JasperReports Server Community Project Upgrade Guide

Release 5.6

http://www.jaspersoft.com

This is version 0614-JSO56-01 of the *JasperReports Server Community Project Upgrade Guide*.

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION

JasperReports Server builds on JasperReports Library as a comprehensive family of Business Intelligence (BI) products, providing robust static and interactive reporting, report server, and data analysis capabilities. These capabilities are available as either stand-alone products, or as part of an integrated end-to-end BI suite utilizing common metadata and providing shared services, such as security, a repository, and scheduling.

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

- Easily view and explore your data in the web-based drag-and-drop Ad Hoc Editor interface.
- Efficiently and securely manage many reports.
- Interact with reports, including sorting, filtering, formatting, entering parameters and drilling on data.
- Arrange reports and web content to create appealing, data-rich Jaspersoft Dashboards that quickly convey business trends.

Jaspersoft OLAP is an optional component of JasperReports Server, controlled by licence and described in its own user guide.

Jaspersoft provides several other sources of information to help extend your knowledge of JasperReports Server:

- Our Ultimate Guides document advanced features, best practices, and numerous examples. Customers can download them freely from our community website.
- Our free Business Intelligence Tutorials let you learn at your own pace, and cover topics for developers, administrators, business users, and data integrators. The tutorials are available online in the Professional Services section of our website.

Our free samples, which are installed with JasperReports Library,Jaspersoft iReport Designer, and JasperReports Server, are documented online. The samples documentation can be found on our community website.

This chapter contains the following sections:

- **JasperReports Server Upgrade Distributions**

1.1 JasperReports Server Upgrade Distributions

The following distribution package is available for JasperReports Server upgrade:

Distribution Package	Description
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-cp-5.6.0-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 Upgrading from 5.5 to 5.6

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

5.5

to JasperReports Server 5.6.

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

This chapter contains the following sections:

2.1 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 5.6 JasperReports Server WAR file distribution zip.
3. Run the js-upgrade script as described in **2.5, "Upgrading to JasperReports Server 5.6," on page 10**.

If your current instance of JasperReports Server has any custom modifications or extensions, keep track of these and re-integrate them into your 5.6 instance after upgrading.

2.2 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` war file. For example, `C:\JS_55_BACKUP` or `/opt/JS_55_BACKUP`.
2. Copy `<tomcat>/webapps/jasperserver` 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*.

2.3 Preparing the JasperReports Server 5.6 WAR File Distribution

Use the buildomatic `js-upgrade` scripts included in the 5.6 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-cp-5.6.0-bin.zip` in the compressed ZIP format. Download the WAR file distribution from http://community.jaspersoft.com. .
2. Extract all files from `jasperreports-server-cp-5.6.0-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-5.6>
```

2.4 Configuring Buildomatic for Your Database and Application Server

This upgrade procedure uses the `js-upgrade-samedb-ce` 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 and MySQL databases.

2.4.1 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`.

2.4.1.1 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-5.6>/buildomatic/sample_conf/postgresql_master.properties

2. Copy the file to `<js-install-5.6>/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`

2.4.1.2 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-5.6>/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`

2.5 Upgrading to JasperReports Server 5.6

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-5.6>/buildomatic`	
`js-upgrade-samedb-ce.bat`	(Windows) Upgrade jasperserver war file, upgrade jasperserver database to 5.6, add 5.6 repository resources into the database
`./js-upgrade-samedb-ce.sh`	(Linux) Upgrade jasperserver war file, upgrade jasperserver database to 5.6, add 5.6 repository resources into the database

2.5.1 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-5.6>/buildomatic
js-upgrade-samedb-ce.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.

2.5.2 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-5.6>/buildomatic/logs/js-upgrade-<date>-<number>.log
```

2.5.3 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.

2.6 Starting and Logging into JasperReports Server 5.6

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

2.6.1 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.

2.6.2 Logging into JasperReports Server

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

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

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

Your JasperReports Server instance has now been upgraded to 5.6. In the event of startup or login problems, refer to the Troubleshooting appendix of the *JasperReports Server Installation Guide*.

2.7 Additional Tasks to Complete the Upgrade

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

2.7.1 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, the buildomatic `deploy-webapp-ce` 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.

2.7.2 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

2.7.3 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;
```

2.8 Old Manual Upgrade Steps: 5.5 to 5.6

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-5.6>/buildomatic`	
`js-ant upgrade-5.5-5.6-ce`	Execute SQL script to upgrade database to 5.6. Executes script buildomatic/install_ resources/sql/<dbType>/upgrade-<dbType>-5.5.0-5.6.0-ce.sql

Commands	Description
`js-ant import-minimal-for-upgrade-ce`	Loads themes and other core resources for 5.6. Note: "import-minimal-for-upgrade" will import core resources in an "update" mode so that the older 5.5 core resources will be overwritten. Additionally, the "skip-user-update" option will be applied so that jasperadmin users will not have their passwords modified.
`js-ant import-sample-data-upgrade-ce`	(Optional) This step is optional. Loads the 5.6 sample data.
`js-ant deploy-webapp-ce`	Delete old 5.5 war file, deploy the 5.6 war file.

CHAPTER 3 UPGRADING FROM 3.7 - 5.2 TO 5.6

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, or 5.2

to JasperReports Server 5.6.

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 5.5 to 5.6. However, we recommend you use the procedure in **Chapter 2, "Upgrading from 5.5 to 5.6," on page 7**.

In this chapter the examples shown will use JasperReports Server 5.2 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 5.6 WAR File Distribution**
- **Configuring Buildomatic for Your Database and Application Server**
- **Upgrading to JasperReports Server 5.6**
- **Starting and Logging into JasperReports Server 5.6**
- **Additional Tasks to Complete the Upgrade**
- **Old Manual Upgrade Steps**

3.1 Upgrade Steps Overview

These are the general steps used in this section:

1. Plan your upgrade (specifically if upgrading from 4.7 or earlier).
2. Back up your current JasperReports Server instance.
3. Export your existing repository data. For example, export your 5.2 data.
4. Download and set up the new 5.6 JasperReports Server WAR file distribution zip.
5. Run the js-upgrade script as described in **3.7, "Upgrading to JasperReports Server 5.6," on page 19**.

If your current instance of JasperReports Server has any custom modifications or extensions, keep track of these and re-integrate them into your 5.6 instance after upgrading.

3.2 Planning Your Upgrade

If you are upgrading from JasperReports Server version 4.7 (or earlier), 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 25**.

3.3 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` war file. For example, `C:\JS_52_BACKUP` or `/opt/JS_52_BACKUP`.
2. Copy `<tomcat>/webapps/jasperserver` to `<path>/JS_52_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_52_BACKUP` or `/opt/JS_52_BACKUP`.
2. Run the following commands for PostgreSQL or MySQL:
   - PostgreSQL

     ```
     cd <path>/JS_52_BACKUP
     pg_dump --username=postgres  jasperserver  >  js-db-5.2-dump.sql
     ```

   - MySQL

     ```
     cd <path>/JS_52_BACKUP
     ```

Windows:

```
mysqldump --user=root --password=<password> jasperserver > js-db-5.2-dump.sql
```

Linux:

```
mysqldump --user=root --password=<password> --host=127.0.0.1 jasperserver >
js-db-5.2-dump.sql
```

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

3.4 Exporting Current Repository Data

You need to export your old repository data, for example your 5.2 repository data, using the JasperReports Server export utility. You can export in the following ways:

- Use the buildomatic scripts (if you originally installed using buildomatic).
- Use the `js-export-ce.bat/.sh` script found in the `<js-install>/buildomatic` folder.

3.4.1 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 5.2 repository data, use the following commands:

1. Navigate to the buildomatic directory:

```
cd <js-install-5.2>/buildomatic
```

2. Run buildomatic with the export target:

Windows:	`js-ant.bat export-everything-ce -DexportFile=js-5.2-export.zip`
Linux:	`./js-ant export-everything-ce -DexportFile=js-5.2-export.zip`

Note the location of this export file so that you can use it during the 5.6 upgrade process.

3.4.2 Using the js-export Script to Export Data

To use the `js-export-ce.bat/.sh` script, navigate to the buildomatic folder, for example, `<js-install-5.2>/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-5.2>/buildomatic
```

2. Run the `js-export` script:

Windows:	`js-export-ce.bat --everything --output-zip js-5.2-export.zip`
Linux:	`js-export-ce.sh --everything --output-zip js-5.2-export.zip`

Note the location of this export file so that you can use it during the 5.6 upgrade process.

3.5 Preparing the JasperReports Server 5.6 WAR File Distribution

Use the buildomatic `js-upgrade` scripts included in the 5.6 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-cp-5.6.0-bin.zip` in the compressed ZIP format. Download the WAR file distribution from http://community.jaspersoft.com.
2. Extract all files from `jasperreports-server-cp-5.6.0-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-5.6>
```

3.6 Configuring Buildomatic for Your Database and Application Server

This upgrade procedure uses the `js-upgrade-newdb-ce` 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 and MySQL databases.

3.6.1 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`.

3.6.1.1 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-5.6>/buildomatic/sample_conf/postgresql_master.properties

2. Copy the file to `<js-install-5.6>/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`

3.6.1.2 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-5.6>/buildomatic/sample_conf/mysql_master.properties

2. Copy the file to `<js-install-5.6>/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`

3.7 Upgrading to JasperReports Server 5.6

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-5.6>/buildomatic`	Change to buildomatic directory
`js-upgrade-newdb-ce.bat <path>\js-5.2-export.zip`	(Windows) Upgrade jasperserver war file, drop and recreate the database, import data file from previous version.
`./js-upgrade-newdb-ce.sh <path>/js-5.2-export.zip`	(Linux) Upgrade jasperserver 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*.

3.7.1 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-5.6>/buildomatic
```

```
js-upgrade-newdb-ce.bat test <path>/js-5.2-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.

3.7.2 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-5.6>/buildomatic/logs/js-upgrade-<date>-<number>.log
```

3.7.3 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.

3.8 Starting and Logging into JasperReports Server 5.6

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

3.8.1 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.

3.8.2 Logging into JasperReports Server

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

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

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

Your JasperReports Server instance has now been upgraded to 5.6. In the event of startup or login problems, refer to the Troubleshooting appendix of the *JasperReports Server Installation Guide*..

3.9 Additional Tasks to Complete the Upgrade

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

3.9.1 Handling JasperReports Server Customizations

If you made modifications or customizations to the original JasperReports Server application, JasperReports Server 5.2 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.

3.9.2 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, the buildomatic `deploy-webapp-ce` 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.

3.9.3 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

3.9.4 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;
```

3.10 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 5.2 upgrade as an example):

Commands	Description
`cd <js-install-5.6>/buildomatic`	

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

`js-ant import-minimal-ce`	

`js-ant import-upgrade`
`-DimportFile="<path-and-filename>"`
	The -DimportFile should point to the <path> and <filename> of the js-5.2-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 5.2 instance in a "non-update" mode (so that core resources from 5.6 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 5.2 instance.

`js-ant import-sample-data-upgrade-ce`	(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-ce`	Delete the existing older war file, deploy the new war file.

CHAPTER 4 UPGRADING JASPERSERVER 3.5 OR EARLIER

4.1 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 5.6.

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.

If you are running a JasperServer version earlier than 3.5, first upgrade to 3.7, then to 5.6.

4.2 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-cp-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-cp-<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.

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 affect XML/A connections.
- Changes in 5.0 affect XML/A connections and deployments with custom settings.
- Changes in 4.7 affect deployments with 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 5.6, you may be affected by changes in 5.6 and 5.0.

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 That May Affect Your Upgrade

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

- 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.1.1 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=JRS
```

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

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.2 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.2.1, "Upgrading Preserving Custom Settings," on page 26** for more information about this feature.

A.2.1 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-ce` script will automatically preserve any custom global properties you have set, whereas the `js-upgrade-newdb-ce` script will not preserve your custom global properties.

A.3 Changes in 4.7 That May Affect Your Upgrade

The following changes in 4.7 can affect your deployment:

- **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 26** 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 **"Planning Your Upgrade " on page 25** for more information on data snapshots.

A.3.1 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.3.1.1 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-1** 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-1 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.3.1.2 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-2 shows an example of an IE7 style rule that has been combined into the base style rules in 4.7 and newer.

Table A-2 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.3.1.3 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-3 lists the images that have been added in the 4.7 default theme.

Table A-3 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-4 lists the images that have been modified for the 5.6 and newer default theme.

Table A-4 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.
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.

Image	Usage	Notes
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-5 lists the images that have not been affected by the refactoring of themes in 4.7.

Table A-5 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
list_pressed_bkgd.png	Background image for the top inner shadow that appears for the pressed state of a .list.filter list item

Image	Usage
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-6 shows images used in the earlier themes that have been removed in 5.6, as well as images that have been refactored by placing them in the file inner_pagination_sprite.png.

Table A-6 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	
column_bkgd_edges_rl.png	Background image used for right and left borders with drop shadow on column decorated elements	

Image	Usage	Notes
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	
menu_context_bkgd_corners.png	Background image used for rounded corners with drop shadow on `.menu.context` elements	

Image	Usage	Notes
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	
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	

Image	Usage	Notes
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	
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	

Image	Usage	Notes
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.3.2 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.3.2.1 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.