasperReports Server 6.0 WAR File Distribution**
- **Configuring Buildomatic for Your Database and Application Server**
- **Upgrading to JasperReports Server 6.0**
- **Starting and Logging into JasperReports Server 6.0**
- **Additional Tasks to Complete the Upgrade**
- **Old Manual Upgrade Steps**

4.1 Upgrade Steps Overview

These are the general steps used in this section:

1. Plan your upgrade.
2. Back up your current JasperReports Server instance.
3. Export your existing repository data. For example, export your 5.5 data.
4. Download and set up the new 6.0 JasperReports Server WAR file distribution zip.
5. Run the js-upgrade script as described in **4.7, "Upgrading to JasperReports Server 6.0," on page 31**.

If your current instance of JasperReports Server has any custom modifications or extensions, keep track of these and re-integrate them into your 6.0 instance after upgrading.

4.2 Planning Your Upgrade

There is a planning chapter you should review in order to see if there are changes which will affect your deployment. This chapter can be found here: **Appendix A, "Planning Your Upgrade ," on page 43**.

4.3 Back Up Your JasperReports Server Instance

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

The following backup example is for Tomcat with the PostgreSQL or MySQL database. For other databases, consult your DB administration documentation for back up information.

Back up your JasperReports Server War File:

1. Create a folder location where you can save your `jasperserver-pro` war file. For example, `C:\JS_55_BACKUP` or `/opt/JS_55_BACKUP`.
2. Copy `<tomcat>/webapps/jasperserver-pro` to `<path>/JS_55_BACKUP`

Back up your jasperserver Database:

1. Create a folder location (if you did not do so in the step above) where you can save your `jasperserver` database, For example, `C:\JS_55_BACKUP` or `/opt/JS_55_BACKUP`.
2. Run the following commands for PostgreSQL or MySQL:
   - PostgreSQL

     ```
     cd <path>/JS_55_BACKUP
     pg_dump --username=postgres  jasperserver  >  js-db-5.5-dump.sql
     ```

   - MySQL

     ```
     cd <path>/JS_55_BACKUP
     ```

Windows:

```
mysqldump --user=root --password=<password> jasperserver > js-db-5.5-dump.sql
```

Linux:

```
mysqldump --user=root --password=<password> --host=127.0.0.1 jasperserver >
js-db-5.5-dump.sql
```

For MySQL, If you receive an error about packet size, see the Troubleshooting appendix of the *JasperReports Server Installation Guide*.

4.4 Exporting Current Repository Data

You need to export your old repository data, for example your 5.5 repository data, using the JasperReports Server export utility. You can export in the following ways:

- Use the JasperReports Server UI (since 5.0 release)
- Use the buildomatic scripts (if you originally installed using buildomatic).
- Use the `js-export.bat/.sh` script found in the `<js-install>/buildomatic` folder.

4.4.1 Exporting from the UI

As of JasperReports Server version 5.0, import-export functionality has been added to the User Interface. Administrators who login under the `superuser` account can export all repository data to a file on their machine. To export your 5.5 repository data from the UI do the following:

> Warning: Export from the UI in Release 5.1 has a bug. So, it is best to export from the command line if upgrading from 5.1. If you do export from the UI with 5.1, you can unpack the resulting export.zip file, edit the index.xml file and change the string “5.0.0 CE” to 5.1.0 PRO”. See the *JasperReports Server Administrator Guide* for details.

1. Start a web browser on your server system (so that you can save to the server’s hard disk)
1. Login to JasperReports Server using the `superuser` account
2. Navigate to the Export tab page: `Manage > Server Settings > Export tab`
3. Click the `Export` button to accept the default values

   Important: If you have a very large amount of repository data, it might be better to run export from the command line instead of from the UI.
4. When you are prompted to save the file, save to a location on the hard disk.

   Remember that the export.zip file will need to be accessible from the command line where you run the upgrade commands. So, if you save the zip locally you will need to upload it to the server where you are running the upgrade commands.

4.4.2 Using Buildomatic Scripts to Export Data

If you configured buildomatic and your `default_master.properties` file for export as described in the JasperReports Server Administrator Guide, you can export your repository data. For example, to export 5.5 repository data, use the following commands:

1. Navigate to the buildomatic directory:

```
cd <js-install-5.5>/buildomatic
```

2. Run buildomatic with the export target:

Windows:	`js-ant.bat export-everything -DexportFile=js-5.5-export.zip`
Linux:	`./js-ant export-everything -DexportFile=js-5.5-export.zip`

> Note the location of this export file so that you can use it during the 6.0 upgrade process.

4.4.3 Using the js-export Script to Export Data

To use the `js-export.bat/.sh` script, navigate to the buildomatic folder, for example, `<js-install-5.5>/buildomatic`. If you are using the PostgreSQL database then the js-export script should already be configured to run. If you are using a different database, or you have changed database passwords, you may need to update the `js-export` configuration.

The import-export utility for JasperServer 3.7 needs additional configuration. For complete information on the standard import-export options refer to the *JasperReports Server Administrator Guide*.

Run the following commands:

1. Navigate to the buildomatic directory:

   ```
   cd <js-install-5.5>/buildomatic
   ```

2. Run the `js-export` script:

Windows:	`js-export.bat --everything --output-zip js-5.5-export.zip`
Linux:	`js-export.sh --everything --output-zip js-5.5-export.zip`

Note the location of this export file so that you can use it during the 6.0 upgrade process.

4.5 Preparing the JasperReports Server 6.0 WAR File Distribution

Use the buildomatic `js-upgrade` scripts included in the 6.0 WAR file distribution ZIP release package to carry out the upgrade. Follow these steps to obtain and unpack the WAR file distribution ZIP file:

1. The WAR file distribution comes in a file named `jasperreports-server-6.0-bin.zip` in the compressed ZIP format. Download the WAR file distribution from Jaspersoft technical support (http://support.jaspersoft.com) or contact your sales representative.
2. Extract all files from `jasperreports-server-6.0-bin.zip`. Choose a destination, such as a `C:\Jaspersoft` folder on Windows, `/home/<user>` on Linux, or `/Users/<user>` on Mac.

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

```
<js-install-6.0>
```

4.6 Configuring Buildomatic for Your Database and Application Server

This upgrade procedure uses the `js-upgrade-newdb` shell script.

For Unix, the bash shell is required for the js-upgrade scripts. If you are installing to a non-Linux Unix platform such as HP-UX, IBM AIX, FreeBSD or Solaris, you need to download and install the bash shell. See the Troubleshooting appendix of the *JasperReports Server Installation Guide* for more information.

This section shows example configurations for the PostgreSQL, MySQL, and Oracle databases. Other databases are similar.

4.6.1 Example Buildomatic Configuration

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

4.6.1.1 PostgreSQL Example

This example shows how to configure `default_master.properties` for PostgreSQL.

1. Locate the `postgresql_master.properties` sample configuration file:

Database	Master Properties File
PostgreSQL	<js-install-6.0>/buildomatic/sample_conf/postgresql_master.properties

2. Copy the file to `<js-install-6.0>/buildomatic`
3. Rename the file `default_master.properties`
4. Edit `default_master.properties` for your database and application server:

Database	Sample Property Values

PostgreSQL	appServerType=tomcat6 [tomcat7, tomcat5, jboss, glassfish2, glassfish3]
	appServerDir=c:\\Apache Software Foundation\\Tomcat 6
	dbUsername=postgres
	dbPassword=postgres
	dbHost=localhost

4.6.1.2 MySQL Example

This example shows how to configure `default_master.properties` for MySQL.

1. Locate the `mysql_master.properties` sample configuration file:

Database	Master Properties File
MySQL	<js-install-6.0>/buildomatic/sample_conf/mysql_master.properties

2. Copy the file to `<js-install-6.0>/buildomatic`
3. Rename the file `default_master.properties`
4. Edit `default_master.properties` for your database and application server:

Database	Sample Property Values

MySQL	appServerType=tomcat6 [tomcat7, tomcat5, jboss, glassfish2, glassfish3]
	appServerDir=C:\\Apache Software Foundation\\Tomcat 6
	dbUsername=root
	dbPassword=password
	dbHost=localhost

4.6.1.3 Oracle Example

This example shows how to configure `default_master.properties` for Oracle.

1. Locate the `oracle_master.properties` sample configuration file:

Database	Master Properties File
Oracle	<js-install-6.0>/buildomatic/sample_conf/oracle_master.properties

2. Copy the file to `<js-install-6.0>/buildomatic`

3. Rename the file to `default_master.properties`
4. Edit `default_master.properties` for your database and application server:

Database	Sample Property Values

Oracle	`appServerType=tomcat6 [tomcat7, tomcat6, jboss, glassfish2, glassfish3]`
	`appServerDir=c:\\Apache Software Foundation\\Tomcat-6 (for example)`
	`dbUsername=jasperserver`
	`dbPassword=password`
	`sysUsername=system`
	`sysPassword=password`
	`dbHost=localhost`

5. As of 5.6.1, JasperReports Server includes the TIBCO JDBC drivers for the following commercial databases: Oracle, SQL Server, or DB2. If you want to use a different JDBC driver, you need to copy it to the correct location and edit default_master.properties before running the upgrade steps.

   With Oracle, for example, the alternate JDBC jar should be copied into the following location:

   ```
   <js-install-6.0>/buildomatic/conf_source/db/oracle/jdbc
   ```

   For SQL Server or DB2, the driver would get copied to the following locations respectively:

   ```
   <js-install-6.0>/buildomatic/conf_source/db/sqlserver/jdbc
   <js-install-6.0>/buildomatic/conf_source/db/db2/jdbc
   ```

   Once you have copied the driver, in your default_master.properties file, go to the Setup Standard JDBC Driver section and edit it according to the instructions in that section.

4.6.2 Additional Step when Using JBoss 7 (and Oracle, SQL Server, or DB2)

If you are using JBoss 7 as your application server and Oracle, SQL Server, or DB2 as your database, and you are using a driver other than the TIBCO JDBC driver, there is an additional set of required steps to handle the JDBC driver.

You will need to make an explicit reference to your JDBC driver file name so that JBoss 7 will know the exact file name.

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

   Edit: `<js-install-6.0>/buildomatic/default_master.properties`

   Look for the section "Setup JDBC Driver"

   Uncomment and edit the two lines shown below:

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

   So that they look like this:

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

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

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

   Edit: `<js-install-6.0>/buildomatic/install_resources/jboss/jboss-deployment-structure.xml`

   Look for the section "Setup JDBC Driver"

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

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

So that it looks like this:

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

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

4.7 Upgrading to JasperReports Server 6.0

Now that your buildomatic scripts have been configured, you can complete the upgrade.

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

Make sure you have backed up your old JasperReports Server WAR file before proceeding.

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

Commands	Description
`cd <js-install-6.0>/buildomatic`	Change to buildomatic directory
`js-upgrade-newdb.bat <path>\js-5.5-export.zip`	(Windows) Upgrade jasperserver-pro war file, drop and recreate the database, import data file from previous version.
`./js-upgrade-newdb.sh <path>/js-5.5-export.zip`	(Linux) Upgrade jasperserver-pro war file, drop and recreate the database, import data file from previous version.

On MySQL, if you receive an error about packet size, see the Troubleshooting appendix of the *JasperReports Server Installation Guide*.

If you have auditing enabled, see the section about including audit events in the Troubleshooting appendix of the *JasperReports Server Installation Guide*.

4.7.1 js-upgrade Test Mode

You can run the `js-upgrade` script in test mode using the `test` option. For example, on Window, enter:

```
cd <js-install-6.0>/buildomatic
js-upgrade-newdb.bat test <path>/js-5.5-export.zip
```

In test mode, the js-upgrade scripts check your default_master.properties settings. The application server location and the capability to connect to the specified database are validated. Using `test` mode can help debug issues such as an incorrect database password. Your system will not be altered when executing the script in `test` mode.

4.7.2 Output Log Location

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

The output log file is located here:

```
<js-install-6.0>/buildomatic/logs/js-upgrade-<date>-<number>.log
```

4.7.3 Errors

If you encounter errors during the `js-upgrade` script execution, first look at the output log to see if you can spot any errors. Also, refer to the Troubleshooting appendix of the *JasperReports Server Installation Guide*. The information in this appendix applies to `js-upgrade` scripts as well as `js-install` scripts.

If you need to modify values in your `default_master.properties` file, you can simply edit the file. When you run the `js-upgrade` script again, the new values are used.

4.8 Starting and Logging into JasperReports Server 6.0

Start your application server. Your database should already be running.

4.8.1 Clearing Your Browser Cache

Before you log in, make sure you and your end users clear the browser cache. JavaScript files, which enable the UI elements of JasperReports Server, are typically cached by the browser. Clear the cache to ensure that the newer files are used.

4.8.2 Logging into JasperReports Server

Log in using the following URL, user IDs, and passwords:

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

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

Your JasperReports Server instance has now been upgraded to 6.0. In the event of startup or login problems, refer to the Troubleshooting appendix of the *JasperReports Server Installation Guide*.

4.9 Additional Tasks to Complete the Upgrade

The tasks described below should be done when the application server is shutdown.

4.9.1 Handling JasperReports Server Customizations

If you made modifications or customizations to the original JasperReports Server application, JasperReports Server 5.5 for example, 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 your previous environment and re-integrated into the upgraded environment.

4.9.2 Clearing the Application Server Work Folder

Application servers have work folders where JasperReports Server files are compiled and cached and other objects are stored. When you update the WAR file or license, the buildomatic `deploy-webapp-pro` target should automatically clear the application server's `work` directory, but it's a good practice to double-check. A permission problem, or some other problem, could prevent the clearing of the work folder.

To clear the work folder in Tomcat:

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

4.9.3 Clearing the Application Server Temp Folder

JasperReports Server uses caching to speed operations within the application. Caching files are created and stored in the application server to support this functionality. Typically, these cached files are stored in a `temp` folder. Clear this `temp` folder to avoid any conflicts after the upgrade is complete. Typically, the `temp` folder used by an application server corresponds to the path pointed at by the `java.io.tmpdir` Java system property. For Apache Tomcat the `temp` folder is `<tomcat>/temp`.

To clear the temp folder in Apache Tomcat:

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

4.9.4 Clearing the Repository Cache Database Table

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

To manually clear the repository cache database table, run a SQL command similar to one shown below:

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

4.10 Old Manual Upgrade Steps

This section has the older, manual upgrade steps that were in place before the `js-upgrade` shell scripts were implemented in the 4.0 release. These are provided in the following table as a reference, mainly for internal use. The `js-upgrade` shell scripts execute these buildomatic targets "behind the scenes." Jaspersoft recommends using the `js-upgrade` scripts described in the beginning of this upgrade chapter instead of these manual steps.

Older buildomatic upgrade steps for this chapter are the following (using a 5.5 upgrade as an example):

Commands	Description
`cd <js-install-6.0>/buildomatic`	

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

`js-ant import-minimal-pro`	

`js-ant import-upgrade`
`-DimportFile="<path-and-filename>"`
	The -DimportFile should point to the <path> and <filename> of the js-5.5-export.zip file you created earlier.
	On Windows, you must use double quotation marks (") if your path or filename contains spaces. On Linux, you must use double quotation marks, escaped with a backslash (\") in this case.
	Note: "import-upgrade" will import resources from the 5.5 instance in a "non-update" mode (so that core resources from 6.0 will stay unchanged). Additionally, the "update-core-users" option will be applied so that the superuser and jasperadmin users will have the same password as set in the 5.5 instance.

`js-ant import-sample-data-upgrade-pro`	(Optional) This step is optional; it loads the new sample data. The old sample data is overwritten, so you may need to redo certain changes such as configuring the sample data sources for your database.
`js-ant deploy-webapp-pro`	Delete the existing older war file, deploy the new war file.

CHAPTER 5 UPGRADING JASPERSERVER 3.5 OR EARLIER

5.1 Upgrading from 3.5 or Earlier

If you are running JasperServer version 3.5, you must upgrade in two steps:

1. Upgrade from version 3.5 to version 3.7.
2. Upgrade from version 3.7 to version 6.0.

The steps to perform this upgrade are documented in the *JasperServer Installation Guide* for the 3.7 release. Download the JasperServer 3.7 WAR file distribution zip package to get the relevant files and documentation. The Installation Guide is in the docs folder.

Download the JasperServer 3.7 WAR file distribution zip package from Jaspersoft technical support or contact your sales representative.

If you are running a JasperServer version earlier than 3.5, first upgrade to 3.7, then to 6.0.

5.2 Best Practices for Upgrading under Windows

There are two standard procedures for installing JasperReports Server. The two installation methods are the following:

1. Installing with the Binary Installer and Bundled Components

   The binary installer is an executable which can put all of the components in place to run JasperReports Server. So, for instance, if you take the default choices during the installation, you will get the Apache Tomcat application server, the PostgreSQL database and Java execution environment.

   However, it should be kept in mind that these components are specially configured to run JasperReports Server. These components are also “hard coded” so that they apply to a specific version of JasperReports Server. This is true with the Windows Start Menu items created to start and stop JasperReports Server.

2. Installing to Pre-existing Components

   When installing a “Production” type instance of JasperReports Server, it is common to pre-install the main components before installing JasperReports Server. This is because the System Administrator will have more control over updating and upgrading these components such as Apache Tomcat (or any other certified application server), PostgreSQL (or any other certified database) and Java.

Once the Administrator puts these pre-existing components in place, there are two ways to install JasperReports Server:

a. Using the War File ZIP distribution (file name: jasperreports-server-bin-<ver>.zip)

   JasperReports Server will be installed to the existing components using the `js-install.bat` scripts. The Administrator will create a default_master.properties file that will specify where to find the application server and database components.

b. Using the Binary Installer (file name: jasperreports-server-<ver>-windows-<x86/64>-installer.exe)

   The installer will prompt the Administrator to specify where to find the application server and database components.

If you are installing JasperReports Server under the Windows operating system and you intend to have this be a long running instance that will be upgraded with future releases then it is recommended that you install to pre-existing components. This will reduce any confusion that might be caused after an upgrade is completed by having Windows Start Menu items that show an older version (that is, the originally installed version number) of JasperReports Server.

Chapter 6 Upgrading from the Community Project

If you are currently running a Community Project (CP) instance of JasperReports Server and you would like to upgrade to a commercial version of JasperReports Server, follow the instructions in this chapter.

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

> This CP to commercial upgrade procedure is only valid for upgrade within a major JasperReports Server release, for example 6.0 CP to 6.0 commercial.

This chapter contains the following sections:

- **General Procedure**
- **Backing Up Your JasperReports Server CP Instance**
- **Exporting Your CP Repository Data**
- **Preparing the JasperReports Server 6.0 WAR File Distribution**
- **Configuring Buildomatic for Your Database and Application Server**
- **Upgrading to the Commercial Version of JasperReports Server 6.0**
- **Starting and Logging into JasperReports Server 6.0**
- **Re-Configuring XML/A Connections (Optional)**

6.1 General Procedure

The upgrade procedure consists of the following main steps:

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

If you customized or extended JasperReports Server CP, you need to keep track of these modifications and re-integrate them into your JasperReports Server commercial instance after completing the upgrade.

6.2 Backing Up Your JasperReports Server CP Instance

First, backup the old JasperReports Server CP WAR file and `jasperserver` database in the event of a problem with the upgrade. These steps are performed from the command line in a Windows or Linux shell.

These instructions assume you have Tomcat application server and the PostgreSQL or MySQL database. You perform a similar procedure if you have another application server. If you have another database, consult your DB administration documentation for back up information.

6.2.1 Backing Up Your JasperReports Server CP WAR File

For example, for Apache Tomcat, back up the `jasperserver` directory from the `<tomcat>/webapps` folder:

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

6.2.2 Backing Up Your JasperReports Server Database

Go to the location where you originally unpacked your CP WAR file distribution zip. (Or create a new local folder to hold your backup file.)

1. Go to the `<js-install-cp>` directory.
2. Run one of the following commands:
   - For PostgreSQL on Windows or Linux:
     ```
     cd <js-install-cp>
     pg_dump --username=postgres  jasperserver  >  js-db-cp-dump.sql
     ```
   - For MySQL on Windows:
     ```
     mysqldump --user=root --password=<password> jasperserver > js-db-cp-dump.sql
     ```
     For MySQL on Linux:
     ```
     mysqldump --user=root --password=<password> --host=127.0.0.1 jasperserver >js-
     db-cp-dump.sql
     ```

> For MySQL, if you receive an error about packet size, see the Troubleshooting appendix of the *JasperReports Server Installation Guide*.

6.3 Exporting Your CP Repository Data

Before exporting your CP repository data, check to see that you have the `default_master.properties` file in this directory.

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

If you do not have a default_master.properties file, see **6.5.1, "Example Buildomatic Configuration ," on page 39**. This file holds the settings that are specific to your JasperReports Server instance, such as your application server location and your database type and location.

To export your CP repository data:

1. Navigate to the buildomatic directory:

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

2. Run buildomatic with the export target:

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

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

Remember the path to your exported file. You need to specify it when you import into your commercial JasperReports Server repository.

6.4 Preparing the JasperReports Server 6.0 WAR File Distribution

Use the buildomatic scripts included in the commercial 6.0 WAR file distribution release package to carry out the upgrade. Follow these steps to obtain and unpack the commercial 6.0 WAR file distribution ZIP file:

1. The WAR file distribution comes in a file named `jasperreports-server-6.0-bin.zip` in the compressed ZIP format. Download the WAR file distribution from Jaspersoft technical support or contact your sales representative.
2. Extract all files from `jasperreports-server-6.0-bin.zip`. Choose a destination, such as `C:\Jaspersoft` on Windows, `/home/<user>` on Linux, or `/Applications` on Mac OSX.

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

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

6.5 Configuring Buildomatic for Your Database and Application Server

This upgrade procedure uses the buildomatic scripts included with the WAR File Distribution ZIP release package to carry out the upgrade.

6.5.1 Example Buildomatic Configuration

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

6.5.1.1 PostgreSQL Example

This example uses PostgreSQL (the same general logic applies to other databases).

1. Copy `postgresql_master.properties` from:

   ```
   <js-install-pro>/buildomatic/sample_conf
   ```

2. Paste the file to:

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

3. Rename the file to: `default_master.properties`
4. Edit default_master.properties for your database and application server. Sample property values are:

   ```
   appServerType=tomcat6 (or tomcat7, tomcat5, jboss, glassfish)
   ```

```
appServerDir=c:\\Apache Software Foundation\\tomcat-6.0.26 (for example)
dbUsername=postgres
dbPassword=postgres
dbHost=localhost
```

6.5.1.2 MySQL Example

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

1. Copy `mysql_master.properties` from:

   ```
   <js-install-pro>/buildomatic/sample_conf
   ```

2. Paste the file to:

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

3. Rename the file to: `default_master.properties`
4. Edit default_master.properties for your database and application server. Sample property values are:

   ```
   appServerType=tomcat6 (or tomcat7, tomcat5, jboss, glassfish)
   appServerDir=c:\\Apache Software Foundation\\tomcat-6.0.26 (for example)
   dbUsername=root
   dbPassword=password
   dbHost=localhost
   ```

6.6 Upgrading to the Commercial Version of JasperReports Server 6.0

After configuring the default_master.properties file, you can complete the upgrade.

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

Make sure you have backed up your old JasperReports Server WAR file before proceeding.

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

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

`js-ant drop-js-db`
`js-ant create-js-db`
`js-ant init-js-db-pro`
	The first command will delete your jasperserver db. Make sure it is backed up. The other commands will recreate and initialize the database.

`js-ant import-minimal-pro`	Adds superuser, Themes, and default tenant structure.

Commands	Description
`js-ant import-upgrade -DimportFile=<path>/js-cp-export.zip -DimportArgs="--include-server-settings"`	The -DimportFile argument should point to the js-cp-export.zip file you created earlier. On Windows, you must use double quotation marks (“) if your path or filename contains spaces. On Linux, you must use double quotation marks escaped with a backslash (\”) in this case.
`js-ant import-sample-data-upgrade-pro`	(Optional) This step is optional. Loads the 6.0 commercial sample data.
`js-ant deploy-webapp-cp-to-pro`	Delete the CP war file, and deploy the commercial (pro) war file.

On MySQL, if you receive an error about packet size, see the Troubleshooting appendix of the *JasperReports Server Installation Guide*..

6.7 Starting and Logging into JasperReports Server 6.0

Before starting the server:

1. Set up the JasperReports Server License.

   Copy the `<js-install-pro>/jasperserver.license` file to the `C:\Users\<user>` directory (Windows 7 example)

   For information about how to set up the license, see the *JasperReports Server Installation Guide*.
2. Delete any files that might exist in the `<tomcat>\temp` folder.
3. Delete any files, directories, or sub-directories that exist in the folder: `<tomcat>\work\Catalina\localhost`.
4. Delete any jasperserver*.xml files that might exist in the folder: `<tomcat>\conf\Catalina\localhost`.
5. Move any existing `<tomcat-install>\logs` files into a backup directory to clean up old CP log data. (optional).

   For information about how to clear directories, see sections **4.9.2, “Clearing the Application Server Work Folder,” on page 33**, **4.9.3, “Clearing the Application Server Temp Folder,” on page 33**, and **4.9.4, “Clearing the Repository Cache Database Table,” on page 33**.

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

6.7.1 Clearing Your Browser Cache

Before you log in, make sure you and your end-users clear the Browser cache. JavaScript files, which enable UI elements of JasperReports Server, are typically cached by the browser. Clear the cache to ensure that the newer files are used.

6.7.2 Logging into the Commercial Version of JasperReports Server 6.0

Log in using the following URL, user IDs, and passwords:

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

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

Your jasperadmin password might be reset to the default setting by the upgrade operation. For example, the jasperadmin password might be reset to `jasperadmin`. For security reasons, you should change your jasperadmin and superuser passwords to non-default values.

Your JasperReports Server instance has now been upgraded from Community Project (CP) to commercial. In the event of startup or login problems, refer to the Troubleshooting appendix of the *JasperReports Server Installation Guide*.

6.8 Re-Configuring XML/A Connections (Optional)

XML/A connection definitions contain a username and password definition in order to make the Web Services connection to the server. A commercial edition of JasperReports Server supports multi-tenancy, which allows multiple organizations to co-existing on a single instance. JasperReports Server users must belong to a specific organization (except for superuser). The default organization in JasperReports Server is `organization_1`.

After upgrading from the CP to commercial version of JasperReports Server, users belong to the default organization that is part of the core data setup in the commercial version. You need to update any XML/A connection definitions to include the organization that the user belongs to.

In addition, the XML/A connection specifies an instance URI. This URI will need to be updated to point to a commercial URI.

If you are using XML/A connections, edit your XML/A connections as shown in the following examples:

- User IDs
  Change “`jasperadmin`” to “`jasperadmin|organization_1`”
  Change “`joeuser`” to “`joeuser|organization_1`”
- URI values
  Change:
  ```
  http://localhost:8080/jasperserver/xmla
  ```
  to
  ```
  http://localhost:8080/jasperserver-pro/xmla
  ```

Appendix A Planning Your Upgrade

Some of the new and enhanced features in JasperReports Server 4.7, 5.0, and 5.6 can affect your deployment and you should plan your upgrade accordingly. Prior to upgrading to you should make sure to:

- Review this information carefully and determine how the changes described affect your deployment.
- Make sure to back up your current JasperReports Server installation and repository before upgrading.

The versions and their affected functionality are as follows:

- Changes in 5.6.1 affect commercial JDBC drivers.
- Changes in 5.6 affect commercial JDBC drivers as well as XML/A connections.
- Changes in 5.0 affect XML/A connections and deployments with custom settings.
- Changes in 4.7 affect deployments with Ad Hoc reports and/or theme customizations. In addition, if you enable the data snapshot functionality added in 4.7, size requirements for your repository may increase.

Changes are cumulative, so review all topics that affect you. For example, if you are upgrading from 4.7 to 6.0, you may be affected by changes in 5.0 and 5.6.

This section describes only those changes that can significantly impact your existing deployment. For an overview of new features, improvements, and bug fixes see the release notes in the root directory of the distribution. For information on how to use the new features, see the *JasperReports Server User Guide* or the *JasperReports Server Administrator Guide*.

This chapter contains the following sections:

A.1 Changes in 5.6.1 That May Affect Your Upgrade

In 5.6.1, Jaspersoft added TIBCO JDBC drivers for the following commercial databases:

- Oracle
- SQL Server
- DB2
- Hive — available as a data source only; cannot be used for the JasperReports Server repository database.

The new drivers have the following impact:

- If you are upgrading and want to use your current driver for the repository, you must copy the driver to the correct location and modify default_master.properties for the driver you choose. See the section on your upgrade procedure for more information.
- If you are using WebLogic or WebSphere and want to use a TIBCO driver, you need to set up the driver in your application server console. See the *JasperReports Server Installation Guide* or the *JasperReports Server Administrator Guide* for more information.
- For supported Hive platforms other than Impala and Cloudera 5, you must use the TIBCO JDBC driver.

A.2 Changes in 5.6 That May Affect Your Upgrade

The following changes in 5.6 and newer can significantly affect your deployment:

- Removal of commercial JDBC drivers: if you are using a commercial JDBC driver, you will need to copy it to the correct location in your upgraded JasperReports Server.
- Changes to OLAP engine: Due to change between version of the OLAP engine, if you use Jaspersoft OLAP's XML/A functionality to connect to a remote JasperReports Server's XML/A sources, you must take additional steps to complete your upgrade to 5.6.

A.2.1 Removal of Commercial JDBC Drivers

The following commercial database drivers were removed from the JasperReports Server 5.6 package: Oracle, SQL Server, or DB2. You will need to obtain a JDBC driver before running the upgrade steps. In this case, for Oracle, since you most likely have an Oracle JDBC driver in your existing JasperReports Server instance, it can be simplest to copy it from that location:

Copy from: `<js-install-existing>/buildomatic/conf_source/db/oracle/jdbc/<driver-name>.jar`

Copy to: `<js-install-5.6>/buildomatic/conf_source/db/oracle/jdbc`

A.2.2 Changes to OLAP Engine

If you use Jaspersoft OLAP's XML/A functionality to connect to a remote JasperReports Server's XML/A sources, you must take additional steps to complete your upgrade to 5.6. This is due to a change between versions of the OLAP engine.

Once the new version of JasperReports Server is installed and running, locate all the XML/A connections that point to a remote JasperReports Server instance. Then, edit the DataSource field to specify `JRS` as the DataSource portion of its value.

For example, in previous versions, the Foodmart XML/A connection specified:

```
Provider=Mondrian;DataSource=Foodmart;
```

During upgrade, this connection must be changed to:

```
Provider=Mondrian;DataSource=JR
```

Note that for 5.6, the trailing semicolon should be removed (the older 5.5 syntax includes a semicolon at the end).

For more information about creating and editing XML/A connections, refer to the *Jaspersoft OLAP User Guide*.

Note that if your instance hosts multiple organizations, XML/A connections will fail for `superuser`. This limitation applies exclusively to the `superuser` account. To test connections in servers that host multiple organizations, Jaspersoft recommends that you log in as `jasperadmin` or another administrative account that is not `superuser`.

One reason you might have XML/A connections to remote instances of JasperReports Server is to create a load-balanced Jaspersoft OLAP environment. For more information, refer to the *Jaspersoft OLAP Ultimate Guide*.

A.3 Changes in 5.0 That May Affect Your Upgrade

The following changes in 5.0 and newer can significantly affect your deployment:

- **Upgrading preserving global properties:** If you perform an upgrade using the WAR file scripts, you might be able to preserve any custom global settings you have set. See **A.3.1, "Upgrading Preserving Custom Settings," on page 45** for more information about this feature.
- **XML/A Connections to Servers with Multiple Organizations:** If you use Jaspersoft OLAP as an XML/A provider for remote clients, and the server hosts cubes for multiple organizations, you must take additional steps during upgrade. See **A.3.2, "XML/A Connections to Servers with Multiple Organizations," on page 45** for more information about this procedure.

A.3.1 Upgrading Preserving Custom Settings

If you perform an upgrade using the WAR file scripts, there is no specific action you need to take. However, you need to be aware that the `js-upgrade-samedb` script will automatically preserve any custom global properties you have set, whereas the `js-upgrade-newdb` script will not preserve your custom global properties.

A.3.2 XML/A Connections to Servers with Multiple Organizations

Unless your JasperReports Server instance includes both Jaspersoft OLAP XML/A connections and multiple organizations, you can skip this section.

If you use Jaspersoft OLAP as an XML/A provider for remote clients, and the server hosts cubes for multiple organizations, you must take additional steps during upgrade.

In previous versions, an XML/A connection pointing to another JasperReports Server that hosted multiple organizations could pass an organization's ID to the remote server as an argument in the **Data Source** field. In this release, this configuration is no longer supported. If your XML/A connections pass organization IDs using the `tenantID` argument in the **Data Source** field, you must edit the XML/A connections and remove the `tenantID` argument. You must still pass a user's organization along with the user name and password when connecting in this configuration; the user's organization determines the cube that is queried.

Note that, if you connect to a specific organization's XML/A source in a remote server as superuser, you must still pass the `tenantID` argument in the XML/A connection's **Data Source** field. However, Jaspersoft recommends that you instead connect as jasperadmin or as a non-administrative user of the specific organization. For more information, refer to the *Jaspersoft OLAP User Guide*.

A.4 Changes in 4.7 That May Affect Your Upgrade

The following changes in 4.7 can affect your deployment:

- **Ad Hoc Reports:** The workflow of the Ad Hoc Editor has been modified to distinguish between Ad Hoc views, which allow you to interact with the data and perform data analysis, and reports generated from Ad Hoc views, which allow you to view the data and generate reports. In addition, you can create multiple reports from a single Ad Hoc view; you can then modify the Ad Hoc view and see your changes reflected in dependent reports. This upgrade makes fundamental changes to the architecture of Ad Hoc objects, and migrates all existing Ad Hoc reports to a new Ad Hoc view format. You may encounter the following when you migrate an Ad Hoc report to an Ad Hoc view:
  - (*Affects all deployments with Ad Hoc reports*) To ensure that your Ad Hoc reports continue to appear in any dashboards where they are used, the Ad Hoc migration process creates two objects for each Ad Hoc report: an *Ad Hoc view* that contains your data analysis and a *report* that refers to the data and contains any additional formatting that has been applied using iReport or Jaspersoft Studio. This means that the upgrade will double the number of Ad Hoc-related objects in your JasperReports repository. For example, if you have 10 Ad Hoc reports prior to the 4.7 upgrade, after migration you will have 20 objects — 10 Ad Hoc views and 10 related reports. Unwanted reports can be manually deleted unless they are used in dashboards.
  - (*Affects tabular Ad Hoc Reports*) The widths of columns in a tabular report are not perservered. Use the interactive report viewer to reset the column widths and save the report again.
  - (*Affects Ad Hoc reports modified in iReport or Jaspersoft Studio*) Modifications made to an Ad Hoc report in iReport or Jaspersoft Studio are not persevered. The JRXML of the original Ad Hoc report is attached to the converted report for reference. Reports created directly in iReport or Jaspersoft Studio are not affected.
  - (*Affects OLAP Ad Hoc reports against XML/A clients)* Ad Hoc OLAP reports against XML/A clients must be migrated manually using a menu selection.
  - *(Affects some Ad Hoc reports with customizations*) Some reports with customizations may not migrate.

  See **“Migrating Ad Hoc Reports to Ad Hoc Views” on page 46** for more information.
- **Themes:** An upgraded user interface eliminates unnecessary white space and lets users see more on a single screen. There have been changes to the resources and CSS used by themes, which make it easier to build and maintain new themes. However, existing themes will have to be redesigned in order to work in 4.7. See **“Upgrading Themes in 4.7” on page 50** for details on how to upgrade themes.
- **Data Snapshots:** Reports stored in JasperReports Server can now be configured to store snapshots of the report data in the repository resource. Data snapshots load quickly without querying the data source, thus increasing performance and reducing database load. If data snapshots are turned on, size requirements for your repository may increase dramatically. Data snapshots are disabled globally by default. See **“Migrating Ad Hoc Reports to Ad Hoc Views” on page 46** for more information on data snapshots.

A.4.1 Migrating Ad Hoc Reports to Ad Hoc Views

The workflow of the Ad Hoc Editor has been modified to distinguish between Ad Hoc views and the reports generated from them. Jaspersoft has deprecated the ambiguous Ad Hoc report resource in favor of an Ad Hoc view resource equivalent to an interactive data source. Ad Hoc views may only be opened in the Ad Hoc Editor, and reports created from Ad Hoc views open in the interactive report viewer.

You may see the following problems when you upgrade Ad Hoc reports:

- Ad Hoc reports migrate, but have formatting problems.
- Changes made to an Ad Hoc Report in iReport or Jaspersoft Studio are not persevered.
- Ad Hoc OLAP reports against XML/A clients must be migrated manually using a menu selection.

- Migration fails for some Ad Hoc reports. You can locate problematic reports in the repository using a simple search.

A.4.1.1 Changes to Ad Hoc

Table A-1 summarizes the differences between Ad Hoc reports and Ad Hoc views.

Table A-1 Comparison of Ad Hoc Reports and Ad Hoc Views

Ad Hoc 4.5.1 and earlier	Ad Hoc in 4.7
Ad Hoc report can be opened in interactive viewer.	Ad Hoc view can be opened in display mode in Ad Hoc Editor.
Ad Hoc report can be opened in interactive viewer, iReport, and Jaspersoft Studio.	Reports created from Ad Hoc views can be opened in interactive viewer, iReport, and Jaspersoft Studio.
In order to Export the results, user must open report in interactive viewer, iReport, or Jaspersoft Studio.	User can Export results directly from Ad Hoc Editor.
Ad Hoc reports can be directly included in dashboards.	Reports created from Ad Hoc views can be included in dashboards.
Results of Ad Hoc analysis tied to a single report.	Multiple reports can be created from a single Ad Hoc view.
Ad Hoc report can be directly modified in iReport or Jaspersoft Studio.	Reports created from an Ad Hoc view can be modified in iReport or Jaspersoft Studio. A single view can be presented in multiple ways. For example, a chart in a dashboard and a detailed PDF report can be created from the same Ad Hoc view.
Modifications made in iReport or Jaspersoft Studio are lost when Ad Hoc report is reopened in Ad Hoc Editor.	When an Ad Hoc view is updated in Ad Hoc Editor, changes are reflected in reports created from that view. However, not all changes can be supported in all reports. A warning is displayed whenever changes are made to an Ad Hoc view with dependent reports.
	Table components in reports created from Ad Hoc views are interactive. Future enhancements will extend Ad Hoc interactivity.
Ad Hoc reports prior to 4.5.0 could apply Ad Hoc styles.	As of 4.5.0, Ad Hoc styles are no longer supported.
Incompatible with future versions of JasperReports Server.	Updated architecture supports future feature enhancements to Ad Hoc.

A.4.1.2 Output of Migration Process

When you upgrade, each Ad Hoc report from 4.5.1 or earlier is transformed to a pair of objects as follows:

- Ad Hoc view: An Ad Hoc view since 6.0 and newer provides functionality corresponding to an Ad Hoc report opened in Design Mode in earlier versions. You can think of an Ad Hoc view as a window into a data set, along with tools that allow you to perform data analysis, such as determining the main layout and selecting fields and filters. The name of the Ad Hoc view is the name of the Ad Hoc report with AdHocDataView appended:
  - filename: <filename>_AdhocDataView — for example, Western_Region_Sales_AdhocDataView
  - label: <report_name> Ad Hoc view — for example, Western Region Sales Ad Hoc View
  - description: Ad Hoc view created from the <report_name> report — for example, Ad Hoc view created from the Western Region Sales report
- Report: Contains any modifications made in the interactive report viewer. The path, name in the repository, and description of the upgraded report are identical to those of the Ad Hoc report prior to upgrade. This ensures the report still appears wherever it is used, such as dashboards, job schedules, or reference links from JasperReports Server or an external system. If you do not need this report, it can be manually deleted from the repository.
  - filename: <filename> — for example, Western_Region_Sales
  - label: <report_name> — for example, Western Region Sales
  - description: <description> — for example, This report shows sales numbers for the Western Region grouped by product class, store location and month

A.4.1.3 Ad Hoc Migration Process

The JasperReports Server upgrade includes a migration script that performs the conversion from Ad Hoc reports to Ad Hoc views and their associated reports.

Report migration information is written to the log. You can get more information about the results of migration by turning on debug mode; this will impact the performance of the upgrade process.

- If you upgrade your repository by running the upgrade scripts on the existing repository database, the script to migrate Ad Hoc reports runs when the new JasperReports Server starts for the first time. Because the conversion process is part of JasperReports Server initialization, it may take longer than usual for the JasperReports Server web application to accept logins. The time required for conversion depends on the speed of your system and the number of Ad Hoc reports that need to be converted.

  You can observe the progress of the report conversion process in the log file:

  ```
  <js-install>/buildomatic/logs/js-upgrade-<date>-<number>.log
  ```
- If you start with a clean 4.7 or newer installation and import repository data from your previous JasperReports Server installation, the script to migrate Ad Hoc reports runs as part of the import process. If some reports fail to migrate, the script also runs during server initialization as described above. Running the script a second time migrates reports that depend on environment variables or services that are not configured during import. For example, Ad Hoc reports that use a JNDI data source configured inside the application server cannot be migrated while the application server is turned off for import.

A.4.1.4 Issues in Migrated Reports

Some Ad Hoc reports may migrate automatically but exhibit formatting issues or other problems. These include:

- Ad Hoc styles: Ad Hoc styles applied in JasperReports Server 4.5.0 or earlier are not perservered.
- Table columns: The widths of columns in an Ad Hoc table report are not perservered in the associated report. Use the interactive viewer to reset the column widths and save the report again.
- Modifications in iReport or Jaspersoft Studio: Ad Hoc reports that have been modified in iReport or Jaspersoft Studio should be accessible in 4.7 and newer but will lose the modifications. Examples of such reports include Ad Hoc reports that have been edited to resize a column, apply a style, or add a logo, as

well as Ad Hoc reports that have had custom filters, custom multi-level sorting, fusion charts, or input controls added in iReport or Jaspersoft Studio.

A.4.1.4.1 Working with Reports Modified in iReport or Jaspersoft Studio

During migration, the pre-4.7 JRXML is attached to the new upgraded report as a local resource. This JRXML, named oldAdhocJRXML, can be opened from iReport or Jaspersoft Studio for reference and compared with the new report. This can help you see report content that did not migrate, and help you understand how best to manually correct the automatically-generated report. Note that the attached JRXML is not used during upgrade in any way.

To better understand the changes that you made in iReport or Jaspersoft Studio, you can work with the Ad Hoc report in your current (pre-upgrade) version of JasperReports Server. You can save a copy of the Ad Hoc report, open this copy in the Ad Hoc editor, and then save it. This copy should contain information similar to the Ad Hoc view you get during migration. You can then compare the JRXML of the copy with the JRXML of the modified report.

A.4.1.5 Unsuccessful Migration

Ad Hoc reports that did not migrate successfully show up in the repository with the old **Ad Hoc Report** type. You can locate these reports using the log or via a special search in the repository. Reports which may not migrate automatically include:

- Ad Hoc OLAP reports against XML/A clients must be manually migrated after the server has been upgraded and has been successfully restarted. This does not affect reports against Mondrian clients.
- Ad Hoc reports that did not work correctly in the existing version of JasperReports Server will not migrate. To mitigate this, you need to modify the report in your current version and ensure that it is working correctly prior to migrating it.
- Ad Hoc reports with a custom data source or other custom components may not migrate successfully.

A.4.1.5.1 Finding Ad Hoc Reports Using the Log

The auto-upgrade script creates an output log that captures standard output and error output. The output log file is located here:

```
<js-install>/buildomatic/logs/js-upgrade-<date>-<number>.log
```

The migration process writes the following information to the log:

- If migration is successful for all Ad Hoc reports, you will see a message such as the following:

  Upgrade process for AdhocReportUnit objects completed successfully.
- If some reports did not migrate successfully, you will see an error such as the following:

  Upgrade process for AdhocReportUnit objects completed. There are Y objects of X for which there were some errors during the conversion. Switch on debug for more information.

  A stack trace is generated for each report that cannot be migrated:

  Error converting AdhocReportUnit (URL=PATH_TO_ADHOC_REPORT_UNIT) to AdhocDataView + ReportUnit
- For Ad Hoc OLAP reports against XML/A clients, conversion is skipped. You will see a message such as the following:

  Current AdhocReportUnit cannot be converted during upgrade process because it has XML/A data source. Conversion skipped. Refer to the documentation for details.

  See **"Migrating Ad Hoc OLAP Reports Against XML/A Clients" on page 50** for information on how to migrate these reports manually.

A.4.1.5.2 Finding Ad Hoc Reports in the Repository

Reports that did not migrate successfully show up in the upgraded repository with the old Ad Hoc report type. A special search allows you to locate these reports:

1. Select **View > Search Results**.
2. In the **Filters** section at the left, click **More choices...** (under **All types**), then click **Deprecated Reports**. You will see all the Ad Hoc reports that you have permission to view.

A.4.1.5.3 Working with Scheduled Reports

When migration fails for an Ad Hoc report with a scheduled run, the schedule for the report is perservered. At the scheduled runtime, JasperReports Server detects that the report is in the old Ad Hoc report format and does not attempt to run the report. Instead the scheduler writes an error to the log and also sends a notification message, which can be accessed by selecting
View > Messages from the menu.

A.4.1.5.4 Working with Reports That Fail to Migrate

When a report fails to migrate, the first thing to do is to open it in the previous version of JasperReports Server. In most cases, investigation shows that reports that fail to migrate do not work in the previous version.

Remember that reports that worked at one time can cease to work when the data source they use changes in some way. For example, if a report has not been used for some time, a schema change such as an updated column name may not be reflected in the report.

You can simplify the upgrade process and reduce the time it takes to migrate your Ad Hoc Reports by deleting unused Ad Hoc reports from your repository prior to upgrade. If auditing is enabled in your current, pre-4.7 version of JasperReports Server, you can use the audit data to locate Ad Hoc reports that have not been accessed for a specified amount of time. You can then delete the reports you no longer want. If desired, you can export each unwanted report individually prior to deletion. See the *JasperReports Server Administrator Guide* for more information about auditing and about import/export.

A.4.1.6 Migrating Ad Hoc OLAP Reports Against XML/A Clients

Ad Hoc OLAP reports can only be migrated when the server is running and able to respond to requests from the XML/A data source, which means they cannot be migrated during upgrade. They can be migrated manually as follows:

1. Locate the Ad Hoc OLAP report you want to migrate in the repository.
2. Right-click the report name and select **Migrate report**.

A.4.2 Upgrading Themes in 4.7

The look and feel of the JasperReports Server web interface has been redesigned to increase usable space and be more compatible across browsers. In addition, navigation has been improved with simplified menus and a new Library page that provides quick access to all reports, dashboards, and views a user may access. To accomplish this, images, markup, and styles have been modified and many elements and images are smaller in size. As a result of these modifications, custom themes developed for the previous interface will need to be updated for the new interface.

This section details the changes made to the user interface and describes some of the steps necessary to update custom themes. For information on developing new themes, see the *JasperReports Server Administrator Guide*.

A.4.2.1 Banner and Toolbar Modifications

Banner settings have changed in 4.7, which means you will need to modify your theme to work with the new banner. **Table A-2** shows the elements you need to modify and their default locations. The default values are for these elements are in the default.css file.

Table A-2 Banner and Toolbar Settings

Element	Classname and Modification	File	Notes
Banner	`.banner` Give custom value to `height`.	containers.css	Default value: height:26px
Banner Logo	`#logo` Give custom values to `height` and `width` that match the dimensions of your logo.	theme.css	Default values: height: 20px width: 115px
Banner Main Navigation	`.menu.primaryNav .wrap` Set `height` and `line-height` to the same measurement as `.banner.`	containers.css	Default values: height: 26px line-height: 26px

Banner Main Navigation	`.menu.primaryNav .wrap.over`
	`.menu.primaryNav .wrap.pressed`
	If you already have values for these defined in your alternate theme, then you need to change `height` and `line-height` to match the height of `.banner`.
		containers.css	Not explicitly defined, but a value of 26px for both elements is cascaded from `.menu.primaryNav .wrap` If the `over` and `pressed` effects fill the banner height after applying the modifications to `.menu.primaryNav .wrap`, then this step is not necessary.

Banner Main Navigation Home icon	`.menu.primaryNav #main_ home.wrap > .icon` Set `height` to be 2px shorter than the `height` of .banner. Set values for `width` and `background-position` to fit your image.	containers.css	Default value: background-position: left -166px. Height should be two pixels shorter than banner height. Height is not explicitly defined, but a value of 24px is cascaded from .button .icon in buttons.css Width is not explicitly defined, but a value of 14px is cascaded from `.menu.primaryNav #main_ home .wrap > .icon` in the containers.css file.

Element	Classname and Modification	File	Notes
Banner Main Navigation Item arrow icon	`.menu.primaryNav .node > .wrap> .icon` Set `height` to your desired value, with the maximum value being the same height measurement as the `.banner` element. Set `background-position` to a value that properly displays the default or your custom image.	containers.css	Default values: height: 30px; background-position: 0 -78px background-position: 0 -78px (IE8-9) background-position: 0 -79px (Ch+Saf) If you are using Chrome or Safari, the classname that you apply background position to must be preceded by "`body:nth-of-type(1)`".

Banner
Main
Navigation Item
arrow icon
	`.menu.primaryNav .wrap.over`
	`.menu.primaryNav .wrap.pressed`
	Set `background-position` to a value that properly displays the default or your custom image.
		containers.css	`background-position` is not explicitly defined. The value is cascaded from `.menu.primaryNav .node > .wrap > .icon`
			This only needs to be adjusted if you want a different color disclosure indicator for the pressed and over states of the main menu links.

Banner Metadata	`#metalinks li` Set `line-height` to the desired value that will vertically center it within the banner.	theme.css	Default value: line-height: 9px
Banner Search container	`#globalSearch.searchLockup` Set `margin-top` to desired value that will vertically center it within the banner.	controls.css	Default value: margin-top:3px
Body	`#frame` Set a custom `top` value that positions the body of the application below the banner.	containers.css	Default value: top: 28px This value needs to be greater than the value you apply to the height of `.banner`.
Toolbar	`.toolbar` Set custom height value that will fit your capsule buttons.	containers.css	Default value: height: 28px This is only necessary if you have customized capsule buttons that differ in height from the default images.

Footer	`#frameFooter`
	`Rename ID in stylesheet`
		containers.css	The #footer element was renamed to #frameFooter. Change style rules for #footer to #frameFooter.

A.4.2.2 Changes to IE Overrides

All style rules have been moved out of the IE-specific CSS files and into the base CSS files. IE-specific notations have been applied to these rules so they can be condensed into the class declarations with the other style rules for a given element. IE-specific styles are commented as shown below:

- The following comment indicates a general IE style rule:
  * IE *
- The following comment indicates a style rule specific to IE7:
  * IE7 *
- The following comment indicates a style rule specific to IE8 and IE9:
  * IE8 and IE9 *

Table A-3 shows an example of an IE7 style rule that has been combined into the base style rules in 4.7 and newer.

Table A-3 Changes to Implementation for IE Style Rules

IE overrides (4.5.1 and earlier)	lists.css (4.5.1 and earlier)	lists.css (4.7)

`.stepIndicator .icon {`
`*top: 2px;`
`}`
	`.stepIndicator .icon {`
	`margin: 0;`
	`top: 1px;`
	`margin-right: 3px;`
	`}`
		`.stepIndicator .icon {`
		`position: relative;`
		`margin: 0px;`
		`top: 1px;`
		`*top: 2px; \\* IE7 \\`
		`margin-right: 3px;`
		`}`

A.4.2.3 Images in JasperReports Server 4.7

Images for navigation, buttons, and backgrounds in the default theme have changed in version 4.7. Some images have been deleted or moved into a sprite file with a different name, other images have been modified and their properties (such as height and width) may have changed. You need to ensure your custom themes refer to these images. If you have used custom images, you may need to adjust them to work with the new default scheme. This section lists the image files and indicates the changes.

Table A-4 lists the images that have been added in the 4.7 default theme.

Table A-4 New Images in 4.7 Default Theme

Image	Usage	Notes
banner_bkgd.png	Background gradient image for application banner	
column_header_bkgd.png	Background gradient image for `.header` elements in columns and dialogs	Replaces panel_dialog_ header_sprite.png image used for dialog box headers in 4.5.1.

Image	Usage	Notes
inner_pagination_sprite.png	Pagination icons (next, previous, etc) for dashboard report widgets	
spacer.gif	Icon that appears in the column header of spacer columns in Ad Hoc Editor	
viewer_toolbar_buttons_sprite.png	Button icons for the Ad Hoc Editor toolbar	

Table A-5 lists the images that have been modified for the 6.0 and newer default theme.

Table A-5 Modified Images in 4.7 Default Theme

Image	Usage	Notes
adhoc_toolbar_buttons_sprite.png	Button icons for the Ad Hoc Editor toolbar	Some icons are new. All icons have new shadow effect.
button_action_primary_sprite.png	Background gradient images for the three states of `.action.primary` buttons	Removed rounded corners. Colors were adjusted.
button_action_sprite.png	Background gradient image for the up, over, and pressed states of action buttons	Removed rounded corners. Colors were adjusted.
button_action_square_icons_sprite.png	Button icons for the .action.square buttons	All icons have new shadow effect.
button_capsule_sprite.png	Background images with rounded corners for capsule buttons	Rounded corners have a smaller radius. Colors adjusted. Buttons are shorter in height.
button_options_sprite.png	Background images for the up, over, and pressed states of options buttons	Removed rounded corners. Added a shadow-only image for the pressed state.
disclosure_indicators_sprite.png	Small icons that disclose functionality in various parts of the application (within buttons, menus, lists, and forms)	Shadow effect added to icons for `.button.capsule.indicator`.
home_bkgd.png	Large background image on home page	Image refactored.

Image	Usage	Notes
home_icons_sprite.png	Large icons for .action.jumbo button on home and admin home pages	Icons for **View Reports** and **Create Ad Hoc View** refactored.
input_bkgd.png	Top inner shadow background image for text inputs	Shadow lightened. Image height changed from 2px tall to 3px tall.
login_welcome_bkgd.jpg	Main image on login page of pro edition	Image refactored.
login_welcome_ce_bkgd.jpg	Main image on login page of community edition	Image refactored.
logo.png	Logo used in the banner of the application	Image is smaller.
menu_primaryNav_sprite.png	Sprite containing background gradients for mouse over effects and the home icon used in the main menu	Home icons reduced in size and shadow effect added.
panel_inset_inset_bkgd.png		
search_sprite.png	Icons used in search controls (search and clear search)	Images refactored. Input background removed.
tabs_horizontal_sprite.png	Background gradient images for the over and pressed states of horizontal tabs	Removed rounded corners. Colors adjusted.
tabs_vertical_sprite.png	Background gradient images for the up, over and pressed states of vertical tabs	Removed rounded corners. Colors adjusted.

Table A-6 lists the images that have not been affected by the refactoring of themes in 4.7.

Table A-6 Images With No Modifications in 4.7

Image	Usage
adhoc_datatree_icons_sprite.png	Node and leaf images for fields, measures, and dimensions in the ad hoc data tree
floatingMenu_sprite.png	Popup menu icons for dashboard widgets
grid_20x20_bkgd.png	Grid background for dashboard designer canvas
list_node_animation.gif	Miniature “please wait” animation used in list nodes

Image	Usage
list_pressed_bkgd.png	Background image for the top inner shadow that appears for the pressed state of a .list.filter list item
lists_sprite.png	Small icons used in various lists throughout the application. Also used for tokens, menus and OLAP cell sorting
loadinfo.gif	“Please wait” animation used for iPad only
message_icons_sprite.png	Sprite containing a single icon used for alert message
panel_sizer_sprite.png	Sprite containing icons for resizing panels and dialogs
report_load_animation.gif	Animation used in the toolbar of the interactive viewer when a report is loading
sort_indicators_sprite.png	Sprite containing icons used for the up, over, and pressed states of sortable lists (ascending and descending)
tabs_horizontal_buttons_bkgd.png	Image used as the top drop shadow for `.tabSet.horizontal.buttons`
wait_animation_large.gif	“Please wait” animation used in dialog boxes on page loads and in dashboard iframes when widgets are loading

Table A-7 shows images used in the earlier themes that have been removed in 6.0, as well as images that have been refactored by placing them in the file inner_pagination_sprite.png.

Table A-7 Images from Previous Default Theme Removed in 4.7

Image	Usage	Notes
ajax-loader.gif	Animation used in the “please wait” dialog for printing dashboards that contain charts	replaced with wait_ animation_large.gif
body_bkgnd.png	Gradient background image used as the body background for all pages	
button_action_jumbo_sprite.png	Background images with rounded corners used for up, over, and pressed states of jumbo buttons	
column_bkgd_corners.png	Background image used for rounded corners with drop shadow on column decorated elements	

Image	Usage	Notes
column_bkgd_edges_rl.png	Background image used for right and left borders with drop shadow on column decorated elements	
column_bkgd_edges_tb.png	Background image used for top and bottom borders with drop shadow on column decorated elements	
first-d.gif	Disabled state of "go to first" icon used for pagination controls inside report dashboard widgets	Image placed in sprite file inner_pagination_sprite.png
first.gif	"Go to first" icon used for pagination controls inside report dashboard widgets	Image placed in sprite file inner_pagination_sprite.png
frame_background_sprite.png	Gradient background image for the `#frame` element	
frame_bkgd_corners.png	Background image used for rounded corners with drop shadow on the `#frame` element	
frame_bkgd_edges_rl.png	Drop shadow image used for right and left edges of the `#frame` element	
frame_bkgd_edges_tb.png	Drop shadow image used for top and bottom edges of the `#frame` element	
frame_header_sprite.png	Gradient background image with rounded corners used for the header of the `#frame` element.	`#frame` header element holds the main navigation.
last-d.gif	Disabled state of "go to last" icon used for pagination controls inside report dashboard widgets	Image moved to sprite file inner_pagination_ sprite.png.
last.gif	"Go to last" icon used for pagination controls inside report dashboard widgets	Image placed in sprite file inner_pagination_ sprite.png.
login_welcome_bkgd_ipad.jpg	Main image on login page for iPad	
logo_small.png	Logo used in the frame header/main navigation bar for iPad	

Image	Usage	Notes
menu_context_bkgd_corners.png	Background image used for rounded corners with drop shadow on `.menu.context` elements	
menu_context_bkgd_edges_rl.png	Background image used for right and left borders with drop shadow on `.menu.context` elements	
menu_context_bkgd_edges_tb.png	Background image used for top and bottom borders with drop shadow on `.menu.context` elements	
menu_dropDown_bkgd_corners.png	Background image used for rounded corners with drop shadow on `.menu.dropDown` elements	
menu_dropDown_bkgd_edges_rl.png	Background image used for right and left borders with drop shadow on `.menu.dropDown` elements	
menu_dropDown_bkgd_edges_tb.png	Background image used for top and bottom borders with drop shadow on `.menu.dropDown` elements	
menu_vertical_bkgd.png	Background for `.menu.vertical`	
next-d.gif	Disabled state of **next** icon used for pagination controls inside report dashboard widgets	Image placed in sprite file inner_pagination_sprite.png.
next.gif	**Next** icon used for pagination controls inside report dashboard widgets	Image placed in sprite file inner_pagination_sprite.png.
panel_bkgd_corners.png	Background image of rounded corners with drop shadow used for the `.dialog.overlay` element	
panel_bkgd_edges_rl.png	Background image of right and left borders with drop shadow used for the `.dialog.overlay` element	
panel_bkgd_edges_tb.png	Background image of top and bottom borders with drop shadow used for the `.dialog.overlay` element	

Image	Usage	Notes
panel_dialog_header_sprite.png	Background for dialog headers	
panel_info_bkgd_corners.png	Background image of rounded corners with drop shadow used for the .info element	
panel_info_bkgd_edges_rl.png	Background image of right and left borders with drop shadow used for the `.info` element	
panel_info_bkgd_edges_tb.png	Background image of top and bottom borders with drop shadow used for the `.info` element	
panel_info_system_bkgd_corners.png	Background image of rounded corners used for the `.info.system` element	
panel_info_system_bkgd_edges_rl.png	Background image of right and left borders used for the `.info.system` element	
panel_info_system_bkgd_edges_tb.png	Background image of top and bottom borders used for the `.info.system` element	
panel_info_tooltip_bkgd_corners.png	Background image of rounded corners used for the `.panel.tooltip.info` element	
panel_info_tooltip_bkgd_edges_rl.png	Background image of right and left borders used for the `.panel.tooltip.info` element	
panel_info_tooltip_bkgd_edges_tb.png	Background image of top and bottom borders used for the `.panel.tooltip.info` element	
panel_inlay_bkgd_corners.png	Background image of rounded corners with drop shadow used for the `.dialog.inlay` element	
panel_inlay_bkgd_edges_rl.png	Background image of left and right borders with drop shadow used for the `.dialog.inlay` element	

Image	Usage	Notes
panel_inlay_bkgd_edges_tb.png	Background image of top and bottom borders with drop shadow used for the `.dialog.inlay` element	
panel_inlay_gradient_left.png	Left background image with gradient and rounded corners used for the `.dialog.inlay .footer` element	
panel_inlay_gradient_right.png	Right background image with gradient and rounded corners used for the `.dialog.inlay .footer` element	
panel_inset_bkgd_corners.png	Background image of rounded corners with drop shadow used for the `.groupBox` element	
panel_inset_bkgd_edges_rl.png	Background image of left and right borders with drop shadow used for the `.groupBox` element	
panel_inset_bkgd_edges_tb.png	Background image of top and bottom borders with drop shadow used for the `.groupBox` element	
panel_widget_header_sprite.png	Background gradient with rounded corners used for the headers of the `.panel.widget` and `.dialog.overlay.widget` elements	
prev-d.gif	Disabled state of **Previous** icon used for pagination controls inside report dashboard widgets	Image placed in sprite file inner_pagination_sprite.png
prev.gif	**Previous** icon used for pagination controls inside report dashboard widgets	Image placed in sprite file inner_pagination_sprite.png
toolbar_bkgd.png	Background gradient image for the `.toolbar` element	

A.4.3 Data Snapshots

As of 4.7, reports in JasperReports Server can store snapshots of the report data as part of the report unit in the repository. Data snapshots load quickly without querying the data source, thus increasing performance and reducing database load. Data snapshots can be refreshed on-demand, by scheduling, or by setting server-wide or report-specific policies. Existing installations should also resolve any upgrade issues before enabling data snapshots.

Data snapshots are stored in the JasperReports Server repository, which may significantly increase the size of the repository. Before enabling data snapshots, you may want to analyze the effect of snapshots on your repository size and upgrade your storage. For most installations, the benefits from reducing the overall number of queries should outweigh the cost of additional storage.

The actual effect on your repository size depends on the nature of your reports and how many snapshots you enable. It is difficult to give a standard estimate, because snapshot size depends on factors such as the number of columns and the complexity of the report. For a given report, the overall snapshot size is usually proportional to the number of rows. For example, if you have a report of 100,000 records, with 20 records per page, and the snapshot size for one page is 2KB, the overall size of the snapshot will be (100,000 ÷ 20) × 2 KB, or 10 MB.

Data snapshots are turned off globally by default in 4.7 and newer. If you enable data snapshots, they can be turned on or off for each individual report.

See the *JasperReports Server Administrator Guide* for more information about enabling data snapshots in your JasperReports Server deployment. *See the JasperReports Server User Guide* for more information about using data snapshots with individual reports.

A.4.3.1 Using Data Snapshots

Enabling data snapshots has the following benefits:

- Queries are only run when a new data snapshot is enabled, when the snapshot is refreshed, and automatically when the server detects that the snapshot is out of date. This allows multiple users to look at the same report without running multiple queries.
- Sharing data snapshots means everyone sees identical data. A data snapshot is a static entity that reflects the state of the data at the time the snapshot was created. This eliminates differences sometimes experienced when data has changed in the repository, or queries are run in different time zones.
- Users can refresh a snapshot and view the latest data by clicking a **Refresh** button. Non-administrative users do not have permissions to overwrite the snapshot after refresh.
- You can turn off snapshots for reports that need to be dynamic, such as real-time reports in dashboards.