



# **JASPERREPORTS SERVER INSTALLATION GUIDE**

RELEASE 5.5

<http://www.jaspersoft.com>

---

Copyright © 2013 JasperSoft Corporation. All rights reserved. Printed in the U.S.A. JasperSoft, the JasperSoft logo, JasperSoft iReport Designer, JasperReports Library, JasperReports Server, JasperSoft OLAP, and JasperSoft ETL are trademarks and/or registered trademarks of JasperSoft Corporation in the United States and in jurisdictions throughout the world. All other company and product names are or may be trade names or trademarks of their respective owners.

This is version 1013-JSP55-33 of the *JasperReports Server Installation Guide*.

## TABLE OF CONTENTS

|  |           |
|--|-----------|
| <b>Chapter 1 Introduction</b>                              | <b>11</b> |
| 1.1 Conventions  | 12        |
| 1.2 Java Version Supported                                 | 12        |
| 1.3 JasperReports Server Distributions                     | 12        |
| 1.3.1 Installer Support                                    | 12        |
| 1.3.2 WAR File Binary Distribution Support                 | 14        |
| 1.4 Release Notes  | 15        |
| 1.5 System Requirements                                    | 15        |
| 1.6 Support for Internationalization                       | 15        |
| <b>Chapter 2 Installing JasperReports Server</b>           | <b>17</b> |
| 2.1 Pre-Installation Steps                                 | 17        |
| 2.2 Starting the Installer                                 | 17        |
| 2.3 Accepting the License Agreement                        | 18        |
| 2.4 Choosing Installation Type                             | 18        |
| 2.5 Choosing an Installation Directory                     | 19        |
| 2.6 Selecting a Tomcat Configuration                       | 19        |
| 2.7 Selecting a PostgreSQL Configuration                   | 19        |
| 2.7.1 Choosing the Bundled PostgreSQL                      | 19        |
| 2.7.2 Choosing an Existing PostgreSQL on a Local Host      | 20        |
| 2.7.3 Using an Existing PostgreSQL on a Remote Host        | 20        |
| 2.7.4 Enabling Connections to a Remote Host                | 21        |
| 2.8 Installing Sample Data                                 | 21        |
| 2.9 Completing the Installation                            | 22        |
| 2.10 Post-Installation Steps                               | 22        |
| 2.10.1 Updates Made by the Installer During Installation   | 22        |
| 2.10.2 Installer Output Log File Location                  | 23        |
| 2.10.3 Installing a New License File                       | 23        |
| 2.10.4 License File for Existing Tomcat as Windows Service | 24        |
| 2.10.5 Checking your Java JVM Options                      | 24        |

|                  |  |           |
|------------------|--|-----------|
| <b>Chapter 3</b> | <b>Starting and Stopping JasperReports Server</b>              | <b>25</b> |
| 3.1              | Start/Stop Menu — Windows                                      | 25        |
| 3.1.1            | Start/Stop Menus — Bundled Tomcat and PostgreSQL               | 25        |
| 3.1.2            | Additional Information about the Bundled Tomcat and PostgreSQL | 25        |
| 3.1.3            | Start/Stop Scripts — No Bundled Applications                   | 26        |
| 3.2              | Start/Stop Scripts — Linux                                     | 26        |
| 3.2.1            | Manual Start/Stop  | 26        |
| 3.2.2            | Auto Start/Stop with Bundled Tomcat and PostgreSQL             | 27        |
| 3.3              | Start/Stop Apps — Mac OSX                                      | 27        |
| 3.3.1            | Start/Stop Apps — Mac Dock                                     | 27        |
| 3.3.2            | Start/Stop JasperReports Server — Mac Terminal Shell           | 27        |
| 3.4              | Logging into JasperReports Server                              | 28        |
| 3.5              | JasperReports Server Log Files                                 | 29        |
| <b>Chapter 4</b> | <b>Uninstalling JasperReports Server</b>                       | <b>31</b> |
| 4.1              | Windows  | 31        |
| 4.2              | Linux  | 31        |
| 4.3              | Mac OSX  | 32        |
| 4.4              | Uninstall Survey   | 32        |
| <b>Chapter 5</b> | <b>Installing the WAR File Distribution</b>                    | <b>33</b> |
| 5.1              | Applications Supported by the WAR File Distribution            | 33        |
| 5.1.1            | Database and Application Server Support                        | 33        |
| 5.1.2            | Operating System Support for Bash Shell                        | 34        |
| 5.2              | Installing the WAR File Using js-install Scripts               | 34        |
| 5.3              | Additional Steps for Using DB2 and js-install Scripts          | 36        |
| 5.4              | Starting JasperReports Server                                  | 37        |
| 5.5              | Logging into JasperReports Server                              | 37        |
| 5.5.1            | JasperReports Server Heartbeat                                 | 38        |
| 5.6              | Troubleshooting Your JasperReports Server Configuration        | 38        |
| 5.6.1            | JasperReports Server Startup Problems                          | 38        |
| 5.6.2            | Error Running a Report   | 38        |
| 5.6.3            | Error Running js-install Scripts (js-install.bat/sh)           | 38        |
| 5.6.4            | Problem Connecting to a Cloud Database Instance                | 39        |
| 5.7              | Installing the WAR File Manually                               | 39        |
| <b>Chapter 6</b> | <b>JVM Options, License Setup, Manual Database Creation</b>    | <b>41</b> |
| 6.1              | Setting JVM Options for Application Servers                    | 41        |
| 6.1.1            | Tomcat and JBoss JVM Options                                   | 41        |
| 6.1.2            | Changing JVM Options for Bundled Tomcat as a Windows Service   | 43        |
| 6.1.3            | Changing JVM Options for Existing Tomcat as a Windows Service  | 43        |
| 6.1.4            | Changing JVM Options for Bundled Tomcat on Linux               | 44        |
| 6.1.5            | Changing GlassFish JVM Options                                 | 44        |
| 6.2              | Setting Up the JasperReports Server License                    | 45        |
| 6.2.1            | Default License Configuration for All Application Servers      | 45        |

|                  |  |           |
|------------------|--|-----------|
| 6.2.2            | User-Defined License Location                                    | 46        |
| 6.3              | Manually Creating the JasperReports Server Database              | 48        |
| 6.3.1            | PostgreSQL   | 48        |
| 6.3.2            | MySQL  | 49        |
| 6.3.3            | Oracle   | 50        |
| 6.3.4            | DB2  | 51        |
| 6.3.5            | SQL Server   | 52        |
| 6.4              | Locating and Changing Buildomatic Configuration Files            | 53        |
| 6.4.1            | Regenerating Buildomatic Settings                                | 53        |
| 6.4.2            | Locating Buildomatic-Generated Property Files                    | 53        |
| 6.4.3            | Changing the JDBC Driver Deployed by Buildomatic                 | 54        |
| 6.4.4            | Buildomatic Location for JDBC Drivers                            | 55        |
| 6.4.5            | Buildomatic Location for JasperReports Server WAR File           | 56        |
| 6.4.6            | Buildomatic Location for SQL Scripts                             | 56        |
| 6.4.7            | Buildomatic Location for Database Creation Scripts               | 56        |
| 6.4.8            | Buildomatic Location for Sample Data Catalog ZIP Files           | 57        |
| 6.4.9            | Hibernate Properties Settings                                    | 57        |
| 6.4.10           | Database Connection Configuration Files                          | 57        |
| 6.5              | Configuring Report Scheduling                                    | 58        |
| 6.5.1            | Mail Server Configuration Settings                               | 59        |
| 6.5.2            | Database Settings for the Quartz Driver Delegate Class           | 59        |
| 6.5.3            | Settings for the Report Scheduler Web URI                        | 60        |
| 6.5.4            | Settings for the Quartz Table Prefix                             | 60        |
| 6.5.5            | Settings for Import-Export                                       | 60        |
| 6.5.6            | Setting Properties in the default_master.properties File         | 61        |
| 6.6              | Updating XML/A Connection Definitions                            | 61        |
| <b>Chapter 7</b> | <b>Installing the WAR File for WebSphere</b>                     | <b>63</b> |
| 7.1              | Procedure for Installing and Deploying the WAR File in WebSphere | 64        |
| 7.1.1            | Installing WebSphere and a Database                              | 64        |
| 7.1.2            | Preparing JasperReports Server Files                             | 64        |
| 7.1.3            | Configuring the Hibernate and Quartz Settings                    | 65        |
| 7.1.4            | Configuring a JDBC Provider in WebSphere                         | 66        |
| 7.1.5            | Deploying the JasperReports Server WAR File in WebSphere         | 71        |
| 7.1.6            | Setting JVM Options  | 72        |
| 7.1.7            | Starting and Restarting JasperReports Server                     | 73        |
| 7.2              | Logging into JasperReports Server                                | 73        |
| 7.3              | Configuring Report Scheduling                                    | 74        |
| 7.3.1            | Additional Fix for Scheduled Report with JNDI Data Source        | 74        |
| 7.3.2            | Additional Change for Mail Server Authentication                 | 74        |
| 7.4              | Updating XML/A Connection Definitions (Optional)                 | 75        |
| 7.5              | Troubleshooting your JasperReports Server Configuration          | 75        |
| 7.5.1            | Startup Problems   | 75        |
| 7.5.2            | Error Running Report   | 75        |
| 7.5.3            | Filter Error Using MySQL   | 75        |

|   |  |           |
|---|--|-----------|
| 7.5.4   | Error Creating Internationalized Name .....                            | 75        |
| 7.5.5   | Xerces Error .....   | 75        |
| 7.5.6   | OLAP View Fails With Exception .....                                   | 76        |
| <b>Chapter 8 Installing the WAR File for WebLogic .....</b> |  | <b>77</b> |
| 8.1   | Procedure for Installing the WAR File for WebLogic .....               | 78        |
| 8.2   | Setting Java Properties .....  | 83        |
| 8.3   | Configuring Other Database Connections .....                           | 84        |
| 8.4   | Starting JasperReports Server .....                                    | 85        |
| 8.5   | Logging into JasperReports Server .....                                | 86        |
| 8.6   | Configuring Report Scheduling .....                                    | 86        |
| 8.7   | Restarting JasperReports Server .....                                  | 86        |
| 8.8   | Updating XML/A Connection Definitions (Optional) .....                 | 86        |
| 8.9   | Troubleshooting Your JasperReports Server Configuration .....          | 86        |
| 8.9.1   | Startup Problems .....   | 87        |
| 8.9.2   | Error Running Report .....   | 87        |
| 8.10  | Troubleshooting Problems with OLAP Views .....                         | 87        |
| <b>Chapter 9 Overlay Upgrade .....</b>                      |  | <b>89</b> |
| 9.1   | Introduction to the Overlay Upgrade .....                              | 89        |
| 9.2   | Upgrade Steps Overview .....   | 90        |
| 9.3   | Back Up Your JasperReports Server Instance .....                       | 90        |
| 9.4   | Unpack the Overlay Upgrade Package .....                               | 90        |
| 9.5   | Run the Overlay Upgrade .....  | 91        |
| 9.6   | Rerun the Overlay Upgrade .....  | 91        |
| 9.7   | Rollback Procedure .....   | 92        |
| 9.8   | Starting and Logging into JasperReports Server 5.5 .....               | 92        |
| 9.8.1   | Clearing Your Browser Cache .....                                      | 92        |
| 9.8.2   | Logging into JasperReports Server .....                                | 92        |
| 9.9   | Additional Tasks to Complete the Upgrade .....                         | 93        |
| 9.9.1   | Clearing the Application Server Temp Folder .....                      | 93        |
| 9.10  | Running Overlay Upgrade a Second Time .....                            | 93        |
| <b>Chapter 10 Upgrading from 5.2 to 5.5 .....</b>           |  | <b>95</b> |
| 10.1  | Upgrade Steps Overview .....   | 95        |
| 10.2  | Back Up Your JasperReports Server Instance .....                       | 95        |
| 10.3  | Preparing the JasperReports Server 5.5 WAR File Distribution .....     | 96        |
| 10.4  | Configuring Buildomatic for Your Database and Application Server ..... | 96        |
| 10.4.1  | Example Buildomatic Configuration .....                                | 96        |
| 10.5  | Upgrading to JasperReports Server 5.5 .....                            | 98        |
| 10.5.1  | js-upgrade Test .....  | 98        |
| 10.5.2  | Output Log Location .....  | 98        |
| 10.5.3  | Errors .....   | 98        |
| 10.6  | Starting and Logging into JasperReports Server 5.5 .....               | 99        |
| 10.6.1  | Clearing Your Browser Cache .....                                      | 99        |

|                   |  |            |
|-------------------|--|------------|
| 10.6.2            | Logging into JasperReports Server                                | 99         |
| 10.7              | Additional Tasks to Complete the Upgrade                         | 99         |
| 10.7.1            | Clearing the Application Server Work Folder                      | 99         |
| 10.7.2            | Clearing the Application Server Temp Folder                      | 99         |
| 10.7.3            | Clearing the Repository Cache Database Table                     | 100        |
| 10.8              | Old Manual Upgrade Steps: 5.2 to 5.5                             | 100        |
| <b>Chapter 11</b> | <b>Upgrading from 3.7 - 5.2 to 5.5</b>                           | <b>101</b> |
| 11.1              | Upgrade Steps Overview   | 102        |
| 11.2              | Planning Your Upgrade  | 102        |
| 11.3              | Back Up Your JasperReports Server Instance                       | 102        |
| 11.4              | Exporting Current Repository Data                                | 103        |
| 11.4.1            | Exporting from the UI  | 103        |
| 11.4.2            | Using Buildomatic Scripts to Export Data                         | 103        |
| 11.4.3            | Using the js-export Script to Export Data                        | 103        |
| 11.5              | Preparing the JasperReports Server 5.5 WAR File Distribution     | 104        |
| 11.6              | Configuring Buildomatic for Your Database and Application Server | 104        |
| 11.6.1            | Example Buildomatic Configuration                                | 104        |
| 11.7              | Upgrading to JasperReports Server 5.5                            | 106        |
| 11.7.1            | js-upgrade Test Mode   | 106        |
| 11.7.2            | Output Log Location  | 106        |
| 11.7.3            | Errors   | 106        |
| 11.8              | Starting and Logging into JasperReports Server 5.5               | 107        |
| 11.8.1            | Clearing Your Browser Cache                                      | 107        |
| 11.8.2            | Logging into JasperReports Server                                | 107        |
| 11.9              | Additional Tasks to Complete the Upgrade                         | 107        |
| 11.9.1            | Handling JasperReports Server Customizations                     | 107        |
| 11.9.2            | Clearing the Application Server Work Folder                      | 107        |
| 11.9.3            | Clearing the Application Server Temp Folder                      | 107        |
| 11.9.4            | Clearing the Repository Cache Database Table                     | 108        |
| 11.10             | Old Manual Upgrade Steps   | 108        |
| <b>Chapter 12</b> | <b>Planning Your Upgrade from 4.7</b>                            | <b>109</b> |
| 12.1              | Understanding the Changes in 4.7                                 | 109        |
| 12.2              | Migrating Ad Hoc Reports to Ad Hoc Views                         | 110        |
| 12.2.1            | Changes to Ad Hoc  | 110        |
| 12.2.2            | Output of Migration Process                                      | 111        |
| 12.2.3            | Ad Hoc Migration Process   | 111        |
| 12.2.4            | Issues in Migrated Reports                                       | 112        |
| 12.2.5            | Unsuccessful Migration   | 112        |
| 12.2.6            | Troubleshooting  | 113        |
| 12.3              | Upgrading Themes in 4.7  | 113        |
| 12.3.1            | Banner and Toolbar Modifications                                 | 114        |
| 12.3.2            | Changes to IE Overrides  | 115        |
| 12.3.3            | Images in JasperReports Server 4.7                               | 116        |

|                   |  |            |
|-------------------|--|------------|
| 12.4              | Data Snapshots   | 121        |
| 12.4.1            | Using Data Snapshots   | 122        |
| 12.5              | Understanding the Changes in 5.0                                 | 122        |
| 12.6              | Upgrading Preserving Custom Settings                             | 122        |
| 12.7              | XML/A Connections to Servers with Multiple Organizations         | 122        |
| <b>Chapter 13</b> | <b>Upgrading JasperServer 3.5 or Earlier</b>                     | <b>123</b> |
| 13.1              | Upgrading from 3.5 or Earlier                                    | 123        |
| 13.2              | Best Practices for Upgrading under Windows                       | 123        |
| <b>Chapter 14</b> | <b>Upgrading from the Community Project</b>                      | <b>125</b> |
| 14.1              | General Procedure  | 125        |
| 14.2              | Backing Up Your JasperReports Server CP Instance                 | 126        |
| 14.2.1            | Backing Up Your JasperReports Server CP WAR File                 | 126        |
| 14.2.2            | Backing Up Your JasperReports Server Database                    | 126        |
| 14.3              | Exporting Your CP Repository Data                                | 126        |
| 14.4              | Preparing the JasperReports Server 5.5 WAR File Distribution     | 127        |
| 14.5              | Configuring Buildomatic for Your Database and Application Server | 127        |
| 14.5.1            | Example Buildomatic Configuration                                | 127        |
| 14.6              | Upgrading to the Commercial Version of JasperReports Server 5.5  | 128        |
| 14.7              | Starting and Logging into JasperReports Server 5.5               | 128        |
| 14.7.1            | Clearing Your Browser Cache                                      | 129        |
| 14.7.2            | Logging into the Commercial Version of JasperReports Server 5.5  | 129        |
| 14.8              | Re-Configuring XML/A Connections (Optional)                      | 129        |
| <b>Chapter 15</b> | <b>Password Encryption in JasperReports Server</b>               | <b>131</b> |
| 15.1              | Backing Up Your JasperReports Server Database                    | 131        |
| 15.2              | Stopping Your Application Server                                 | 131        |
| 15.3              | Running the Repository Export Utility                            | 132        |
| 15.4              | Specifying Encryption Settings in the JasperReports Server WAR   | 132        |
| 15.4.1            | Specifying Encryption Settings - Reference Table                 | 132        |
| 15.5              | Specifying Encryption Settings for the Import Utility            | 133        |
| 15.6              | Recreating the JasperReports Server Database                     | 133        |
| 15.6.1            | Dropping and Recreating the Database in PostgreSQL               | 133        |
| 15.6.2            | Dropping and Recreating the Database in MySQL                    | 134        |
| 15.6.3            | Dropping and Recreating the Database in Oracle                   | 134        |
| 15.6.4            | Dropping and Recreating in the Database in Microsoft SQL Server  | 134        |
| 15.7              | Importing Your Repository Data                                   | 135        |
| 15.8              | Starting the Application Server                                  | 135        |
| 15.9              | Logging into JasperReports Server                                | 135        |
| <b>Chapter 16</b> | <b>Configuring the Import-Export Utilities</b>                   | <b>137</b> |
| 16.1              | Introduction   | 137        |
| 16.2              | Import-Export Configuration Files                                | 138        |
| 16.3              | Changing Your Configuration Settings                             | 138        |



|                   |   |            |
|-------------------|---|------------|
| 16.3.1            | Creating a default_master.properties File . . . . .                       | 138        |
| 16.3.2            | Location of Properties Files and JDBC Driver . . . . .                    | 139        |
| 16.3.3            | Checking the js.jdbc.properties File . . . . .                            | 139        |
| 16.3.4            | Checking the js.quartz.properties File . . . . .                          | 140        |
| 16.4              | Adding an Additional JDBC Driver . . . . .                                | 141        |
| 16.5              | Running Import or Export . . . . .  | 141        |
| 16.5.1            | Running Import or Export from the UI . . . . .                            | 141        |
| 16.5.2            | Running Import from Buildomatic . . . . .                                 | 142        |
| 16.5.3            | Running Export from Buildomatic . . . . .                                 | 142        |
| 16.5.4            | Running the Import-Export Shell Scripts . . . . .                         | 143        |
| <b>Appendix A</b> | <b>Troubleshooting . . . . .</b>  | <b>145</b> |
| A.1               | Binary Installer Freezes . . . . .  | 145        |
| A.1.1             | Installer Log Files . . . . .   | 145        |
| A.1.2             | Installer DebugTrace Mode . . . . .                                       | 146        |
| A.2               | Error Running Buildomatic Scripts . . . . .                               | 146        |
| A.2.1             | Missing Java JDK . . . . .  | 146        |
| A.2.2             | Forgot to Copy the File ant-contrib.jar . . . . .                         | 146        |
| A.2.3             | Failure with '\$' Character in Passwords in Buildomatic Scripts . . . . . | 146        |
| A.2.4             | Older Apache Ant Version . . . . .  | 147        |
| A.3               | Unable to Edit Files on Windows 7 . . . . .                               | 147        |
| A.4               | Bash Shell for Solaris, IBM AIX, HP UX and FreeBSD . . . . .              | 147        |
| A.5               | Linux Installer Issue with Unknown Host Error . . . . .                   | 148        |
| A.6               | Installation Error with Windows Path . . . . .                            | 149        |
| A.7               | Mac OSX Issues . . . . .  | 149        |
| A.7.1             | Problem Starting JasperReports Server on Mac . . . . .                    | 149        |
| A.7.2             | Installation Error on Mac OSX 10.8 (Lion) . . . . .                       | 150        |
| A.8               | Database-related Problems . . . . .                                       | 150        |
| A.8.1             | Database Connectivity Errors . . . . .                                    | 150        |
| A.8.2             | Case Sensitive Collation in SQL Server . . . . .                          | 151        |
| A.8.3             | Maximum Packet Size in MySQL . . . . .                                    | 151        |
| A.8.4             | Case Sensitivity for Table and Column Names . . . . .                     | 151        |
| A.8.5             | PostgreSQL: Job Scheduling Error . . . . .                                | 152        |
| A.8.6             | Oracle js-install Script Hangs with Oracle 10g . . . . .                  | 152        |
| A.8.7             | Performance Issues with Oracle JDBC Queries . . . . .                     | 152        |
| A.8.8             | Using an Oracle Service Name . . . . .                                    | 153        |
| A.8.9             | Oracle Error on Upgrade when PL/SQL Not Enabled . . . . .                 | 153        |
| A.8.10            | Error Running Scheduled Report . . . . .                                  | 153        |
| A.8.11            | Error Running a Report . . . . .  | 153        |
| A.8.12            | Save Error with DB2 Database . . . . .                                    | 154        |
| A.9               | Application Server-related Problems . . . . .                             | 154        |
| A.9.1             | Memory Issues Running Under Tomcat . . . . .                              | 154        |
| A.9.2             | Java Out of Memory Error . . . . .  | 154        |
| A.9.3             | Configuration File Locations . . . . .                                    | 154        |
| A.9.4             | Context.xml under Tomcat: Special Case . . . . .                          | 155        |

|        |  |     |
|--------|--|-----|
| A.9.5  | Tomcat 6 Installed Using apt-get .....   | 155 |
| A.9.6  | GlassFish Modifications .....  | 155 |
| A.9.7  | JBoss Modifications .....  | 156 |
| A.9.8  | Websphere Modifications .....  | 159 |
| A.9.9  | Disabling User Session Persistence in Application Servers .....                                | 159 |
| A.9.10 | Session Error Using JasperReports Server and Tomcat 7 .....                                    | 160 |
| A.10   | License-related Errors .....   | 161 |
| A.10.1 | License Not Found Errors .....   | 161 |
| A.10.2 | License Not Found or License Corrupt Error with Tomcat as a Service .....                      | 161 |
| A.11   | Problems Importing and Exporting Data from the Repository .....                                | 161 |
| A.11.1 | Exporting a Repository That Contains UTF-8 .....   | 161 |
| A.12   | Problems with Upgrade .....  | 162 |
| A.12.1 | DB2 Script Error on Upgrade .....  | 162 |
| A.12.2 | Include Audit Events on Upgrade .....  | 162 |
| A.13   | Installing on Windows XP if Memory Error .....   | 163 |
| A.14   | Property File Updates .....  | 163 |
| A.14.1 | Owasp.CsrfGuard.properties - Security File .....   | 163 |
| A.15   | Permissions Error in Overlay Upgrade with PostgreSQL and Bundled Jaspersoft Installation ..... | 164 |

## CHAPTER 1 INTRODUCTION

---

JasperReports Server builds on JasperReports 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.

The heart of the 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 Jaspersoft OLAP, which runs on the server.

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

- Our Ultimate Guides document advanced features and configuration. They also include best practice recommendations and numerous examples. The guides are available as downloadable PDFs. Community project users can purchase individual guides or bundled documentation packs from the Jaspersoft [online store](#). Commercial customers can download them freely from the [support portal](#).
- Our free [Business Intelligence Tutorials](#) let you learn at your own pace, and cover topics for developers, system administrators, business users, and data integration users. The tutorials are available online from the Professional Services section of our [website](#).
- Free samples installed with JasperReports, and JasperReports Server, are documented online in the [Community Wiki](#). For more information, visit our [community website](#).

This chapter contains the following sections:

- **Conventions**
- **Java Version Supported**
- **JasperReports Server Distributions**
- **Release Notes**
- **System Requirements**
- **Support for Internationalization**

## 1.1 Conventions

This document uses the following conventions when referring to file locations:

| Convention   | Description   |
|--------------|---|
| <js-install> | The root directory where JasperReports Server will be installed.<br>For manual installations, the directory where you unpack the WAR file distribution.                     |
| <glassfish>  | The directory where GlassFish is installed.   |
| <java>       | The directory where java is installed.  |
| <jboss>      | The directory where JBoss is installed.   |
| <postgresql> | The directory where PostgreSQL is installed. If you use the instance of PostgreSQL that is bundled by the installer, <postgresql> is located in the <js-install> directory. |
| <tomcat>     | The directory where Apache Tomcat is installed. If you use the instance of Tomcat that is bundled by the installer, <tomcat> is located in <js-install>.                    |

## 1.2 Java Version Supported

JasperReports Server supports Java 1.6 and 1.7. Versions earlier than Java 1.6 are not supported.

JasperReports Server is tested and certified using Oracle/Sun Java. OpenJDK 1.6 has also been certified to run with JasperReports Server.

## 1.3 JasperReports Server Distributions

There are two main distribution packages for JasperReports Server.

| Distribution Package      | Description   |
|---------------------------|---|
| Installer                 | Runs on Windows, Linux, and Mac OSX (32 or 64 bit).                       |
| WAR File Distribution Zip | Used for manual installation on Windows, Linux, Mac, and other platforms. |

The installer distribution package installs JasperReports Server, automatically configures the JasperReports Server database, and, if you choose the Install Sample Data option, installs sample data for working with tutorials.

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

### 1.3.1 Installer Support

The installers support the following operating systems (32 and 64 bit):

| Platform | Versions supported  |
|----------|---|
| Linux    | Red Hat Enterprise Linux 5, 6<br>Novell SUSE Linux Enterprise 10, 11<br>Debian 6<br>Ubuntu 10 |

| Platform | Versions supported                           |
|----------|--|
| Mac OSX  | 10.6 (Snow Leopard)<br>10.7 (Lion)           |
| Windows  | XP<br>Windows 2008<br>Windows 7<br>Windows 8 |

### 1.3.1.1 Installer Naming for 32-bit and 64-bit

Native 32- and 64-bit installers are supported. The 64-bit installer will put 64-bit versions of Java 7 and PostgreSQL 9 onto your system for increased speed and performance.

The installer file naming distinguishes the 32-bit installer from the 64-bit installer.

| Installer Type  | Naming   |
|---|--|
| 32-bit installer  | jasperreports-server-5.5-windows-x86-installer.exe<br>jasperreports-server-5.5-linux-x86-installer.run   |
| 64-bit installer  | jasperreports-server-5.5-windows-x64-installer.exe<br>jasperreports-server-5.5-linux-x64-installer.run<br>jasperreports-server-5.5-osx-x64-installer.app.zip |
| Note: x86 is shorthand referring to the 386, 486, and 586 CPU architecture. |  |

Note: You can install the 32-bit installer onto a 64-bit operating system, but we recommend that you install the 64-bit installer onto a 64-bit system. The 64-bit installer will not execute on a 32-bit system.

### 1.3.1.2 Installer Distribution Components

The installer is designed to get JasperReports Server up and running quickly. The server requires the Java environment, an application server, and database to run. The installer distribution bundles these components:

| Component                          | Description  |
|------------------------------------|--|
| JasperReports Server Application   | WAR file and configuration support scripts.  |
| JasperReports Server Documentation | Found in the <js-install>/docs directory.  |
| Apache Tomcat 7                    | Web application container. You can use the bundled version or an existing version. |
| Java 1.7 Runtime                   | Runs the web application container.  |
| PostgreSQL 9 Database              | Database server. You can use the bundled version or an existing version.           |

### 1.3.1.3 Installing with Existing Components

You can choose to deploy the bundled application or if you have existing components, the installer can deploy to these components. Both Apache Tomcat and the PostgreSQL database can be independently used as bundled or existing instances.

If you would like the installer to install Tomcat, choose the bundled Tomcat. If you already have Tomcat on your computer you can choose an existing Tomcat.



If you use an existing Tomcat, it must be on the local machine.

If you use an existing PostgreSQL, it can be on a local or remote machine. If it's on a remote Linux machine, configure PostgreSQL to allow remote connections as described in [2.7.4, “Enabling Connections to a Remote Host,” on page 21](#).

For information about specific versions of third party applications supported by the installer, refer to the JasperReports Server release notes in the root of the installation directory.

### 1.3.1.4 Running Components as Windows Services

The Windows installer installs PostgreSQL and Tomcat as Windows Services. Users can manage JasperReports Server under the Windows operating system using Services in the Control Panel:

**Control Panel > System and Security > Administrative Tools > Services**

The bundled PostgreSQL and Tomcat applications restart automatically when the host Windows system restarts. If you do not want to run these components to automatically restart, you can change the Startup Type from automatic to manual.

You can find the PostgreSQL and Tomcat services under the following names:

- ♦ jasperreportsPostgreSQL
- ♦ jasperreportsTomcat

You can also start JasperReports Server from the Windows Start menu.

### 1.3.2 WAR File Binary Distribution Support

Use the WAR file binary distribution package to install the JasperReports Server application if you cannot use the installer. The WAR file supports more applications than the installer. If you want to use a database other than PostgreSQL and an application server other than Apache Tomcat, install JasperReports Server using the WAR file.



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

The target database can be on a remote server. Using a remote PostgreSQL database on some Linux platforms requires a change to its configuration file, as described in section [2.7.4, “Enabling Connections to a Remote Host,” on page 21](#).

The application server should reside on the local machine.

There are js-install shell scripts (for Linux and Window) included in the WAR file distribution which automate much of the installation tasks by using a single properties file. These scripts are named:

- ♦ js-install.bat
- ♦ js-install.sh

The main contents of the WAR file binary distribution are:

| Content Item                              | Description  |
|---|--|
| JasperReports Server js-install scripts   | Found at <js-install>/buildomatic/js-install.bat and js-install.sh.                                    |
| JasperReports Server Database Scripts     | SQL scripts for each supported database.   |
| JasperReports Server Documentation        | Guides for end users and administrators.   |
| JasperReports Server Extra Samples        | Web Service example applications, sample reports, custom data source examples, and other sample files. |
| JasperReports Server Standard Sample Data | Sample data that highlights JasperReports Server features.   |
| JasperReports Server WAR file archive     | All of the JasperReports Server class files and dependent jars.  |

### 1.3.2.1 About Bundled Apache Ant

The War File Distribution ZIP comes with a bundled version of Apache Ant so you do not need to download or install Ant. The buildomatic Ant scripts come with Windows and Linux batch scripts that are pre-configured to use the bundled version of Apache Ant. The buildomatic Ant scripts are called from the command line in the following manner:

Windows: `js-ant <target-name>`  
 Linux and Mac OSX: `./js-ant <target-name>`

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/lib` folder.



On Linux and Solaris, the `js-ant` commands may not be compatible with all shells. If you have errors, use the `bash` shell explicitly. For more information, see section [A.4, “Bash Shell for Solaris, IBM AIX, HP UX and FreeBSD,” on page 147](#).

## 1.4 Release Notes

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

Not all applications are immediately supported when a new JasperReports Server version is released. For instance, some applications require additional testing beyond what is completed for the initial General Availability (GA) release. To find out exactly what applications are supported with a particular distribution refer to the release notes found in that distribution.

## 1.5 System Requirements

The following table contains the minimum and recommended resources for a full installation that includes PostgreSQL and an application server. The values are based on our own testing. You may find that JasperReports Server can run on systems with fewer resources or slower systems than stated in the minimum resources column. At the same time, it is possible to run out of resources with the recommended configuration. The success of your deployment depends on the intended load of the system, the number of concurrent users, the data sets, and whether the databases are installed on the same system as the JasperReports Server.

| Resource  | Footprint      | Minimum        | Recommended   |
|-----------|----------------|----------------|---|
| Disk      | ~1.3 Gigabytes | 10GB free      | 40GB +  |
| RAM       |                | 4GB            | 8GB +   |
| Processor |                | 2 core minimum | 2.5GHz + multi-core Pentium for Windows, Mac, and Linux |

**Note on Windows XP:** The maximum RAM that can be addressed on a 32 bit machine is about 3.5 Gigabytes. So, for a Windows XP installation this amount of memory is the minimum required for JasperReports Server. If you install to Windows XP using the binary installer please see Troubleshooting Section A.10 “Installing to Windows XP” for more information.

## 1.6 Support for Internationalization

JasperReports Server supports the full Unicode character set using UTF-8 encoding. It also depends on the underlying database and application server to support the UTF-8 character encoding. If you use the bundled Tomcat and PostgreSQL software, UTF-8 is configured by default. If you use any other software, refer to the *JasperReports Server Administrator Guide* for instructions about configuring software to support UTF-8.





## CHAPTER 2    INSTALLING JASPERREPORTS SERVER

---

This chapter contains the following sections:

- **Pre-Installation Steps**
- **Starting the Installer**
- **Accepting the License Agreement**
- **Choosing Installation Type**
- **Selecting a Tomcat Configuration**
- **Selecting a PostgreSQL Configuration**
- **Installing Sample Data**
- **Completing the Installation**
- **Post-Installation Steps**

### 2.1    Pre-Installation Steps

When you run the installation executable, you are given the option to install a bundled Apache Tomcat application server and PostgreSQL database or to use an existing Tomcat and PostgreSQL.



If you want to use an existing database instance, the database must be running at install time. If you want to use an existing Apache Tomcat, the Tomcat instance should be stopped.

If you choose to install the bundled Tomcat and database, both are installed on the same host with the server.



The bundled installer is not meant for use in Enterprise Production environments.

### 2.2    Starting the Installer

In Windows, the installer is an executable file that you can double-click to run. The installer under Windows will need a user account with Administrative privileges. The installer should also be started and run with “Run as administrator”. For example, right click on the binary installer file to bring up the context menu. Choose “Run as administrator”:

|   |          |
|---|----------|
| <code>jasperreports-server-5.5-windows-x86-installer.exe</code> | (32 bit) |
| <code>jasperreports-server-5.5-windows-x64-installer.exe</code> | (64 bit) |



The Windows installer will get an error installing the PostgreSQL database if the Windows user does not have sufficient Administrative privileges and if the installer is not started by right-clicking to use “Run as administrator”.

In Linux, the installer is a .run file; you can run it from the command line or from a graphical environment. To start the installer from the command line, open a bash shell, and enter the name of the installer file. For example:

```
./jasperreports-server-5.5-linux-x86-installer.run      (32 bit)
./jasperreports-server-5.5-linux-x64-installer.run      (64 bit)
```

In Mac OSX, the installer is a .zip file. Typically, after download, the installer will be found in your <user>/Downloads folder, and it will already be unpacked. After the download is complete, double-click the following:

```
jasperreports-server-5.5-osx-x64-installer.app          (64 bit Only)
```

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

The welcome screen introduces the installer and allows you to continue or exit. Click **Next**.



If you are installing a 32-bit installer onto a 64-bit operating system you will normally get a popup message reminding you that a 64-bit installer is available. You may continue the 32-bit installation if you choose to.



The Windows installer will get an error installing the PostgreSQL database if the Windows user does not have sufficient Administrative privileges and if the installer is not started by right-clicking to use “Run as administrator”.

## 2.3 Accepting the License Agreement

You are prompted to read and accept the license agreement. Read the agreement, agree to the terms by clicking **I accept the agreement**, and click **Next**. On the command line, you must page through several screens of text to read the full agreement.

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

## 2.4 Choosing Installation Type

As of the 5.5 Release, the installer has been updated to offer a choice of “Install Type”. The first option will install all installer components and sample data resources. The second option is the “Custom Install”. With the custom install, you can choose which components to install and whether to include sample data resources.

### Install All Components and Samples Option

This option will copy a Bundled version of the Apache Tomcat package and a Bundled version of the PostgreSQL database to your file system. Additionally, all sample data resources (Reports, Data Sources, OLAP Views, etc) are added to your JasperReports Server and additional sample databases are created. With this install option, the installer will attempt to find open Tomcat ports in the 8080 and higher range. And for the PostgreSQL port, the installer will start with port 5432 and then try values higher than this if 5432 is already being used.

After you choose this first option, you can next choose the installation directory for JasperReports Server. Then, next, all files and components can be installed without requiring any further information.

### Custom Install

With the custom install, you will have the same choices that the installer has had in the past: install a Bundled Tomcat or use an Existing Tomcat, install a Bundled PostgreSQL or use an Existing PostgreSQL, choose ports for Tomcat and PostgreSQL, and choose whether to install sample data resources or not.

## 2.5 Choosing an Installation Directory

You are prompted for the directory where JasperReports Server is installed, referred to as the <js-install> directory. Accept the default or click **Browse** and select a different location, and click **Next**. On the command line, press Enter to accept the default. To choose a different directory location, enter that location at the prompt.

The default <js-install> directory depends on your operating system:

|                 |  |
|-----------------|--|
| Windows:        | C:\Jaspersoft\jasperreports-server-5.5 |
| Linux:          | <USER_HOME>/jasperreports-server-5.5   |
| Linux (as root) | /opt/jasperreports-server-5.5          |
| Mac OSX         | /Applications/jasperreports-server-5.5 |



On Linux, choose a <js-install> path that's no more than 84 characters.

## 2.6 Selecting a Tomcat Configuration

JasperReports Server requires an application server in order to run. The installer is pre-configured to run with the Apache Tomcat server. When you run the installer, two options appear on **Setup — Please select the Tomcat configuration you want to use:**

- **I want to use the bundled Tomcat**

If you choose this option, the installer puts an instance of Tomcat 6 onto your system. Later, after choosing a bundled or existing database, you are prompted for the server port and shutdown port that Tomcat will use. Most users accept the default values that are displayed. Accept the default values or enter alternate values, then click **Next**.

- **I want to use an existing Tomcat**

If you already have an instance of Tomcat on your system, you can choose this option. Later, after choosing a bundled or existing database, you are prompted for the location of Tomcat. You can browse to the folder where you installed Tomcat, such as C:\Apache Software Foundation\Tomcat 7.

After selecting a PostgreSQL configuration, you are prompted for Tomcat's server port and shutdown port. Accept the default values, 8080 and 8005 by default, or enter alternate values.

## 2.7 Selecting a PostgreSQL Configuration

JasperReports Server requires a database in order to run. The installer is pre-configured to run with the PostgreSQL database. There are two options available for your PostgreSQL database:

- **I want to use the bundled PostgreSQL database**
- **I want to use an existing PostgreSQL database**

### 2.7.1 Choosing the Bundled PostgreSQL

If you choose the option to install the bundled PostgreSQL, the installer puts PostgreSQL 9 onto your system. The default PostgreSQL port 5432 will be used. If the installer finds that port 5432 is already in use, you are prompted to pick an alternate port. In this case, choose an alternative port value. The installer sets the PostgreSQL administrator password to postgres and also creates a PostgreSQL database user with administrator privileges and credentials of jasperdb/password.

The following table summarizes the parameters set during installation of the bundled PostgreSQL:

| Parameter                          | Default Value and Description  |
|------------------------------------|--|
| Binary Directory                   | The directory where the postgres and pgAdmin3 binaries are located.  |
| Port                               | The port number that PostgreSQL uses (default is 5432). User must choose an alternate port if 5432 is in use.  |
| IP or Host Name                    | The IP address or name of the machine where PostgreSQL is installed. The default value is 127.0.0.1.   |
| PostgreSQL Administrative Password | Password of the database administrative user: postgres. The installer cannot handle special characters at the end of a password string. Incompatible characters include: & ; \$  |
| Database User Name                 | Hard coded default: jasperdb - The installer creates this user which is used to connect to the JasperReports Server database   |
| Database User Password             | Hard coded default: password - The installer uses this password for the jasperdb user.   |
| Additional notes for Linux         | If your Linux installation does not have a locale setting that supports UTF-8 encoding, your Bundled PostgreSQL instance will be initialized using a temporary locale (--locale=C). This will allow the PostgreSQL initdb to succeed with the desired UTF-8 database encoding. |

### 2.7.2 Choosing an Existing PostgreSQL on a Local Host

If you choose the option to use an existing PostgreSQL database, you are eventually prompted for the location of PostgreSQL and the port to use. If you have an instance of PostgreSQL installed locally, accept the default, which is 127.0.0.1, the localhost. Accept the default location for the PostgreSQL \bin directory, or click **Browse** to locate and select another location. You are also prompted for the default administrative account password of the PostgreSQL administrative user. The database administrative user account name postgres is used by default. Enter the database administrative user password and click **Enter**.



If the installer displays an error message saying FATAL: password authentication failed for user postgres, try re-entering the administrative password for your PostgreSQL database.

The following table summarizes the parameters set during the installation of an existing PostgreSQL:

| Defaults Used                       | Hardcoded Default Values Used or Created   |
|-------------------------------------|--|
| PostgreSQL Administrative User Name | postgres - The default administrative database user.   |
| jasperserver Database User Name     | jasperdb - The installer creates this database user which is used to connect to jasperserver database. |
| jasperserver Database User Password | password - The installer creates this password for the jasperdb database user.                         |



To improve system security, Jaspersoft recommends that you change the default password for jasperdb as soon as possible. To change the jasperdb connection password in JasperReports Server, edit: <js-install>/apache-tomcat/jasperserver-pro/META-INF/context.xml. (And delete, if it exists: <js-install>/apache-tomcat/conf/Catalina/localhost/jasperserver-pro.xml.) Then, make the same change in PostgreSQL using pgAdmin III or psql.

### 2.7.3 Using an Existing PostgreSQL on a Remote Host

If you are installing to a remote instance of PostgreSQL, you need the PostgreSQL client tools on your local machine. The version of client tools should match the remote PostgreSQL version. You can check the version of PostgreSQL instance by entering this command on the computer where it's installed:

```
psql --version
or
<path-to-postgresql-bin-folder>/psql --version
```

For instance: C:/Jaspersoft/PostgreSQL/9.0/bin/psql --version

**To verify that you can connect to the target remote PostgreSQL from the local installation machine:**

1. If necessary, install PostgreSQL client tools on your local, JasperReports Server machine.
2. Using your local PostgreSQL client tools, enter this command:

```
psql -U postgres -h <remote-host> -d postgres
or
<path-to-postgresql-bin-folder>/psql -U postgres -h <remote-host> -d postgres
```

You might also need to enable connections as described in the next section.

## 2.7.4 Enabling Connections to a Remote Host

On most platforms, the default PostgreSQL installation doesn't allow remote connections (as a security feature). You need to enable remote connections as described in this documentation:

- The PostgreSQL configuration documentation on the PostgreSQL web site
- The \docs directory of your PostgreSQL installation

**To enable connections from the installation machine to the remote PostgreSQL server:**

1. Locate the following PostgreSQL host-based authentication (hba) configuration file on the remote PostgreSQL server instance:  
 Windows: C:\Program Files\PostgreSQL\9.0\data\pg\_hba.conf  
 Linux: /var/lib/pgsql/data/pg\_hba.conf
2. Add the IP address of your local JasperReports Server installation machine to this file. For example, to allow the local installation machine with address 192.168.12.10 to connect to the PostgreSQL server, add this entry to the pg\_hba.conf file:  

```
host all 192.168.12.10/32 trust
```
3. Allow TCP/IP connections to the remote PostgreSQL server instance by making the following change to the postgresql.conf file on the remote machine:  
 From: listen\_addresses = 'localhost'  
 To: listen\_addresses = '\*'
4. Restart PostgreSQL.
5. Using your local PostgreSQL client tools, verify that you can connect to the target remote PostgreSQL from the local installation machine, as described in section [2.7.3, “Using an Existing PostgreSQL on a Remote Host,” on page 20](#).

## 2.8 Installing Sample Data

JasperReports Server can be installed with sample databases and sample reports for evaluating its features. Included are:

- Sugar CRM data that simulates three years of operations for a fictitious company that relies on the SugarCRM open source application.
- Foodmart data that simulates three years of operations for a fictitious company.
- JasperReports Server repository resources such as Reports, OLAP Views, Ad Hoc Topics, Domains, Data Sources, and Input Controls.
- Jaspersoft strongly recommends that you install this data, unless you are not interested in testing or evaluating with the default sample data.

During installation, the following prompt appears:

**Would you like to install sample databases and sample reports?**

Click **Yes** to install the sample data, and click **Next**.

## 2.9 Completing the Installation

After the files have been installed, you see the final installation screen. There are several post-installation options:

- **View Release Notes** - If you choose to view the release notes, you must exit the release notes text viewer before JasperReports Server will launch and open a browser (if you have chosen that option below).
- **Launch JasperReports Server Now** - If you choose to launch JasperReports Server from the installer, the installer exits and the application server starts if you chose the bundled Tomcat and PostgreSQL. A pause that lasts approximately 25 seconds occurs as the server starts up, then the login page appears in your system default browser. If you're installing under Linux, do not close the terminal window running the start script. For information about logging in, see section 3.4, [“Logging into JasperReports Server,” on page 28](#).



The **Launch JasperReports Server Now** check box option will only be displayed if you have chosen to install a bundled Tomcat and a bundled PostgreSQL. The menu based start/stop scripts only control the bundled applications that you chose to be installed. For more information, see [Chapter 3, “Starting and Stopping JasperReports Server,” on page 25](#).

Additionally, if you do not choose to Launch JasperReports Server Now the bundled components will not be started. If you only have one bundled component this component will not be started unless you use the Start/Stop menus or scripts. To Start and Stop JasperReports Server see [Chapter 3, “Starting and Stopping JasperReports Server,” on page 25](#).

- **Opt-in for JasperServer Heartbeat** - When the heartbeat is enabled, the server sends anonymous system and version information to Jaspersoft using https. JasperReports Server heartbeat information helps Jaspersoft create better products by improving our understanding of customer installation environments. For more information, see section 5.5.1, [“JasperReports Server Heartbeat,” on page 38](#).

Make your choices, then click **Finish**.

You should now be ready to log into the server.

## 2.10 Post-Installation Steps

### 2.10.1 Updates Made by the Installer During Installation

This section lists the standard updates that the installer makes to your local environment if you install to existing applications. When the installation completes, you can check that the updates, or corresponding changes, were successful.

#### Updates made to the application server

If you installed to an existing Tomcat, the following modifications to the Tomcat environment were attempted:

| File or Directory   | Updates  |
|---|--|
| Windows: bin/setclasspath.bat<br>Linux and Mac OSX: bin/setclasspath.sh | Modifies JAVA_OPTS to add -Djs.license.directory.<br>(Commercial installer only) |

| File or Directory   | Updates   |
|---|---|
| Windows: bin/setenv.bat<br>Linux and Mac OSX: bin/setenv.sh | Creates this file. Sets increased Java memory allocation values to JAVA_OPTS. For additional settings, refer to section 6.1, “ <a href="#">Setting JVM Options for Application Servers</a> ,” on page 41. |
| Tomcat 5: common/lib<br>Tomcat 6 and 7: lib                 | Adds PostgreSQL JDBC driver to this directory.<br>As of 5.1, add additional JDBC drivers for other databases.   |

### Updates made to the PostgreSQL database

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

| PostgreSQL Updates                            | Description   |
|---|---|
| Database <code>jasperserver</code> created    | This is the JasperReports Server repository database. This database holds all of system information, such as users, roles, datasources, and report definitions. |
| Database user <code>jasperdb</code> created   | The JasperReports Server application uses this user to connect to the database.   |
| Sample database <code>foodmart</code> created | (optional) Database created if install sample data option was chosen.   |
| Sample database <code>sugarcrm</code> created | (optional) Database created if install sample data option was chosen.   |

## 2.10.2 Installer Output Log File Location

The installer creates a log during installation that records information as the installation progresses. If you encounter any problems when you install JasperReports Server, it can be helpful to look at the installer log. You can find the installer log at `<js-install>/installation.log`.

## 2.10.3 Installing a New License File

By default, JasperReports Server is installed with an evaluation license that expires a number of days after the software is installed. After the license expires, you can start the server, but you cannot log in.

To obtain a commercial license, contact Technical Support or your sales representative.

To upgrade the evaluation license to a commercial one, copy the commercial license file over the evaluation license file. Application servers have work directories where JSP files are compiled and cached and other objects are stored. These directories can cause errors when updating to a new license. To avoid errors, clear the work directory before upgrading your license. For instance, if you’re using Tomcat:

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

After changing to a commercial license, make sure you stop the server before replacing the license file:

1. Stop the server
2. Replace the license named `jasperserver.license` in the deployed jasperreports server root directory with the new license file

The file name should be `jasperserver.license`

3. Restart the server

By default, the license is in the `<js-install>` directory, but can be located elsewhere. You need to define the `-Djs.license.directory` Java Environment Variable in the Tomcat startup scripts to point to the license location. The name of the license file is `jasperserver.license`. You may have to rename the new license file to this name.

Restart JasperReports Server and log in to see if the license grants access. For information about license errors, see the troubleshooting section [A.10, “License-related Errors,” on page 161](#).

For additional license configuration options, refer to section [6.2, “Setting Up the JasperReports Server License,” on page 45](#).

### 2.10.4 License File for Existing Tomcat as Windows Service

If you are installing JasperReports Server into an existing Tomcat installation on a Windows system that is running as a Windows Service *and* the license file is not in the default location because you did not choose the default `<js-install>` installation directory (section [2.4, “Choosing Installation Type,” on page 18](#)), you will probably have to manually configure Tomcat to locate the license file.

Follow these steps to examine and update the license location:

1. Open the Tomcat configuration tool by right-clicking the Tomcat icon in your quick-launch bar (usually in the lower-right corner of your desktop) or from the Windows 7 menu, expand **Start > All Programs > Apache Tomcat**. Right-click **Configure Tomcat** and select **Run as administrator**.
2. Select the **Java** tab.
3. At the bottom of the **Java Options** field, enter the following option:  
`-Djs.license.directory=<js-install>`  
 For example:  
`-Djs.license.directory=C:\Jaspersoft\jasperreports-server-5.5`
4. Stop and restart the application server.

You should now be able to run JasperReports Server.

### 2.10.5 Checking your Java JVM Options

For both the bundled Tomcat and the existing Tomcat, the installer attempts to set Java JVM options to help with memory allocation. You can double-check the values set to see that they are appropriate for your installation. If you installed a bundled version of Tomcat from the installer, these are the default Java JVM options for heap memory allocation:

| Installer Type                 | Setting                                     | File Location                              |
|--------------------------------|---|--|
| 32 bit (x86) Windows           | -Xms512m -Xmx1024m<br>-XX:MaxPermSize=512m  | <js-install>/apache-tomcat/bin/service.bat |
| 32 bit (x86) Linux and Mac OSX | -Xms512m -Xmx1024m<br>-XX:MaxPermSize=512m  | <js-install>/apache-tomcat/scripts/ctl.sh  |
| 64 bit (x64) Windows           | -Xms1024m -Xmx2048m<br>-XX:MaxPermSize=512m | <js-install>/apache-tomcat/bin/service.bat |
| 64 bit (x64) Linux and Mac OSX | -Xms1024m -Xmx2048m<br>-XX:MaxPermSize=512m | <js-install>/apache-tomcat/scripts/ctl.sh  |

**Note on Windows XP:** On a Windows XP installation you may need to decrease the JAVA\_OPTS memory settings. Otherwise, you might receive an “Out of Memory” error which you would see inside the `<js-install>/apache-tomcat/logs/catalina.out` log file. If you are installing to Windows XP, please see section [A.13, “Installing on Windows XP if Memory Error,” on page 163](#) for more information.



## CHAPTER 3 STARTING AND STOPPING JASPERREPORTS SERVER

---

This chapter contains the following sections:

- [Start/Stop Menu — Windows](#)
- [Start/Stop Scripts — Linux](#)
- [Start/Stop Apps — Mac OSX](#)
- [Logging into JasperReports Server](#)
- [JasperReports Server Log Files](#)

### 3.1 Start/Stop Menu — Windows

This section describes different start and stop procedures depending on how you installed JasperReports Server: using the bundled Tomcat and PostgreSQL or using an existing Tomcat and PostgreSQL.

#### 3.1.1 Start/Stop Menus — Bundled Tomcat and PostgreSQL

If you chose to install a bundled Tomcat and a bundled PostgreSQL with JasperReports Server, use the Windows Start menu items to start and stop JasperReports Server.

**To start or stop JasperReports Server from the Windows Start menu:**

- Click **Start > All Programs > JasperReports Server > Start or Stop Services > Start Service**.
- Click **Start > All Programs > JasperReports Server > Start or Stop Services > Stop Service**.

#### 3.1.2 Additional Information about the Bundled Tomcat and PostgreSQL

**JasperReports Server Windows Service Names:**

The Windows Services Panel lists entries for PostgreSQL and Tomcat which are installed as Windows Services by the installer. These services are listed as:

- jasperreportsPostgreSQL
- jasperreportsTomcat

**Preventing JasperReports Server from starting up automatically:**

By default, the bundled services are started automatically on a reboot. Consequently, the JasperReports Server also automatically starts. You can change the startup mode for the services from automatic to manual:

- In the Windows Services Panel, select jasperreportsTomcat
- Right-click the jasperreportsTomcat service, and select properties
- Change the Startup type drop-down setting from Automatic to Manual
- Do the same for the jasperreportsPostgreSQL service

### To Start JasperReports Server from the Windows Services Panel:

- Open the Windows Services Panel
- Select jasperreportsPostgreSQL, click Start
- Select jasperreportsTomcat, click Start

### To Start JasperReports Server from the CMD Shell:

JasperReports Server can be manually started from a Windows Command Shell:

- Open a Windows CMD Shell
- Navigate to the root of the <js-install> folder (i.e. C:\Jaspersoft\jasperreports-server-<ver>)
- `servicerun START`
- `servicerun STOP` (to shutdown JasperReports Server)

### Running Processes:

When JasperReports Server is running, the Windows Task Manager lists information about the processes running under the SYSTEM user name:

- `postgres.exe`
- `tomcat7.exe`

## 3.1.3 Start/Stop Scripts — No Bundled Applications

During installation, if you chose to install one bundled and one existing Tomcat or PostgreSQL, you can use the Windows start/stop scripts to start and stop only the bundled one.

For example, if you have an existing Tomcat and you install the bundled PostgreSQL, the scripts and menus specified in the previous section would start and stop the PostgreSQL application. To start and stop the existing Tomcat, you would use the management scripts provided by the Tomcat application.



JasperReports Server needs to have database and application servers started in this order:

- First, start the database server.
- Next, start the application server.

## 3.2 Start/Stop Scripts — Linux

This section describes different start and stop procedures depending on how you installed JasperReports Server: using the bundled Tomcat and PostgreSQL or using an existing Tomcat and PostgreSQL.

### 3.2.1 Manual Start/Stop

You typically start and stop JasperReports Server at the Linux command line. Run the following commands in a Linux shell.

Start JasperReports Server:

```
cd <js-install>
./ctlscript.sh start
```

Stop JasperReports Server:

```
cd <js-install>
./ctlscript.sh stop
```

To start and stop individual components:

```
cd <js-install>
./ctlscript.sh start|stop postgresql
./ctlscript.sh start|stop tomcat
```

### 3.2.2 Auto Start/Stop with Bundled Tomcat and PostgreSQL

To have JasperReports Server automatically start when you reboot your Linux server, you need to install the JRS database and application server as services. If you have installed JasperReports Server using the binary installer with the bundled Tomcat and bundled PostgreSQL options, an example jasperserver service script can be found in the following location:

<js-install>/scripts/linux/jasperserver

Edit this script and set permissions as described in the <js-install>/scripts/linux/readme file in the same location.

Once installed, these services are started automatically when you reboot. Consequently, the JasperReports Server also automatically restarts.

## 3.3 Start/Stop Apps — Mac OSX

After you complete the Mac OSX installation, you typically find JasperReports Server installed in the following location:

/Applications/jasperreports-server-<ver>

When JasperReports Server is running, you can see the names of the Java and PostgreSQL processes in the Activity Monitor.

To start JasperReports Server, locate this folder in Finder and double-click the following app:

jasperServerStart.app

To stop JasperReports Server, locate this folder in Finder and double-click the following app:

jasperServerStop.app

The Mac lists the following information in the Activity Monitor:

- java
- or
- org.apache.catalina.startup.Bootstrap
- postgres

### 3.3.1 Start/Stop Apps — Mac Dock

Using Finder, move the following apps into the Mac Dock to start, stop, and login to JasperReports Server:

- jasperServerStart.app
- jasperServerStop.app
- jasperServerLogin.app

### 3.3.2 Start/Stop JasperReports Server — Mac Terminal Shell

**To start and stop JasperReports Server using the Mac terminal shell:**

1. Open a Terminal shell (Finder > Go > Utilities > Terminal Icon).
2. Navigate to the <js-install> folder. For instance: /Applications/jasperreports-server-<ver>
3. To start PostgreSQL, Tomcat, and JasperReports Server, enter:
 

```
./ctlscript.sh start
```
4. To shutdown PostgreSQL, Tomcat, and JasperReports Server, enter:
 

```
./ctlscript.sh stop
```

5. To start and stop individual components:

```
cd <js-install>
./ctlscript.sh start|stop postgresql
./ctlscript.sh start|stop tomcat
```

## 3.4 Logging into JasperReports Server

### To log into JasperReports Server on any operating system:

1. Start JasperReports Server.
2. Open a supported browser: Firefox, Internet Explorer, Chrome, and Safari.
3. Log into JasperReports Server by entering the startup URL in your browser's address field. The URL depends upon your application server. If you installed the default, bundled Tomcat use:

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

- <hostname> is the name or IP address of the computer hosting JasperReports Server.
- 8080 is the default port number for the Apache Tomcat application server. If you used a different port when installing your application server, specify its port number instead of 8080.

The login page appears.

4. Log in using the following credentials:

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

If you installed the sample data, these additional sample end-users are also created. These users are non-administrative users who have fewer system privileges than an administrative user.

| User ID | Password | Description   |
|---------|----------|---|
| joeuser | joeuser  | Sample end-user   |
| demo    | demo     | Sample end-user for the SuperMart Dashboard demonstration |



When you complete the evaluation or testing of your JasperReports Server instance, change the administrator and superuser passwords (jasperadmin and superuser) and remove any sample end-users. Leaving the default passwords and end-users in place weakens the security of your installation.

### To log into JasperReports Server on Windows:

On Windows, you can launch the login page from the desktop of the JasperReports Server host computer by clicking **Start > All Programs > JasperReports Server > JasperReports Server Login**.

### To log into JasperReports Server on Mac OSX:

On Mac OSX, you can launch the login page by going to Finder and clicking the following script:

`/Applications/<js-install>/jasperServerLogin`

For example: `/Applications/jasperreports-server-<ver>/jasperServerLogin`

### To use the Dock to log into JasperReports Server:

From Finder, you can drag the `/Applications/<js-install>/jasperServerLogin.app` to the Dock to handle logging into JasperReports Server using your default system browser.

## 3.5 JasperReports Server Log Files

Log files contain important information about JasperReports Server operations. If your application server is Tomcat, JBoss, or GlassFish, the log output goes to one of the following files:

Tomcat: <tomcat>/webapps/jasperserver-pro/WEB-INF/logs/jasperserver.log

JBoss: <jboss>/server/default/deploy/jasperserver-pro.war/WEB-INF/logs/jasperserver.log

GlassFish: <glassfish>/domains/domain1/autodeploy/jasperserver-pro.war/WEB-INF/logs/jasperserver.log

You can configure the log outputs and logging levels in the log4j.properties file in the WEB-INF folder.

### **To change the logging levels while you are running JasperReports Server:**

1. Browse to [http://<hostname>:8080/jasperserver-pro/log\\_settings.html](http://<hostname>:8080/jasperserver-pro/log_settings.html).

The Log Settings page appears.

2. Change logging levels using the drop-down menus.

Changes to logging levels affect only the current session of JasperReports Server. Logging levels revert to default settings as defined in the properties files at the next startup.

For more information about system logging, see the *JasperReports Server Administrator Guide*.



## CHAPTER 4 UNINSTALLING JASPERREPORTS SERVER

---

This chapter contains the following sections:

- [Windows](#)
- [Linux](#)
- [Mac OSX](#)
- [Uninstall Survey](#)

### 4.1 Windows

**To uninstall JasperReports Server on Windows 7:**

Click **Start > All Programs > JasperReports Server > Uninstall JasperReports Server**.

**To uninstall JasperReports Server on Windows XP:**

1. Click **Start > All Programs > JasperReports Server > Uninstall JasperReports Server**.  
Typically a popup window asks which user account to run as.
2. Uncheck the check box that says:  
Protect my computer and data from unauthorized program activity



The uninstaller will not execute if **Protect my computer and data from unauthorized program activity** is checked.

### 4.2 Linux

Under Linux, the `<js-install>` folder includes an executable that removes JasperReports Server from the host.

**To uninstall JasperReports Server:**

1. From the command line, log in as the root user (or any user with sufficient privileges).
2. Enter the following commands:  

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

3. Respond Y or yes to the prompt that asks if you want to remove JasperReports Server from this computer.

## 4.3 Mac OSX

### To use Finder to uninstall JasperReports Server:

1. Navigate to the <js-install> folder.  
For example: /Applications/jasperreports-server-<ver>
2. Click the uninstall.app to launch the uninstaller.

## 4.4 Uninstall Survey

After running the uninstaller, you are prompted to take an uninstall survey from Jaspersoft. Survey answers are anonymous and help Jaspersoft improve the products we make. When you click **Yes**, the survey launches on the Jaspersoft web site in a new browser window. Select all the reasons that led you to uninstall JasperReports Server, or enter a short explanation if none match. Thank you for your feedback.



## CHAPTER 5 INSTALLING THE WAR FILE DISTRIBUTION

In addition to the installer binaries, you can install the JasperReports Server application using the stand-alone WAR file distribution. For production environments, use the WAR file distribution. Download the WAR file distribution from [Jaspersoft technical support](#), or contact your sales representative. The WAR file distribution comes in a file named jasperreports-server-5.5-bin.zip in compressed ZIP format.

This chapter contains the following sections:

- [Applications Supported by the WAR File Distribution](#)
- [Installing the WAR File Using js-install Scripts](#)
- [Additional Steps for Using DB2 and js-install Scripts](#)
- [Starting JasperReports Server](#)
- [Logging into JasperReports Server](#)
- [Troubleshooting Your JasperReports Server Configuration](#)
- [Installing the WAR File Manually](#)

### 5.1 Applications Supported by the WAR File Distribution

#### 5.1.1 Database and Application Server Support

The instructions in this and subsequent chapters support the following configurations:

| Database                   | Application Server                  | Instructions Located In  |
|----------------------------|-------------------------------------|--|
| PostgreSQL<br>MySQL<br>DB2 | Apache Tomcat<br>JBoss<br>GlassFish | This chapter.  |
| Oracle                     | WebSphere                           | <a href="#">Chapter 7, “Installing the WAR File for WebSphere,” on page 63</a> |
| SQL Server                 | WebLogic                            | <a href="#">Chapter 8, “Installing the WAR File for WebLogic,” on page 77</a>  |

For version information about these databases and application servers refer to the release notes in the root of the unpacked distribution ZIP.

## 5.1.2 Operating System Support for Bash Shell

JasperReports Server is a Java Web Application. Therefore, it supports all operating system platforms where Java is fully supported. However, for the js-install shell scripts (described in the section below), the default shell required is the bash shell. Here is a list of shells required:

| Operating System | Required Shell for js-install scripts | System Default Shell   | Script to Run  |
|------------------|---------------------------------------|------------------------|----------------|
| Windows          | CMD shell                             | CMD shell              | js-install.bat |
| Linux            | Bash shell                            | Bash shell             | js-install.sh  |
| Solaris          | Bash shell                            | Korn shell (ksh)       | js-install.sh  |
| IBM AIX          | Bash shell                            | Korn shell (ksh)       | js-install.sh  |
| HP UX            | Bash shell                            | Posix shell (posix/sh) | js-install.sh  |
| FreeBSD          | Bash shell                            | C shell (tcsh)         | js-install.sh  |

## 5.2 Installing the WAR File Using js-install Scripts

Follow the steps in this procedure to install JasperReports Server using WAR file distribution. The js-install shell scripts, supported on Windows, Linux, and Mac, do most of the work for you.

### Prerequisites for installing the WAR file:

1. Install the Oracle/Sun Java JDK 1.6 or 1.7. OpenJDK 1.6 has also been certified.
2. Create and set the JAVA\_HOME system environment variable to point to the Java JDK location.
3. Locate or install one of the following application servers:
  - Apache Tomcat 5.5, 6, or 7
  - JBoss 5.1 or 7.1
  - Glassfish 2.1 or 3.0 using the default domain (domain1)

If you use GlassFish 3.1.0 or a custom domain, see section [A.9.6, “GlassFish Modifications,” on page 155](#).
4. Locate or install the PostgreSQL, MySQL, Oracle, SQL Server, or DB2 database. If you use DB2, follow the steps in section [5.3, “Additional Steps for Using DB2 and js-install Scripts,” on page 36](#).



The target database can be on a remote server. The application server should reside on the local machine.

If you would like to run a pre-install validation test, you can run `js-install.bat test` or a similar command. For more information about how to perform a validation test in your environment, see section [5.6.3.1, “js-install Script Test Mode,” on page 39](#).

### To install the WAR file using js-install scripts:

The scripts are intended for the bash shell.



If installing to non-linux Unix platforms such as HP-UX, IBM AIX, FreeBSD, or Solaris the bash shell is required for using the js-install scripts.

1. Extract all files from `jasperreports-server-5.5-bin.zip`. Choose a destination, such as `C:\Jaspersoft` on Windows, `/home/<user>` on Linux, or `/Users/<user>` on Mac.  
The directory, `jasperreports-server-5.5-bin`, appears in the file location you choose.
2. Copy the `<database>_master.properties` file for your database from `sample_conf` and paste it to `buildomatic`:
  - Copy from — `<js-install>/buildomatic/sample_conf/`

- ♦ Paste to — `<js-install>/buildomatic`

For example, if your database is PostgreSQL, copy `postgresql_master.properties` to `<js-install>/buildomatic`.

3. Rename the file you copied to `default_master.properties`.
4. Edit the `default_master.properties` file to add the settings for your database and application server. [Table 5-1](#) lists sample property values for each supported database.

**Table 5-1 Sample Values for the default\_master.properties File**

| Database   | Sample Property Values  |
|------------|---|
| PostgreSQL | <pre>appServerType=tomcat6 [tomcat7, tomcat5, jboss, jboss-eap-6, jboss-as-7, glassfish2, glassfish3, skipAppServerCheck†] appServerDir=c:\\Program Files\\Apache Software Foundation\\Tomcat 7† dbHost=localhost dbUsername=postgres dbPassword=postgres</pre>   |
| MySQL      | <pre>appServerType=tomcat6 [tomcat7, tomcat5, jboss, jboss-eap-6, jboss-as-7, glassfish2, glassfish3, skipAppServerCheck†] appServerDir=c:\\Program Files\\Apache Software Foundation\\Tomcat 7† dbUsername=root dbPassword=password dbHost=localhost</pre>   |
| Oracle*    | <pre>appServerType=tomcat6 [tomcat7, tomcat5, jboss, jboss-eap-6, jboss-as-7, glassfish2, glassfish3, skipAppServerCheck†] appServerDir=c:\\Program Files\\Apache Software Foundation\\Tomcat 7† sysUsername=system sysPassword=password dbUsername=jasperserver dbPassword=password dbHost=hostname</pre>  |
| DB2        | <pre>appServerType=tomcat6 [tomcat7, tomcat5, jboss, jboss-eap-6, jboss-as-7, glassfish2, glassfish3, skipAppServerCheck†] appServerDir=c:\\Program Files\\Apache Software Foundation\\Tomcat 7† dbUsername=db2admin dbPassword=password dbHost=localhost</pre> <p>If you use DB2, follow the steps in <a href="#">section 5.3, “Additional Steps for Using DB2 and js-install Scripts,” on page 36</a></p> <p>For DB2 8.x, change your deployed JDBC driver as described in <a href="#">section 6.4.3.1, “DB2 8.x Driver Example,” on page 54</a>.</p> |
| SQL Server | <pre>appServerType=tomcat6 [tomcat7, tomcat5, jboss, jboss-eap-6, jboss-as-7, glassfish2, glassfish3, skipAppServerCheck†] appServerDir=c:\\Program Files\\Apache Software Foundation\\Tomcat 7† dbUsername=sa dbPassword=sa dbHost=localhost</pre>   |

\* The dbUsername must be the same as the Oracle user name. In addition, buildomatic will not work with the “sys as sysdba” syntax.

- † When the property `appServerType` is set to `skipAppServerCheck`, `buildomatic` skips any application server validation.
- ‡ Backslashes in paths must be doubled in properties files, for example `appServerDir=c:\Apache Software Foundation\Tomcat 7`.



If, under Linux, Tomcat is installed using `apt-get`, `yum`, or `rpm`, see section [A.9.5, “Tomcat 6 Installed Using apt-get,”](#) on page 155.

## 5. Password Encryption

The `default_master.properties` file now has a property setting to enable encryption of passwords that reside on the file system. This will apply to all files found under the `buildomatic` folder. As well as the connection pooling file used by Apache Tomcat (i.e. `context.xml`). Currently, password encryption support for connection pooling only supports the Tomcat application server.

To enable encryption on the file system, uncomment the `encrypt` property so that it looks like the following:

```
encrypt=true
```



For more information of the Encryption functionality, refer to the *JasperReports Server Administrator Guide*.

## 6. Run the `js-install` scripts:

- a. Start your database server.
- b. Stop your application server.
- c. Open Command Prompt as Administrator on Windows or open a terminal window on Linux and Mac OSX.
- d. Run the `js-install` script:

| Commands  | Description  |
|---|--|
| <code>cd &lt;js-install&gt;/buildomatic</code>  |  |
| <code>js-install.bat</code> (Windows)<br><code>./js-install.sh</code> (Linux and Mac OSX)                 | Installs JasperReports Server, sample data, and sample databases (foodmart and sugarcrm) |
| <code>js-install.bat minimal</code> (Windows)<br><code>./js-install.sh minimal</code> (Linux and Mac OSX) | Installs JasperReports Server, but does not install sample data and sample databases     |

If you encounter errors during the `js-install` script execution, see section [5.6.3, “Error Running js-install Scripts \(js-install.bat/sh\),”](#) on page 38.

7. Set Java JVM Options, as described in [6.1, “Setting JVM Options for Application Servers,”](#) on page 41. This step is required.
8. Set up the JasperReports Server License as described in [6.2, “Setting Up the JasperReports Server License,”](#) on page 45. This step is required.



To view the output log, look in: `<js-install>/buildomatic/logs/js-install-<date>.log`

## 5.3 Additional Steps for Using DB2 and `js-install` Scripts

The `buildomatic` scripts cannot automatically connect to a remote DB2 database and carry out Admin operations, so you have to perform additional steps to create the databases.

The DB2 client software, `db2` or `db2cmd`, can be used to interact with DB2.

1. Enter commands similar to the ones below in the DB2 command window to create and initialize the repository database, called jsprsrvr in DB2 to conform to the 8-character limitation:

```
db2 create database jsprsrvr using codeset utf-8 territory us pagesize 16384
```

2. (Optional) Run the following commands in the DB2 command window if you want to install sample databases:

```
db2 create database sugarcrm
db2 create database foodmart
```

3. Now, continue installing JasperReports Server as described in section 5.2, “Installing the WAR File Using js-install Scripts,” on page 34.

Further considerations:

- If you are using DB2 8.1, set the LOGFIL\_SIZ parameter to a minimum of 3000 to avoid possible log file errors while loading the foodmart database. Configure your foodmart database right after creating it by using Control Center.
- If JasperReports Server is deployed on the same host as DB2, delete the following file to avoid conflicts:  
<db2>/SQLLIB/java/db2jcc.jar

## 5.4 Starting JasperReports Server

**To run JasperReports Server:**

Start your application server using one of these commands:

|            |                             |                              |
|------------|-----------------------------|------------------------------|
| Tomcat:    | Windows                     | <tomcat>/bin/startup.bat     |
|            | Linux and Mac OSX           | <tomcat>/bin/startup.sh      |
| JBoss:     | Windows                     | <jboss>/bin/run.bat          |
|            | Linux and Mac OSX           | <jboss>/bin/run.sh           |
| GlassFish: | Windows, Linux, and Mac OSX | asadmin start-domain domain1 |

To view the JasperReports Server application logs, see section 3.5, “JasperReports Server Log Files,” on page 29.

## 5.5 Logging into JasperReports Server

After JasperReports Server starts up, login by going to this URL:

http://<hostname>:8080/jasperserver-pro

Example:

http://localhost:8080/jasperserver-pro

http://jasperserver.example.com:8080/jasperserver-pro

The login page appears after compiling the necessary JSP files (this will take a few moments).

Use one of the following sets of credentials to log into JasperReports Server:

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

If you logged in successfully, your JasperReports Server home page appears.



When you complete the evaluation or testing of your JasperReports Server instance, change the administrator and superuser passwords (jasperadmin and superuser) and remove any sample end-users. Leaving the default passwords and end-users in place weakens the security of your installation.

Refer to the *JasperReports Server User Guide* to begin adding reports and other objects to the server.

## 5.5.1 JasperReports Server Heartbeat

After initially logging into JasperReports Server, you are asked to opt-in to the JasperReports Server Heartbeat.

To opt-in, click **OK**. To opt-out, click the check box to remove the check and click **OK**.

The heartbeat helps Jaspersoft understand customer installation environments to improve our products. If you choose to enable the heartbeat, an HTTPS call at server startup time sends information like this to Jaspersoft:

- Operating System and JVM type and version
- Application Server and Database type and version
- JasperReports Server type and version
- Unique, anonymous identifier value

You can manually enable or disable the heartbeat by modifying the following property file `jasperserver-pro/WEB-INF/js.config.properties`. To disable the heartbeat, set the `heartbeat.enabled` property to `false`:

```
heartbeat.enabled=false
```

For additional information about enabling and disabling the heartbeat component, see the *JasperReports Server Administrator Guide*.

## 5.6 Troubleshooting Your JasperReports Server Configuration

This section describes the most common installation problems.

### 5.6.1 JasperReports Server Startup Problems

If you encounter a problem trying to run a new JasperReports Server, an incorrect database configuration is the likely culprit. Another common cause is a mistake in the application server configuration files. For information about resolving these types of errors, see [Appendix A, “Troubleshooting,” on page 145](#).

### 5.6.2 Error Running a Report

If you have trouble running reports in your new JasperReports Server instance, see section [A.8.11, “Error Running a Report”](#) in [Appendix A, “Troubleshooting,” on page 145](#).

### 5.6.3 Error Running js-install Scripts (js-install.bat/sh)

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

**To troubleshoot problems running js-install scripts:**

1. Open the output log file located in:  
`<js-install>/buildomatic/logs/js-install-<date>-<number>.log`
2. Try to find the first error encountered by the `js-install` steps.
  - Go to the end of the output log.
  - Scroll back through lines of error messages until you find the first error logged. Typically, this error causes more errors later in the log.

- ♦ Finding the original error is the way to understand the problem. However, this can often be tricky because Java stack traces in conjunction with the Spring application component framework can make the error output quite long.
3. Incorrect settings in the `default_master.properties` file cause most problems, which you can correct by editing your `default_master.properties` settings. Common errors are:
    - ♦ Typos in the path for the application server
    - ♦ Misspelling the hostname or password for the database

**To recreate your `default_master.properties` settings:**

1. Open the file `<js-install>/buildomatic/default_master.properties`, make corrections, and save it.
2. Re-run the `js-install` script.

The `js-install` script uses the current values in the `default_master.properties` file.

To help isolate errors, run the `js-install` scripts in test mode.

### 5.6.3.1 js-install Script Test Mode

You can run the `js-install` and `js-upgrade` scripts in test mode using the `test` option. In test mode, the `js-install` scripts check your `default_master.properties` settings and validate the application server location and connection to the specified database. Using test mode can help debug issues, such as an incorrect database password. Your system isn't altered when executing the script in test mode.

**To run the `js-install` script in test mode on Windows:**

1. Navigate to the `buildomatic` directory:
 

```
cd <js-install>/buildomatic
```
2. Enter the following command to run the `js-install` script in test mode:
 

```
js-install.bat test
```

**To run the `js-install` script in test mode on Linux or Mac OSX:**

1. Navigate to the `buildomatic` directory:
 

```
cd <js-install>/buildomatic
```
2. Enter the following command to run the `js-install` script in test mode:
 

```
./js-install.sh test
```

### 5.6.4 Problem Connecting to a Cloud Database Instance

A cloud database instance (such as Amazon EC2) typically disables unused IP ports. When the `js-install` script runs, it validates the database hostname using the built-in ant operation `<isreachable>`. This operation is similar to a network ping and may cause a “hang” issue if the port is unavailable. In this case, the `validateHost` step can be commented out in the `buildomatic/validation.xml` file. See the comment in the `do-pre-install-test` target.

## 5.7 Installing the WAR File Manually

In some case, you may need to install the WAR file manually when you cannot use the `js-install` scripts.

The manual buildomatic steps described in this procedure execute the same Ant targets as the `js-install` scripts (`js-install.sh/.bat`). The procedure shows which buildomatic targets to execute manually if, for some reason, you are unable to use the `js-install` scripts.

**To install the WAR file distribution using manual buildomatic steps:**

1. Start your database server.
2. Stop your application server.
3. Create and edit a `default_master.properties` file to add the settings in for your database and application server as described in [5.2, “Installing the WAR File Using `js-install` Scripts,” on page 34](#).

- Open a Command Prompt as Administrator on Windows or open a terminal window on Linux or Mac. Run the following commands:

**Table 5-2 Buildomatic Targets to Execute to Install the WAR File**

| Commands   | Description   |
|--|---|
| <code>cd &lt;js-install&gt;/buildomatic</code>                                   | Makes the buildomatic directory your current directory.   |
| <code>js-ant create-js-db</code>   | Creates the JasperReports Server repository database.   |
| <code>js-ant create-sugarcrm-db</code><br><code>js-ant create-foodmart-db</code> | (Optional) Creates the sample databases.  |
| <code>js-ant load-sugarcrm-db</code><br><code>js-ant load-foodmart-db</code>     | (Optional) Loads sample data into the sample databases.   |
| <code>js-ant init-js-db-pro</code><br><code>js-ant import-minimal-pro</code>     | Initializes the <code>jasperserver</code> database, loads core application data. Running <code>js-ant import-minimal-pro</code> is mandatory. The server cannot function without this data. |
| <code>js-ant import-sample-data-pro</code>                                       | (Optional) Loads the demos that use the sample data.  |
| <code>js-ant deploy-webapp-pro</code>  | Configures and deploys the WAR file to Tomcat, JBoss, or Glassfish.   |



On non-Linux Unix platforms, the `js-ant` commands may not be compatible with all shells. If you have errors, use the `bash` shell explicitly. For more information, see section [A.4, “Bash Shell for Solaris, IBM AIX, HP UX and FreeBSD,”](#) on page 147.

If you encounter an error when running `create-sugarcrm-db`, `create-foodmart-db`, or `create-js-db`, you can create the JasperReports Server database manually using the database administration tool for your particular database type. To create the JasperReports Server database manually for PostgreSQL, MySQL, Oracle, SQL Server, and DB2, see section [6.3, “Manually Creating the JasperReports Server Database,”](#) on page 48.

If you have previously installed the databases, you can drop the old versions and then recreate the databases. To do this, run the following drop commands before running the commands in [Table 5-2](#):

**Table 5-3 Buildomatic Targets to Execute to Delete Sample Databases**

| Commands   | Description   |
|--|---|
| <code>js-ant drop-sugarcrm-db</code><br><code>js-ant drop-foodmart-db</code> | (Optional) Deletes the sample databases.  |
| <code>js-ant drop-js-db</code>   | (WARNING) This will delete the JasperReports Server repository database. Only run this command if you intend to recreate the <code>jasperserver</code> database |

- Set Java JVM Options, as described in section [6.1, “Setting JVM Options for Application Servers,”](#) on page 41. This step is required.
- Set up the JasperReports Server License as described in section [6.2, “Setting Up the JasperReports Server License,”](#) on page 45. This step is required.



## CHAPTER 6 JVM OPTIONS, LICENSE SETUP, MANUAL DATABASE CREATION

---

This chapter contains the following sections:

- [Setting JVM Options for Application Servers](#)
- [Setting Up the JasperReports Server License](#)
- [Setting JVM Options for Application Servers](#)
- [Locating and Changing Buildomatic Configuration Files](#)
- [Configuring Report Scheduling](#)
- [Updating XML/A Connection Definitions](#)

### 6.1 Setting JVM Options for Application Servers

JasperReports Server is supported on Java 1.6 and 1.7. Java Virtual Machine (JVM) runtime parameters normally need to be explicitly set so that the memory settings have values that are larger than the default settings. The options that you should set and the values they are set to depend on your version of Java and the application server that you use.

The settings in this section apply specifically to the Oracle/Sun JVM. Other JVMs may or may not have equivalent settings.



For the Oracle database, setting the Oracle localization option, `defaultNChar`, can substantially impact the performance of JDBC queries. If you do not need to support UTF-8 for your Oracle database, you can omit this setting.

#### 6.1.1 Tomcat and JBoss JVM Options

The following tables present some typical settings of JVM options that affect JasperReports Server. For information about changing a JVM option setting for your particular environment, see your application server documentation.



The following example settings are for 64-bit systems. For 32-bit systems, see [“Checking your Java JVM Options” on page 24](#).

| JVM Options on Windows (64 bit)               |   |
|---|---|
| Options for Java 1.6 and 1.7                  | set JAVA_OPTS=%JAVA_OPTS% -Xms1024m -Xmx2048m -XX:PermSize=32m<br>set JAVA_OPTS=%JAVA_OPTS% -XX:MaxPermSize=512m -Xss2m -XX:+UseConcMarkSweepGC<br>set JAVA_OPTS=%JAVA_OPTS% -XX:+CMSClassUnloadingEnabled  |
| For Oracle                                    | set JAVA_OPTS=%JAVA_OPTS% -Doracle.jdbc.defaultNChar=true   |
| Additional options for Java 1.6-1.7 and JBoss | set JAVA_OPTS=%JAVA_OPTS% -Djavax.xml.soap.MessageFactory=org.apache.axis.soap.MessageFactoryImpl<br>set JAVA_OPTS=%JAVA_OPTS% -Djavax.xml.soap.SOAPConnectionFactory=org.apache.axis.soap.SOAPConnectionFactoryImpl<br>set JAVA_OPTS=%JAVA_OPTS% -Djavax.xml.soap.SOAPFactory=org.apache.axis.soap.SOAPFactoryImpl |
| Additional option for JBoss 5.1               | set JAVA_OPTS=%JAVA_OPTS% -Djavax.xml.transform.TransformerFactory=org.apache.xalan.processor.TransformerFactoryImpl  |

JasperReports Server doesn't provide a virtual X frame buffer on Linux. If your Linux applications are graphical, set the `-Djava.awt.headless=true` to prevent Java from trying to connect to an X-Server for image processing.

| JVM Options on Linux and Mac OSX (64 bit)     |  |
|---|--|
| Options for Java 1.6 and 1.7                  | export JAVA_OPTS="\$JAVA_OPTS -Xms1024m -Xmx2048m -XX:PermSize=32m"<br>export JAVA_OPTS="\$JAVA_OPTS -XX:MaxPermSize=512m -Xss2m"<br>export JAVA_OPTS="\$JAVA_OPTS -XX:+UseConcMarkSweepGC"<br>export JAVA_OPTS="\$JAVA_OPTS -XX:+CMSClassUnloadingEnabled"  |
| For Oracle                                    | export JAVA_OPTS="\$JAVA_OPTS -Doracle.jdbc.defaultNChar=true"   |
| Additional options for Java 1.6-1.7 and JBoss | export JAVA_OPTS="\$JAVA_OPTS -Djavax.xml.soap.MessageFactory=org.apache.axis.soap.MessageFactoryImpl"<br>export JAVA_OPTS="\$JAVA_OPTS -Djavax.xml.soap.SOAPConnectionFactory=org.apache.axis.soap.SOAPConnectionFactoryImpl"<br>export JAVA_OPTS="\$JAVA_OPTS -Djavax.xml.soap.SOAPFactory=org.apache.axis.soap.SOAPFactoryImpl" |
| Additional option for JBoss 5.1               | export JAVA_OPTS="\$JAVA_OPTS -Djavax.xml.transform.TransformerFactory=org.apache.xalan.processor.TransformerFactoryImpl"  |

There are a number of ways to set JVM options. Sections [6.1.2](#) - [6.1.5](#) present step-by-step instructions for performing this task. Alternatively, you can add your `JAVA_OPTS` settings to any one of these files:

| File                                 | Add JVM Options After This Line on Windows          |
|--------------------------------------|---|
| <tomcat>/bin/setclasspath.bat        | set JAVA_ENDORSED_DIRS=%BASEDIR%\common\endorsed    |
| <tomcat>/bin/setenv.bat              | JAVA_OPTS setting can go anywhere in this file.     |
| <jboss>/bin/run.bat (JBoss 5.1)      | set JAVA_OPTS=%JAVA_OPTS% -Dprogram.name=%PROGNAME% |
| <jboss>/bin/standalone.bat (JBoss 7) | rem Setup JBoss specific properties                 |

| File                         | Add JVM Options After This Line on Linux        |
|------------------------------|---|
| <tomcat>/bin/setclasspath.sh | JAVA_ENDORSED_DIRS="\$BASEDIR"/common/endorsed  |
| <tomcat>/bin/setenv.sh       | JAVA_OPTS setting can go anywhere in this file. |

| File                                | Add JVM Options After This Line on Linux                     |
|-------------------------------------|--|
| <jboss>/bin/run.sh (JBoss 5.1)      | export JAVA_OPTS="\$JAVA_OPTS -Dprogram.name=\$PROGNAME"     |
| <jboss>/bin/standalone.sh (JBoss 7) | Add JAVA_OPTS setting <b>before</b> #Display our environment |

### 6.1.2 Changing JVM Options for Bundled Tomcat as a Windows Service

The Windows binary installer installs the bundled Tomcat component as a Windows Service by default. The steps to change JVM options are:

1. Open this file for editing:

```
cd <js-install>/apache-tomcat/bin/service.bat
```

2. Look for the following line to change the JVM heap size, for example:

```
"%EXECUTABLE%" //US//%SERVICE_NAME% --Startup auto --JvmOptions "-Xms1024M;-Xmx2048M;-Xss2M;-Dcatalina.base=%CATALINA_BASE%;-Dcatalina.home=%CATALINA_HOME%;-Djava.endorsed.dirs=%CATALINA_HOME%\endorsed" --StartMode jvm --StopMode jvm
```

3. Update this line to increase the maximum heap size from 2048M to 3072M, for example:

```
-Xmx3072M
```

4. Because Tomcat is installed as a service, you need to re-install the service. From a Windows Command shell, enter these commands (Note: the cmd shell will disappear when these commands are run. You need to open a new cmd shell for each command.). To open a cmd shell: Start Menu > Run... > cmd

```
cd <js-install>\apache-tomcat\scripts
serviceinstall.bat REMOVE
serviceinstall.bat INSTALL
```

The Tomcat service is removed and then installed. After execution of the commands, the service is running.

### 6.1.3 Changing JVM Options for Existing Tomcat as a Windows Service

If you installed JasperReports Server to use an existing Tomcat (not the bundled component) that is running as a Windows service, you can set Java options on the Java Tab of the Tomcat Properties dialog:

1. Launch the Tomcat configuration application from the Windows Start menu:  
**Start > Programs > Apache Tomcat > Configure Tomcat (Run as administrator)**
2. In the Apache Tomcat Properties dialog, click the **Java** tab.
3. In the Java Options field, add your JAVA\_OPTS values according to the tables above.  
Enter only the options preceded by -X or -D, not set JAVA\_OPTS=%JAVA\_OPTS%.  
Enter only one Java option setting per line.
4. For instance, add options as follows:



These example settings are for 64-bit systems. For 32-bit systems, see [“Checking your Java JVM Options” on page 24](#).

```
-Xms1024m
-Xmx2048m
-XX:PermSize=32m
-XX:MaxPermSize=512m
-Xss2m
```

5. Click Apply, then click OK.

6. Stop and restart Tomcat.

### 6.1.4 Changing JVM Options for Bundled Tomcat on Linux

If, under Linux, you installed JasperReports Server to use the bundled Tomcat, you can set Java options by editing the appropriate Tomcat configuration script. The steps to change JVM options are:

1. Open the following file for editing:

```
cd <js-install>/apache-tomcat/scripts/ctl.sh
```

2. Look for the `start_tomcat()` function and locate the `JAVA_OPTS` variable inside it.
3. Modify the `JAVA_OPTS` values according to the tables above:

```
start_tomcat() {  
    is_tomcat_running  
    ...  
    export JAVA_OPTS="-Xms1024m -Xmx2048m -XX:PermSize=32m -XX:MaxPermSize=512m -  
Xss2m -XX:+UseConcMarkSweepGC -XX:+CMSClassUnloadingEnabled"  
    ...  
}
```



There may be more than one occurrence of the `Java_OPTS` variable in the `ctl.sh` file. Make sure to edit the instance inside the `start_tomcat()` function.

4. Save and close the `ctl.sh` file.
5. Stop and restart PostgreSQL and Tomcat as described in [Chapter 3, “Starting and Stopping JasperReports Server,” on page 25](#).

### 6.1.5 Changing GlassFish JVM Options

The following sections describe how to set the JVM options for GlassFish for Java 1.6 and 1.7 using the command line or a configuration file.

#### 6.1.5.1 Setting GlassFish JVM Options with `asadmin` Command

1. First make sure your GlassFish instance is up and running, then enter the following command as a single line:

```
asadmin create-jvm-options -Xms1024m:-Xmx2048m:-XX\:PermSize=32m:  
-XX\:MaxPermSize=512m:-Xss2m:-XX\:+UseConcMarkSweepGC:  
-XX\:+CMSClassUnloadingEnabled:  
-Djavax.xml.soap.MessageFactory=org.apache.axis.soap.MessageFactoryImpl:  
-Djavax.xml.soap.SOAPConnectionFactory=org.apache.axis.soap.SOAPConnectionFactoryImpl:  
-Djavax.xml.soap.SOAPFactory=org.apache.axis.soap.SOAPFactoryImpl:  
-Doracle.jdbc.defaultNChar=true
```

If you are not using an Oracle database, you can omit the last option in example above.

2. Restart the application server using the following commands:

```
asadmin stop-domain domain1  
asadmin start-domain domain1
```

When running the `asadmin create-jvm-options` command, error messages like this might appear:

```
[exec] CLI167 Could not create the following jvm options. Options exist:  
[exec] -Xmx512m  
[exec] CLI137 Command create-jvm-options failed.
```

This message indicates that one of the options specified was already set in the JVM. The command will succeed for all other JVM options on the command line. No further action is necessary.

### 6.1.5.2 Setting GlassFish JVM Options by Editing domain.xml

1. Open the <glassfish>/domains/domain1/config/domain.xml configuration file for editing.
2. Add the following lines to the section entitled java-config:

```
<jvm-options>-Xms1024m -Xmx2048m -XX:PermSize=32m -XX:MaxPermSize=512m -Xss2
-XX:+UseConcMarkSweepGC -XX:+CMSClassUnloadingEnabled
-Djavax.xml.soap.MessageFactory=org.apache.axis.soap.MessageFactoryImpl
-Djavax.xml.soap.SOAPConnectionFactory=org.apache.axis.soap.SOAPConnectionFactoryImpl
-Djavax.xml.soap.SOAPFactory=org.apache.axis.soap.SOAPFactoryImpl
-Doracle.jdbc.defaultNChar=true
</jvm-options>
```

If you are not using an Oracle database, you can omit the last option in example above.

3. If you are modifying the settings for a running instance of GlassFish, restart the application server using the following commands:

```
asadmin stop-domain domain1
asadmin start-domain domain1
```

## 6.2 Setting Up the JasperReports Server License

JasperReports Server requires a license to run and comes with an evaluation license that is valid for 30 days. Please contact Technical Support or your sales representative to get your commercial license.

The license file is in the following location:

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

The license file specifies the terms of your license, such as the following:

- Expiration date, number of users, and/or number of CPUs
- Also, features licensed separately from the basic commercial license, such as multi-tenancy

Jaspersoft receives information about your system periodically. The information is used only to monitor compliance with your license. No personal information is collected or transmitted.

### 6.2.1 Default License Configuration for All Application Servers

At startup time, JasperReports Server automatically looks for the jasperserver.license file in the home directory of the system user who is running the application server. **Table 6-1** lists the application server user home directories for supported operating systems.

To configure the license:

1. Stop the application server.
2. Copy the jasperserver.license file in <js-install> to the directory for your operating system:

**Table 6-1 License Locations**

| Operating System                  | Application Server User Home Directory |
|-----------------------------------|--|
| Linux                             | /home/<user>/                          |
| Mac OSX                           | /Users/<user>/                         |
| Windows 7 installed from WAR file | C:\Users\<user>\                       |

**Table 6-1 License Locations**

| Operating System                              | Application Server User Home Directory |
|---|--|
| Windows 7 installed from the binary installer | C:\Users\                              |
| Windows XP                                    | C:\Documents and Settings\<user>\      |
| Windows 2003                                  | C:\Documents and Settings\<user>\      |
| Windows 2008                                  | C:\Documents and Settings\<user>\      |

## 6.2.2 User-Defined License Location

If you prefer to put your license in another directory, modify your application server startup script to set a `JAVA_OPT` value to explicitly point to that directory.

### 6.2.2.1 Alternate License Setup for Tomcat

If your license is not located in the home directory of the application server user, you can set a `JAVA_OPT` value to explicitly point to your license.

On Windows:

1. In the file `<tomcat>/bin/setclasspath.bat`, locate the following line:

```
set JAVA_ENDORSED_DIRS=%BASEDIR%\common\endorsed
```

Alternatively, create an empty file called `<tomcat>/bin/setenv.bat`.

2. Below that line or in the new file, insert the following line:

```
set JAVA_OPTS=%JAVA_OPTS% -Djs.license.directory="<js-install>"
```

For example:

```
set JAVA_OPTS=%JAVA_OPTS% -Djs.license.directory="C:\jasperserver-bin"
```

On Linux and Mac OSX:

1. In the file `<tomcat>/bin/setclasspath.sh`, locate the following line:

```
JAVA_ENDORSED_DIRS="$BASEDIR"/common/endorsed
```

Alternatively, create an empty file called `<tomcat>/bin/setenv.sh`.

2. Below that line or in the new file, insert the following line:

```
export JAVA_OPTS="$JAVA_OPTS -Djs.license.directory=<js-install>"
```

For example:

```
export JAVA_OPTS="$JAVA_OPTS -Djs.license.directory=/home/user/jasperserver-bin"
```

### 6.2.2.2 Alternate License Setup for Bundled Tomcat as a Windows Service

The Windows binary installer installs the bundled Tomcat component as a Windows Service by default. The steps to specify a specific folder to hold the `jasperserver.license` are the following:

1. Open this file for editing:

```
cd <js-install>/apache-tomcat/bin/service.bat
```

2. Look for the following line (second line of two that set JVM options):

```
"%EXECUTABLE%" //US//%SERVICE_NAME% ++JvmOptions
"-Djs.license.directory=C:\Jaspersoft\jasperreports-server-5.1;
-XX:PermSize=32m;-XX:MaxPermSize=512m;-XX:+UseConcMarkSweepGC;
-XX:+CMSClassUnloadingEnabled" "-Djava.io.tmpdir=%CATALINA_BASE%\temp;
-Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager;
-Djava.util.logging.config.file=%CATALINA_BASE%\conf\logging.properties"
```

Update the line above to point to this example license location:

```
-Djs.license.directory=C:\MyLicenses
```

Because Tomcat is installed as a service, you need to re-install the service. From a Windows Command shell, enter these commands (Note: the cmd shell will disappear when these commands are run. You need to open a new cmd shell for each command.). To open a cmd shell: Start Menu > Run... > cmd:

```
cd <js-install>\apache-tomcat\scripts
serviceinstall.bat REMOVE
serviceinstall.bat INSTALL
```

The Tomcat service is removed and then installed. After execution of these commands, the service is running.

### 6.2.2.3 Alternate License Location for Existing Tomcat as a Windows Service

#### Windows XP:

Some customers have reported that running an existing Tomcat as a Windows Service under Windows XP, they needed to put the license in this location for the server to detect it by default:

```
C:\Documents and Settings\LocalService\jasperserver.license
```

This location can be unusual because the LocalService folder is not displayed when you click on the Documents and Settings folder in Windows Explorer. If you go to the Explorer Address Bar and type LocalService, it is displayed. You can copy the jasperserver.license file to this location if your Tomcat as a Windows Service is unable to find the license in the Documents and Settings/<user> location described in the previous sections.

#### Windows 7:

If you have an existing Tomcat as a Windows Service under Windows 7, copy your license to the root of the C: drive. This is the home folder for the SYSTEM user. The location is:

```
C:\jasperserver.license
```

### 6.2.2.4 Alternate License Setup for JBoss

If your license will not be located in the home directory of the application server user, you can set a JAVA\_OPT value to explicitly point to your license.

#### On Windows:

1. In the file <jboss>/bin/run.bat, locate the following line:

```
set JAVA_OPTS=%JAVA_OPTS% -Dprogram.name=%PROGNAME%
```

2. Below that line, insert the following line:

```
set JAVA_OPTS=%JAVA_OPTS% -Djs.license.directory="<js-install>"
```

For example:

```
set JAVA_OPTS=%JAVA_OPTS% -Djs.license.directory="C:\jasperserver-bin"
```

#### On Linux and Mac OSX:

1. In the file <jboss>/bin/run.sh, locate the following line:

```
export JAVA_OPTS="$JAVA_OPTS -Dprogram.name=$PROGNAME"
```

2. Below that line, insert the following line:

```
export JAVA_OPTS="$JAVA_OPTS -Djs.license.directory=<js-install>"
```

For example:

```
export JAVA_OPTS="$JAVA_OPTS -Djs.license.directory=/home/user/jasperserver-bin"
```

## 6.3 Manually Creating the JasperReports Server Database

If you can't use the `js-install` scripts to create the JasperReports Server database and the sample databases, you can create them manually.

### To manually create the JasperReports Server database:

1. Follow these instructions to create the repository database and optional sample databases:

- ♦ [6.3.1, “PostgreSQL,” on page 48](#)
- ♦ [6.3.2, “MySQL,” on page 49](#)
- ♦ [6.3.3, “Oracle,” on page 50](#)
- ♦ [6.3.4, “DB2,” on page 51](#)
- ♦ [6.3.5, “SQL Server,” on page 52](#)

The commands in these sections have been tested at Jaspersoft, but the commands you need to use on your database instance may be different.

2. If you did not install the optional sample databases, complete the installation with these commands:

```
cd <js-install>/buildomatic
js-ant import-minimal-pro
js-ant deploy-webapp-pro
```

If you installed the optional sample databases, complete the installation with these commands:

```
cd <js-install>/buildomatic
js-ant import-sample-data-pro
js-ant deploy-webapp-pro
```

For more information about executing the Ant scripts, see section [5.7, “Installing the WAR File Manually,” on page 39](#).

3. Set Java JVM Options, as described in section [6.1, “Setting JVM Options for Application Servers,” on page 41](#). This step is required.
4. Set up the JasperReports Server License as described in section [6.2, “Setting Up the JasperReports Server License,” on page 45](#). This step is required.

### 6.3.1 PostgreSQL

#### To manually create the JasperReports Server database in PostgreSQL:

1. On the Windows, Linux, or Mac command line, enter these commands to create and initialize the JasperReports Server database:

```
cd <js-install>/buildomatic/install_resources/sql/postgresql

psql -U postgres -W
postgres=#create database jasperserver encoding='utf8';
postgres=#\c jasperserver;
postgres=#\i js-pro-create.ddl
postgres=#\i quartz.ddl
postgres=#\q
```



2. (Optional) Run the following commands if you want to install sample databases:

```
cd <js-install>/buildomatic/install_resources/sql/postgresql

psql -U postgres -W
postgres=#create database sugarcrm encoding='utf8';
postgres=#create database foodmart encoding='utf8';
postgres=#\c sugarcrm;
postgres=#\i sugarcrm.sql; (first make sure the file is unzipped)
postgres=#\c foodmart;
postgres=#\i foodmart-postgresql.sql; (first make sure the file is unzipped)
postgres=#\i supermart-update.sql;
postgres=#\q
```

3. Continue from [step 2 in section 6.3, "Manually Creating the JasperReports Server Database,"](#) on page 48.

### 6.3.2 MySQL

#### To manually create the JasperReports Server database in MySQL:

The MySQL client software, `mysql.exe` or `mysql`, can be used to interact with the MySQL database.



For specific details on connecting to the MySQL database and setting privileges for databases and db users, please refer to the documentation provided with your database.

1. On the Windows, Linux, or Mac command line, enter these commands to create and initialize the JasperReports Server database:

```
cd <js-install>/buildomatic/install_resources/sql/mysql

mysql -u root -p
mysql>create database jasperserver character set utf8;
mysql>use jasperserver;
mysql>source js-pro-create.ddl
mysql>source quartz.ddl
mysql>exit
```

2. (Optional) Run these commands to install sample databases:

```
cd <js-install>/buildomatic/install_resources/sql/mysql

mysql -u root -p
mysql>create database sugarcrm;
mysql>create database foodmart;
mysql>use sugarcrm;
mysql>source sugarcrm.sql; (first make sure the file is unzipped)
mysql>use foodmart;
mysql>source foodmart-mysql.sql; (first make sure the file is unzipped)
mysql>source supermart-update.sql;
mysql>exit
```

3. Continue from [step 2 in section 6.3, "Manually Creating the JasperReports Server Database,"](#) on page 48.

### 6.3.3 Oracle

#### To manually create the JasperReports Server database in Oracle:

The Oracle client software, `sqlplus.exe` or `sqlplus`, can be used to interact with Oracle.



For specific details on connecting to the Oracle database and setting privileges for databases and db users, please refer to the documentation provided with your database.

1. On the Windows, Linux, or Mac command line, enter these commands to create and initialize the JasperReports Server database:

```
cd <js-install>/buildomatic/install_resources/sql/oracle

sqlplus /nolog (start sqlplus client)
SQL> connect system/password (use your sysUsername and password)
(or SQL>connect sys/password as sysdba
SQL> create user jasperserver identified by password;
SQL> grant connect, resource to jasperserver;
SQL> connect jasperserver/password@ORCL (use your password, your SID)
SQL> @js-pro-create.ddl
SQL> @quartz.ddl
SQL> exit
```

2. (Optional) Special edit to the `sugarcrm.sql` script which creates the `sugarcrm` sample database. The `sqlplus` command line tool interpretes SQL statements differently than a JDBC call (ie the way `buildomatic` executes SQL scripts). Because of this, the `sugarcrm.sql` file must be edited in order to execute using `sqlplus`. To make these edits do the following:

- ♦ Unzip the `sugarcrm.zip` file to get the `sugarcrm.sql` file. Open `sugarcrm.sql` for editing:
- ♦ Uncomment the "`-- set define off`" line to look like this "`set define off`" (Line 7)
- ♦ Uncomment the "`--/`" line that follows the `CREATE TRIGGER` statements (there are 12 of these toward the very end of the file on line 71,282. Just before the `CREATE INDEX` statements). Change to be just `/`. (This terminates the trigger procedure definition in `sqlplus`.)
- ♦ Save the file.



If you build and load the sample databases using `buildomatic`, the `NLS_LANG` setting is automatically handled via a JDBC driver setting.

If you load the sample databases using `buildomatic`, you will not need to set any variables or make any script edits.

3. (Optional) Set the `NLS_LANG` variable. The `sugarcrm` database has test data that requires a specific `NLS_LANG` setting in order to load into Oracle correctly. You will need to set this in your shell environment if you are manually loading the `sugarcrm` database.

Windows:     `set NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P1`

Linux:        `export NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P1`

4. (Optional) Run the following commands if you want to install sample databases:

```
cd <js-install>/buildomatic/install_resources/sql/oracle

sqlplus /nolog (start sqlplus client)
SQL> connect system/password (use your sysUsername and password)
(or SQL>connect sys/password as sysdba
SQL> create user sugarcrm identified by password;
SQL> create user foodmart identified by password;
SQL> grant connect, resource to sugarcrm;
SQL> grant connect, resource to foodmart;
SQL> connect sugarcrm/password@ORCL
SQL> @sugarcrm.sql (First, make sure file is unzipped)
SQL> connect foodmart/password@ORCL
SQL> @foodmart-oracle.sql (First, make sure file is unzipped)
SQL> @supermart-update.sql
SQL> exit
```

5. Continue from [step 2 in section 6.3, "Manually Creating the JasperReports Server Database," on page 48](#).

### 6.3.4 DB2

#### To manually create the JasperReports Server database in DB2:

The DB2 client software, db2 or db2cmd, can be used to interact with DB2.



For specific details on connecting to the DB2 database and setting privileges for databases and db users, please refer to the documentation provided with your database.

1. Change to the following directory:

```
cd <js-install>/buildomatic/install_resources/sql/db2
```

2. Enter these commands in the DB2 command window to create and initialize the repository database, called jsprsrvr in DB2 to conform to the 8-character limitation:

```
db2 create database jsprsrvr using codeset utf-8 territory us pagesize 16384
db2 -tf js-pro-create.ddl
db2 -tf quartz.ddl
```

3. (Optional) Run the following commands in the DB2 command window if you want to install sample databases:

```
db2 create database sugarcrm
db2 -tf sugarcrm.sql (first make sure file is unzipped)
db2 create database foodmart
db2 -tf foodmart-db2.sql (first make sure file is unzipped)
db2 -tf supermart-update.sql (if script is available)
```

4. Continue from [step 2 in section 6.3, "Manually Creating the JasperReports Server Database," on page 48](#).

Further considerations:

- If you are using DB2 8.1, set the LOGFIL\_SIZ parameter to at least 3000 to avoid possible log file errors while loading the foodmart database. Configure your foodmart database right after creating it by using Control Center.
- If JasperReports Server is deployed on the same host as DB2, delete the following file to avoid conflicts:

```
<db2>/SQLLIB/java/db2jcc.jar
```

### 6.3.5 SQL Server

Using the `sqlcmd` utility to manually build the `jasperserver` database.



For specific details on connecting to the SQL Server database and setting privileges for databases and db users, please refer to the documentation provided with your database.

#### To manually create the JasperReports Server database in SQL Server:

1. Open a Command Prompt and enter the following commands using the administrator (sa) user name and password.

```
cd <js-install>\buildomatic\install_resources\sql\sqlserver

sqlcmd -S ServerName -Usa -Psa
1> CREATE DATABASE [jasperserver]
2> GO

1> USE [jasperserver]
2> GO

1> :r js-pro-create.ddl
2> GO

1> :r quartz.ddl
2> GO
```

2. From the Windows Start Menu, select **Microsoft SQL Server > SQL Server Management Studio**.
3. Connect to SQL Server as the administrative database user, and check that the `jasperserver` database appears in the Object Explorer.
4. Expand Tables in the `jasperserver` database, and check that the tables have been added.

#### To manually create the optional sample databases in SQL Server:

1. Extract the files in the `sugarcrm.zip` file to the level above your current directory, placing the `sugarcrm.sql` file in this directory:

```
<js-install>\jasperserver\buildomatic\install_resources\sql\sqlserver
```

2. Enter these commands to create and initialize the `sugarcrm` database:

```
1> CREATE DATABASE [sugarcrm]
2> GO

1> USE [sugarcrm]
2> GO

1> :r sugarcrm.sql
2> GO
```

3. Extract the files in the `FoodMartCreateData.zip` into your current directory.

4. Enter these commands to create and initialize the foodmart database:

```
1> CREATE DATABASE [foodmart]
2> GO

1> USE [foodmart]
2> GO

1> :r FoodMartCreateData.sql
2> GO
```

5. Continue from [step 2 in section 6.3, "Manually Creating the JasperReports Server Database,"](#) on page 48.

## 6.4 Locating and Changing Buildomatic Configuration Files

The Ant-based buildomatic scripts contain support files for setting up and configuring a number of databases and application servers. This section describes the locations and content of some of these files and how to change the content.

### 6.4.1 Regenerating Buildomatic Settings

Whenever you change your `default_master.properties` file and re-run the `js-install` scripts (or any other buildomatic target), your generated configuration settings are automatically updated. The generated settings are in this location:

```
<js-install>/buildomatic/build_conf/default
```

The settings are automatically regenerated based on the new timestamp found on the properties file.

If you want to explicitly cause your generated configuration to be regenerated, you can run the following buildomatic targets:

```
cd <js-install>/buildomatic
js-ant clean-config
js-ant gen-config
```

The first target clears the configuration template files in `buildomatic/build_conf/default` directory. The second rebuilds the configuration settings.



These commands exist as a convenience. Whenever `default_master.properties` is edited, the resulting configuration templates are regenerated automatically based on the updated time-stamp associated with the edited file.

### 6.4.2 Locating Buildomatic-Generated Property Files

After you set your database and application server property values, you initiate buildomatic which automatically generates the database and application server configuration files needed to prepare for a JasperReports Server installation.

The generated property files are in this location:

```
<js-install>/buildomatic/build_conf/default
```

Some of the key configuration files are:

```
js.jdbc.properties
js.quartz.properties
js-glassfish-ds.xml
js-jboss-ds.xml
maven_settings.xml - (used for source code build)
```

More generated property files are:

```
<js-install>/buildomatic/build_conf/default/webapp
```

Included in the /webapp directory are configuration files, such as:

```
META-INF/context.xml
WEB-INF/hibernate.properties
WEB-INF/js.quartz.properties
```

These autogenerated files are removed if you run the buildomatic target: `clean-config`. You can then regenerate the files by running the target: `gen-config`. (Also, after running `clean-config`, any subsequent target will regenerate the configuration files.)

### 6.4.3 Changing the JDBC Driver Deployed by Buildomatic

When you run the buildomatic target `deploy-webapp-pro` the JDBC driver for your specified database is copied to your application server.

If there is a different or more up-to-date JDBC driver that you prefer to use, change the driver used by updating your `default_master.properties` file:

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

In `default_master.properties`, set the `maven.jdbc.artifactId` and the `maven.jdbc.version` to point to the name of the driver you would like to use:

```
maven.jdbc.artifactId=<first-part-of-filename>
maven.jdbc.version=<version-part-of-filename>
```

Make sure that the property settings and the JDBC driver name exactly match, as shown in examples in the following sections.

Put new JDBC drivers in the same location as the existing JDBC drivers for your DB type. The buildomatic scripts will look in the default jdbc folder location described in [6.4.4, “Buildomatic Location for JDBC Drivers,” on page 55](#) that is associated with your DB type:

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

If you plan to run the `js-import.bat/.sh`, `js-export.bat/.sh` shell scripts, manually update the JDBC driver version found in the `<js-install>/buildomatic/conf_source/iePro/lib` folder.

#### 6.4.3.1 DB2 8.x Driver Example

The `js-install` or buildomatic Ant scripts install the `db2jcc-9.7.jar` (for DB2 9.x) by default. To change the driver installed by the scripts to `db2jcc-8.2.jar` (for DB2 8.x), uncomment the following lines at the bottom of the `default_master.properties` file and change 9.7 to 8.2:

```
maven.jdbc.artifactId=db2jcc
maven.jdbc.version=8.2
```

#### 6.4.3.2 Oracle 5 Driver Example

The `js-install` or buildomatic Ant scripts install the `ojdbc14.10.2.0.jar` by default. To changed the driver installed by the scripts to `ojdbc5.11.2.0.jar`, uncomment the following lines at the bottom of the `default_master.properties` file:

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

To change the driver used, edit your `default_master.properties` file as follows:

```
maven.jdbc.artifactId=ojdbc14
maven.jdbc.version=10.2.0
```

Now, when you run `deploy-webapp-pro`, the version 14 driver is used.

### 6.4.3.3 SQL Server Driver Example

The `js-install` or `buildomatic` Ant scripts install the `sqljdbc-1.6.jar` (for Java 6) by default. To change the driver installed by the scripts to a difference JAR `sqljdbc-1.x.jar` (where `x` = some future version), uncomment the following lines at the bottom of the `default_master.properties` file and change `1.6` to `1.x`:

```
maven.jdbc.artifactId=sqljdbc
maven.jdbc.version=1.x
```

The `buildomatic` Ant scripts will automatically copy the appropriate JDBC driver to your application server when you run the `deploy-webapp-pro` target. The `js-install` scripts will copy the appropriate JDBC driver to this location, also. Here are some typical locations where you can expect the JDBC driver to be copied:

```
Tomcat 5:      <tomcat>/common/lib
Tomcat 6 and 7: <tomcat>/lib
JBoss:         <jboss>/server/default/lib
GlassFish:     <glassfish>/domains/domain1/lib/ext
```

### 6.4.3.4 PostgreSQL Driver Example

You can use the same logic above to deploy other JDBC drivers. For example, you want to use the JDBC3 PostgreSQL driver and automatically deploy the driver to your application server when you run the `deploy-webapp-pro` target. However, the default driver setting, shown in the following file, is JDBC4:

```
<js-install>/buildomatic/conf_source/db/postgresql/db.properties
maven.jdbc.artifactId=postgresql
maven.jdbc.version=9.2-1002.jdbc4
So the driver used is: postgresql-9.2-1002.jdbc4.jar
```

To change the default driver setting, edit your `default_master.properties` file, add the following lines:

```
maven.jdbc.artifactId=postgresql
maven.jdbc.version=9.2-1002.jdbc3
```

Now, when you run `deploy-webapp-pro`, the driver used is: `postgresql-9.2-1002.jdbc3.jar`.

### 6.4.3.5 MySQL Driver Example

You can use the same logic above to deploy other JDBC drivers. For example, you want to use the `mariadb.jdbc` driver and automatically deploy the driver to your application server when you run the `deploy-webapp-pro` target.

To change the default driver settings, edit your `default_master.properties` file to use the following lines:

```
jdbcDriverClass=org.mariadb.jdbc.Driver
maven.jdbc.groupId=org.mariadb.jdbc
maven.jdbc.artifactId=mariadb-java-client
maven.jdbc.version=1.1.2
```

## 6.4.4 Buildomatic Location for JDBC Drivers

The installer provides JDBC drivers for all databases listed in the following table:

| Database   | Buildomatic JDBC Driver Location and Sample JARs   |
|------------|--|
| PostgreSQL | <code>&lt;js-install&gt;/buildomatic/conf_source/db/postgresql/jdbc/postgresql-9.2-1002.jdbc3.jar</code><br><code>&lt;js-install&gt;/buildomatic/conf_source/db/postgresql/jdbc/postgresql-9.2-1002.jdbc4.jar</code> |
| MySQL      | <code>&lt;js-install&gt;/buildomatic/conf_source/db/mysql/jdbc/mariadb-java-client-1.1.2.jar</code>  |
| Oracle     | <code>&lt;js-install&gt;/buildomatic/conf_source/db/oracle/jdbc/ojdbc6.11.2.0.2.jar</code><br><code>&lt;js-install&gt;/buildomatic/conf_source/db/oracle/jdbc/ojdbc14.10.2.0.jar</code>                              |

| Database                        | Buildomatic JDBC Driver Location and Sample JARs   |
|---------------------------------|--|
| DB2                             | <js-install>/buildomatic/conf_source/db/db2/jdbc/db2jcc-9.5.jar (for DB2 9.x)<br><js-install>/buildomatic/conf_source/db/db2/jdbc/alt-drivers/db2jcc-8.2.jar (for DB2 8.x) |
| SQL Server                      | <js-install>/buildomatic/conf_source/db/sqlserver/jdbc/sqljdbc-1.6.jar (for Oracle/Sun Java 6)   |
| Vertica (as a data source only) | <js-install>/buildomatic/conf_source/db/vertica/jdbc/vertica_4.1.19_jdk_5.jar  |

### 6.4.5 Buildomatic Location for JasperReports Server WAR File

Buildomatic takes the JasperReports Server WAR file from the root of the <js-install> directory:

```
<js-install>/jasperserver-pro.war
```

When you run the `deploy-webapp-pro` target, buildomatic takes the war archive and unpacks it into your application server. Next, the database configuration files needed by the application server are copied to the appropriate locations. For instance, in the case of Tomcat:

- <js-install>/jasperserver-pro.war  
Unpacked and copied to <tomcat>/webapps/jasperserver-pro/\*
- <js-install>/buildomatic/build\_conf/default/webapp/META-INF/context.xml  
Copied to <tomcat>/webapps/jasperserver-pro/META-INF/context.xml
- <js-install>/buildomatic/build\_conf/default/webapp/WEB-INF/hibernate.properties  
Copied to <tomcat>/webapps/jasperserver-pro/WEB-INF/hibernate.properties
- <js-install>/buildomatic/build\_conf/default/webapp/WEB-INF/js.quartz.properties  
Copied to <tomcat>/webapps/jasperserver-pro/WEB-INF/js.quartz.properties
- <js-install>/buildomatic/build\_conf/db/postgres/jdbc/postgresql-9.2-1002.jdbc4.jar  
Copied to <tomcat>/lib

### 6.4.6 Buildomatic Location for SQL Scripts

Buildomatic comes with SQL scripts and other utilities that support a number of databases. These files are in:

```
<js-install>/buildomatic/install_resources/sql/
```

For example, some key files are (same pattern for additional databases):

```
<js-install>/buildomatic/install_resources/sql/postgresql/js-pro-create.ddl
<js-install>/buildomatic/install_resources/sql/postgresql/quartz.ddl
<js-install>/buildomatic/install_resources/sql/postgresql/upgrade-postgresql-5.0.0-5.1.0-pro.sql
<js-install>/buildomatic/install_resources/sql/postgresql/js-pro-drop.ddl
<js-install>/buildomatic/install_resources/sql/postgresql/drop-quartz.ddl
```



You can run these scripts manually by copying them to a location where your database client software is located.

### 6.4.7 Buildomatic Location for Database Creation Scripts

For most databases the buildomatic scripts are able to create the metadata repository database used by JasperReports Server. This is the database where the data defining users, roles, data sources, reports, OLAP views, domains, and other data are stored. This database is normally named `jasperserver`.

Buildomatic attempts to create the `jasperserver` database via JDBC when the `create-js-db` target is executed.

The scripts and property files used to create the `jasperserver` database are here:

```
<js-install>/buildomatic/conf_source/db/
```



```
postgresql/scripts.properties
mysql/scripts.properties
oracle/scripts.properties
(same pattern for additional databases)
```

### 6.4.8 Buildomatic Location for Sample Data Catalog ZIP Files

Buildomatic includes export files which hold the JasperReports Server sample data (that have examples of new features). This sample data is loaded when you run the buildomatic target `import-sample-data-pro`, for instance. These export files along with other important export files are located here:

```
<js-install>/buildomatic/install_resources/export/
```

Here are some key files (same pattern for additional databases):

```
js-catalog-postgresql-minimal-pro.zip
js-catalog-postgresql-pro.zip
js-catalog-mysql-minimal-pro.zip
js-catalog-mysql-pro.zip
js-catalog-oracle-minimal-pro.zip
js-catalog-oracle-pro.zip
```

### 6.4.9 Hibernate Properties Settings

Your `hibernate.properties` settings are in the following directory after buildomatic has been run to automatically generate your configuration files:

```
<js-install>/buildomatic/build_conf/default/webapp/WEB-INF/hibernate.properties
```

Within the `jasperserver-pro` WAR file the `hibernate.properties` file is found at the following location:

```
<appserver-path>/jasperserver-pro/WEB-INF/hibernate.properties
```

The buildomatic scripts automatically create this configuration file. When you run the buildomatic target `deploy-webapp-pro` this file is copied to JasperReports Server in your application server.

Hibernate property values are:

```
PostgreSQL metadata.hibernate.dialect=com.jaspersoft.hibernate.dialect.PostgresqlNoBlobDialect
MySQL 5.1 metadata.hibernate.dialect=org.hibernate.dialect.MySQLInnoDBDialect
MySQL 5.5 metadata.hibernate.dialect=org.hibernate.dialect.MySQL5InnoDBDialect
DB2: metadata.hibernate.dialect=com.jaspersoft.ji.hibernate.dialect.DB2JICustomDialect
Oracle: metadata.hibernate.dialect=com.jaspersoft.ji.hibernate.dialect.OracleJICustomDialect
SQL Server metadata.hibernate.dialect=com.jaspersoft.ji.hibernate.dialect.SQLServerJICustomDialect
```

### 6.4.10 Database Connection Configuration Files

#### 6.4.10.1 Tomcat

After setting up the buildomatic configuration for your database, the Tomcat `context.xml` will be automatically created with the appropriate settings for JasperReports Server.

When the buildomatic target `deploy-webapp-pro` is run, the `context.xml` will be automatically copied into the `jasperserver-pro` WAR set of files.

You can view the automatically generated `context.xml` at the following location:

```
<js-install>/buildomatic/build_conf/default/webapp/META-INF/context.xml
```

The final location of the `context.xml` is:

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

Tomcat will often create a copy of the context.xml file with a changed name that will be read instead of the one found in the jasperserver-pro war file. This is often a source of confusion for Tomcat users who attempt change their database settings. If you change your settings, delete the file in this location:

<tomcat>/conf/Catalina/localhost

### 6.4.10.2 JBoss

After setting up the buildomatic configuration for your database, the JBoss data source definition file will be automatically created with the appropriate settings for JasperReports Server.

When the buildomatic target `deploy-webapp-pro` is run, the `js-jboss-ds.xml` will be automatically copied into the JBoss instance.

You can view the automatically generated `js-jboss-ds.xml` at the following location:

<js-install>/buildomatic/build\_conf/default/js-jboss-ds.xml (JBoss 5.1)

<js-install>/buildomatic/build\_conf/default/js-jboss7-ds.xml (JBoss 7.1)

The final location of the `js-jboss-ds.xml` is:

<jboss>/server/default/deploy/js-jboss-ds.xml (JBoss 5.1)

<jboss>/standalone/deployments/jasperserver-pro.war/WEB-INF/js-jboss7-ds.xml

When JasperReports Server is running under JBoss, there are a couple of INFO log messages and an XML/A connection error that might occur depending on the version of JBoss you are running with.

For more information, refer to troubleshooting section [A.9.7, “JBoss Modifications,” on page 156](#).

### 6.4.10.3 Glassfish

After setting up the buildomatic configuration for your database, the Glassfish data source definition file `js-glassfish-ds.xml` will be automatically created with the appropriate settings. When the buildomatic target `deploy-webapp-pro` is run, the file is automatically deployed to the Glassfish instance.

You can view the automatically generated `js-glassfish-ds.xml` at the following location:

<js-install>/buildomatic/build\_conf/default/js-glassfish-ds.xml

To deploy the datasource definition manually, you can run a command similar to the following:

```
asadmin add-resources "<js-install>/buildomatic/build_conf/default/js-glassfish-ds.xml"
```

## 6.5 Configuring Report Scheduling

The JasperReports Server report scheduling feature is powered by the Quartz scheduler tool. The configuration settings for Quartz-based report scheduling is automatically handled by buildomatic.

In a deployed JasperReports Server instance, you will find the `js.quartz.properties` file in the following location:

<app-server-path>/jasperserver-pro/WEB-INF/js.quartz.properties

For mail server configuration, there is an additional property setting for authentication in the following file:

<app-server-path>/webapps/jasperserver-pro/WEB-INF/applicationContext-report-scheduling.xml

There are four main configurations to be discussed in this section:

- ♦ Mail Server Configuration
- ♦ Quartz Driver Delegate Class
- ♦ Report Scheduler Web URI
- ♦ Quartz Table Prefix
- ♦ Settings for import-export
- ♦ Setting Properties in the default\_master.properties File

### 6.5.1 Mail Server Configuration Settings

If you schedule reports or run them in the background, you can specify email addresses to notify when the report completes. To use this feature, configure JasperReports Server to contact an email server:

| Configuration File   |  |   |
|--|--|---|
| <app-server>/<deployment>/WEB-INF/js.quartz.properties                     |  |   |
| Property   | Description  |   |
| report.scheduler.mail.sender.host  | The name of the computer hosting the mail server   |   |
| report.scheduler.mail.sender.username                                      | The name of the user in the mail server that JasperReports Server can use  |   |
| report.scheduler.mail.sender.password                                      | The password of the mail server user   |   |
| report.scheduler.mail.sender.from  | The address that appears in the <b>From</b> field on email notifications   |   |
| report.scheduler.mail.sender.protocol                                      | The protocol that the mail server uses. JasperReports Server only supports SMTP.<br>Note: Your entry must be lower case. For example: smtp             |   |
| report.scheduler.mail.sender.port  | The port number that the mail server uses. For SMTP, the default is typically 25 (values other than 25 may not work in earlier JasperServer versions). |   |
| Configuration File   |  |   |
| <app-server>/<deployment>/WEB-INF/applicationContext-report-scheduling.xml |  |   |
| Property   | Bean   | Description   |
| javaMailProperties<br>key="mail.smtp.auth"                                 | reportScheduler<br>MailSender  | If your mail server requires authentication, change this property from false to true. |

### 6.5.2 Database Settings for the Quartz Driver Delegate Class

Quartz uses the Quartz driver delegate class to interact with the JDBC driver.



If you used buildomatic to install JasperReports Server, the correct value of the Quartz driver delegate class is set automatically for your database.

If you didn't use buildomatic to install JasperReports Server, use this table to edit the `js.quartz.properties` file and set the value of the Quartz driver delegate class to the correct value for your database:

| Configuration File                                     |            |   |
|--|------------|---|
| <app-server>/<deployment>/WEB-INF/js.quartz.properties |            |   |
| Property   | Database   | Value   |
| quartz.delegateClass                                   | DB2        | org.quartz.impl.jdbcjobstore.DB2v8Delegate      |
|  | Oracle     | org.quartz.impl.jdbcjobstore.StdJDBCDelegate    |
|  | PostgreSQL | org.quartz.impl.jdbcjobstore.PostgreSQLDelegate |
|  | SQL Server | org.quartz.impl.jdbcjobstore.StdJDBCDelegate    |
|  | MySQL      | org.quartz.impl.jdbcjobstore.StdJDBCDelegate    |

### 6.5.3 Settings for the Report Scheduler Web URI

JasperReports Server uses the Report Scheduler Web URI to construct the link it sends in the output from a scheduled job. This link must be correct for the user to access the report on the server.

The port on which you run JasperReports Server and context root of the deployed JasperReports Server web application determine the report scheduler Web URI. The default context root is `jasperserver`.

To set this value manually, edit the following file: `<app-server>/<deployment>/WEB-INF/js.quartz.properties`. Change the properties for your application server as shown in the following table.

| Property   | App Server    | Example Value                                       |
|--|---------------|---|
| <code>report.scheduler.web.deployment.uri</code> | Apache Tomcat | <code>http://localhost:8080/jasperserver-pro</code> |
|  | JBoss         | <code>http://localhost:8080/jasperserver-pro</code> |
|  | GlassFish     | <code>http://localhost:8080/jasperserver-pro</code> |
|  | WebLogic      | <code>http://localhost:7001/jasperserver-pro</code> |
|  | WebSphere     | <code>http://localhost:9080/jasperserver-pro</code> |

### 6.5.4 Settings for the Quartz Table Prefix

For databases that support schemas, such as Oracle, SQL Server, and DB2, you can set the Quartz table prefix so that it includes the schema if you use one. In the default configuration, only DB2 requires an explicit schema name.



If you installed JasperReports Server using buildomatic the Quartz table prefix is set automatically.

To set this value, edit the file `<app-server>/<deployment>/WEB-INF/js.quartz.properties`. Change the following property:

| Property                        | Description   |
|---------------------------------|---|
| <code>quartz.tablePrefix</code> | The prefix for the quartz table, including any schema name, for example <code>JSPRSVR.QRTZ_</code> for DB2. |

### 6.5.5 Settings for Import-Export

If you are manually configuring the import-export shell scripts instead of using the buildomatic, make sure your settings for the Quartz driver delegate class property are correct for the database you are using.



If you installed using buildomatic these settings are handled automatically (in buildomatic import-export).

To configure the import-export scripts manually, edit the following file:

```
<js-install>/buildomatic/conf_source/iePro/js.quartz.properties
```

Change the following properties:

| Property                          | Description  |
|-----------------------------------|--|
| <code>quartz.delegateClass</code> | Set to the same value as described in section 6.5.2, “ <a href="#">Database Settings for the Quartz Driver Delegate Class</a> ,” on page 59. |
| <code>quartz.tablePrefix</code>   | Set to the same value as described in section 6.5.4, “ <a href="#">Settings for the Quartz Table Prefix</a> ,” on page 60                    |

## 6.5.6 Setting Properties in the default\_master.properties File

At installation time, there is a group of properties that can be set in order to configure JasperReports Server functionality. These properties are found in the `default_master.properties` file. They can be un-commented to set them to the non-default values that will take effect upon installation. Look in the `default_master.properties` file for additional details.

Your `default_master.properties` file should be found here:

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

A sample master.properties can be found here (in the case of PostgreSQL):

```
<js-install>/buildomatic/sample_conf/postgresql_master.properties
```

When the `js-install.sh/bat` script is executed (or the underlying `deploy-webapp-pro` ant target), these properties will set into the deployed JasperReports Server in the `js.quartz.properties` file.

### 6.5.6.1 Report Scheduler Email Properties

As of Release 5.5, properties have been added that will set configuration for the Report Scheduler Email properties.

The properties that can be set are the following (default values are shown):

```
quartz.mail.sender.host=mail.localhost.com
quartz.mail.sender.port=25
quartz.mail.sender.protocol=smtp
quartz.mail.sender.username=admin
quartz.mail.sender.password=password
quartz.mail.sender.from=admin@localhost.com
quartz.web.deployment.uri=http://localhost:8080/jasperserver-pro
```

### 6.5.6.2 Diagnostic Properties

Properties which will set the configuration of the Diagnostic functionality:

```
diagnostic.jmx.usePlatformServer = false
diagnostic.jmx.port = 10990
diagnostic.jmx.name = jasperserver
diagnostic.jmx.rmiHost = localhost
```

Look at the descriptions of the properties in the `default_master.properties` file and also refer to the *JasperReports Server Administrator Guide* for more information on these settings.

## 6.6 Updating XML/A Connection Definitions

Sample XML/A connections are included with the JasperReports Server sample data. If you plan to use XML/A Web Services in your environment, then you may want to check and possibly update the hard coded values in the sample connections.

If you have Jaspersoft OLAP enabled (via your license), JasperReports Server is able to make XML/A connections over the Web Services interface. These HTTP-based connections use a user account for authentication. You may have different usernames and passwords than the defaults that get loaded from the sample data load in the sections above. Additionally, your application server hostnames and port values might be different than the default values. In such cases, the connections and any other resources that rely on them will fail.

There are two sample connections:

- Foodmart Sample XML/A connection
- SugarCRM Sample XML/A connection

If you would like to validate and update these resources, do the following:

1. Log into JasperReports Server as an administrator (such as `jasperadmin`).
2. Navigate to the Repository Management page by selecting the **View > Repository** menu item.

3. Click to expand the Analysis Components folder, then the Analysis Connections folder. Click to highlight the **Foodmart XML/A Connection** resource, then click **Edit**.
4. Edit the following information on this screen:
  - ♦ URI (hostname and port)
  - ♦ Login Username
  - ♦ Login Password
5. Click **Next**, then **Save**.
6. Make the same updates for the **SugarCRM XML/A Connection** resource.

## CHAPTER 7    INSTALLING THE WAR FILE FOR WEBSPHERE

---

JasperReports Server supports deployment on the IBM WebSphere Application Server, but requires its own database to store information such as users, organizations, and the repository. WebSphere users need the WAR file distribution to install JasperReports Server. Download the WAR file distribution from [Jaspersoft technical support](#) or contact your sales representative. The WAR file distribution comes in a file named `jasperreports-server-5.5-bin.zip`, the compressed ZIP format.

The WAR file distribution also includes two sample databases containing data for optional demos. For evaluation, Jaspersoft recommends you install the sample databases. In a production environment, you typically don't install the sample databases. You create and initialize the required repository database and the optional sample databases before JasperReports Server is deployed in WebSphere. Deployment is performed by the WebSphere administrator using the WebSphere Administrative Console.

This chapter contains the following sections:

- **Procedure for Installing and Deploying the WAR File in WebSphere**
- **Logging into JasperReports Server**
- **Configuring Report Scheduling**
- **Updating XML/A Connection Definitions (Optional)**
- **Troubleshooting your JasperReports Server Configuration**

## 7.1 Procedure for Installing and Deploying the WAR File in WebSphere

Perform the procedures in this section to install and deploy the JasperReports Server WAR file in WebSphere.

### 7.1.1 Installing WebSphere and a Database

#### Install WebSphere and a database:

1. (If you are not using WebSphere 7.0, you can skip this step.) Apply WebSphere Fix 7.0.0.11 to WebSphere 7.0 or install WebSphere 7.0.0.11.
2. Check that the WebSphere installation created a JAVA\_HOME system environment variable. The variable needs to be set to the JAVA directory in the WebSphere installation.
3. Install the PostgreSQL, MySQL, Oracle, SQL Server, or DB2 database.



The target database can be on a remote server. WebSphere should reside on the local machine.

4. If you use the MySQL database, go to the following web site and download the MySQL JDBC driver for your database:  
<http://dev.mysql.com/downloads/connector/j/>

Save the driver in:

```
<js-install>\buildomatic\conf_source\db\mysql\jdbc\
```

Next, prepare the JasperReports Server files.

### 7.1.2 Preparing JasperReports Server Files

#### To prepare JasperReports Server files:

1. Unpack jasperreports-server-5.5-bin.zip to a top-level directory.  
Unpacking the ZIP file creates the directory jasperreports-server-5.5-bin.
2. Manually create and load the JasperReports Server database. Required.
3. Manually create and load the sample databases. Optional but recommended.  
Section [6.1, “Setting JVM Options for Application Servers,” on page 41](#) contains instructions for manually creating the databases.
4. Manually import the default users and organization. Required.
  - a. Copy the <database>\_master.properties file for your database from /sample\_conf and paste it to /buildomatic:
    - Copy from — <js-install>/buildomatic/sample\_conf/
    - Paste to — <js-install>/buildomaticFor example, copy postgresql\_master.properties to <js-install>/buildomatic.
  - b. Rename the file you copied to default\_master.properties.
  - c. Edit the default\_master.properties file:
    - Set appServerType to skipAppServerCheck.
    - Change dbUsername, dbPassword, and dbHost to appropriate settings for your database.
    - If you are using a different port from the default for your database, or if you have installed the database on a remote machine, change the dbPort field under Custom Properties to the appropriate settings.Each sample\_conf.properties file contains appropriate sample values.
  - d. Start your database server.



- e. Open a Command Prompt as Administrator and run these commands:

**Table 7-1 Buildomatic Targets to Execute**

| Commands                                       | Description  |
|--|--|
| <code>cd &lt;js-install&gt;/buildomatic</code> | Go to the buildomatic directory  |
| <code>js-ant init-js-db-pro</code>             | Initializes database. If the database has already been initialized, you may see messages that data already exists. You can safely ignore these messages. |
| <code>js-ant import-minimal-pro</code>         | Loads core application data. Running <code>js-ant import-minimal-pro</code> is mandatory. The server cannot function without this data.                  |
| <code>js-ant import-sample-data-pro</code>     | (Optional) Loads the demos that use the sample data.   |

5. Set up your license file. For information, refer to [6.2, “Setting Up the JasperReports Server License,” on page 45](#).

Next, configure Hibernate and Quartz information in the WAR file.

### 7.1.3 Configuring the Hibernate and Quartz Settings

Before deploying the JasperReports Server WAR file, update the Hibernate and Quartz settings that it contains, as described in this procedure.

#### To configure the Hibernate and Quartz settings in the WAR file:

1. The WAR file is an archive format in a single file. To access its configuration files, extract them using the following commands.

If you are using PostgreSQL, you do not need to extract the `hibernate.properties` file.

```
cd <js-install>
"%JAVA_HOME%\bin\jar" xf jasperserver-pro.war WEB-INF/hibernate.properties
"%JAVA_HOME%\bin\jar" xf jasperserver-pro.war WEB-INF/js.quartz.properties
```

The `jar` command creates the `WEB-INF` directory in the current location and places the extracted files there.

2. Open the file `WEB-INF/hibernate.properties` for editing and locate the `metadata.hibernate.dialect` property. Set its value according to your database.

If you are using PostgreSQL, you can skip this step.

MySQL 5.1 `metadata.hibernate.dialect=org.hibernate.dialect.MySQLInnoDBDialect`

MySQL 5.5 `metadata.hibernate.dialect=org.hibernate.dialect.MySQL5InnoDBDialect`

DB2: `metadata.hibernate.dialect=com.jaspersoft.ji.hibernate.dialect.DB2JICustomDialect`

Oracle: `metadata.hibernate.dialect=com.jaspersoft.ji.hibernate.dialect.OracleJICustomDialect`

SQL Server `metadata.hibernate.dialect=com.jaspersoft.ji.hibernate.dialect.SQLServerJICustomDialect`

3. Open the file `WEB-INF/js.quartz.properties` for editing and locate the `report.scheduler.web.deployment.uri` property. Set its value to the hostname, port, and path where you intend to deploy JasperReports Server, for example:

```
report.scheduler.web.deployment.uri=http://localhost:9080/jasperserver-pro
```

If you are using PostgreSQL or DB2, set the Quartz delegate class in the `js.quartz.properties` file:

PostgreSQL: `quartz.delegateClass=org.quartz.impl.jdbcjobstore.PostgreSQLDelegate`

DB2: `quartz.delegateClass=org.quartz.impl.jdbcjobstore.DB2v8Delegate`

All others: `quartz.delegateClass=org.quartz.impl.jdbcjobstore.StdJDBCDelegate`

- If you are running Websphere 7, open the file `WEB-INF/esapi/Owasp.CsrfGuard.properties` and make the following change:

From:

```
org.owasp.csrfguard.PRNG=SHA1PRNG
```

To:

```
org.owasp.csrfguard.PRNG=IBMSecureRandom
```

```
org.owasp.csrfguard.PRNG.Provider=IBMJCE
```

- If you want to configure JasperReports Server to automatically schedule and email reports, enter your mail server information in the `js.quartz.properties` file. Modify all `report.scheduler.mail.sender.*` properties as necessary for your mail server.
- Now that you have modified the individual configuration files, you must replace them into the WAR file archive using the following commands.

If you are using PostgreSQL, you do not need to replace the `hibernate.properties` file.

```
cd <js-install>
"%JAVA_HOME%\bin\jar" uf jasperserver-pro.war WEB-INF/hibernate.properties
"%JAVA_HOME%\bin\jar" uf jasperserver-pro.war WEB-INF/js.quartz.properties
```

- Delete the `WEB-INF` directory that you created, along with the edited files it contains.

Next, configure a JDBC Provider in WebSphere.

### 7.1.4 Configuring a JDBC Provider in WebSphere

**To configure a JDBC Provider in WebSphere:**

- Launch the WebSphere Administrative Console and navigate to **Resources > JDBC > JDBC Providers**.
- On the JDBC providers page, click the **Guided Activity** link at the top of the **JDBC Providers** page and follow Integrated Solutions Console instructions:

- Configure credentials for a secure database.

Use the J2C authentication aliases panel to create a new authenticated user.

In Global Security, click **New** and enter the following user alias, user ID, and password. These are the credentials that WebSphere uses to access the database:

**Table 7-2 J2C Authentication Alias Settings**

|                  | PostgreSQL          | MySQL          | Oracle            | DB2           | SQL Server        |
|------------------|---------------------|----------------|-------------------|---------------|-------------------|
| Alias            | postgresql_jasperdb | mysql_jasperdb | jasperserver_user | db2admin_user | jasperserver_user |
| Example User ID  | postgres            | root           | jasperserver      | db2admin      | sa                |
| Example Password | postgres            | password       | password          | password      | sa                |

- Connect to a database panel. From the Scope drop-down, choose **Node:<node\_name>,Server=<server\_name>**
- Click the **New** button to create a new JDBC Provider.
- If your database is PostgreSQL or MySQL, select User-defined for the database type; otherwise, select DB2, Oracle, or SQL Server.

Different options appear depending on your database type.

d. Select or enter these options, depending on your database type:

| Database type             | Implementation class name or type                               | Name                                  |
|---------------------------|---|---------------------------------------|
| User-defined (PostgreSQL) | org.postgresql.jdbc2.optional.ConnectionPool                    | PostgreSQL JDBC Provider              |
| User-defined (MySQL)      | com.mysql.jdbc.jdbc2.optional.Mysql<br>ConnectionPoolDataSource | MySQL JDBC Provider                   |
| DB2                       | Connection pool data source                                     | DB2 Universal JDBC Driver<br>Provider |
| Oracle                    | Connection pool data source                                     | Oracle JDBC Driver                    |
| SQL Server                | Connection pool data source                                     | Microsoft SQL Server<br>JDBC Driver   |

3. Click **Next** and enter the database classpath information for the JDBC provider:

<js-install>\buildomatic\conf\_source\db\<your database>\jdbc\

For example, enter:

C:\jasperreports-server-5.5-bin\buildomatic\conf\_source\db\postgresql\jdbc\postgresql-9.2-1002.jdbc4.jar

Alternatively, you can copy the JAR file to a location in your WebSphere deployment and specify that location for the JDBC driver path.



If JasperReports Server is deployed on the same host as DB2, delete the following file to avoid conflicts:  
<db2>/SQLLIB/java/db2jcc.jar

4. Click **Next** to proceed to the next step.
5. Review the JDBC provider information you have entered and click **Finish**.
6. Configure data sources and custom properties using one of these procedures:
  - Section 7.1.4.1, “[Defining a JNDI Name and Sample Data Sources for PostgreSQL and MySQL,](#)” on [page 67](#)
  - Section 7.1.4.2, “[Defining a JNDI Name and Sample Data Sources for DB2,](#)” on [page 68](#)
  - Section 7.1.4.3, “[Defining a JNDI Name and Sample Data Sources for Oracle,](#)” on [page 70](#)
  - Section 7.1.4.4, “[Defining a JNDI Name and Sample Data Sources for SQL Server,](#)” on [page 71](#)

#### 7.1.4.1 Defining a JNDI Name and Sample Data Sources for PostgreSQL and MySQL

**To define the JDBC data source and expose it through JNDI:**

1. Click the name of the JDBC provider that you just created. For example, **PostgreSQL JDBC Provider**.
2. Click **Data sources** under the Additional Properties of the JDBC provider details panel.
3. To create a new data source, click **New** and the new data source wizard appears.
4. Enter the data source name: `jasperserver`
5. Enter the JNDI name: `jdbc/jasperserver`
6. Click **Next**, choose **Select an existing JDBC provider**, then select PostgreSQL JDBC Provider or MySQL JDBC Provider from the drop-down list.
7. Click **Next** and accept the default helper class (`com.ibm.websphere.rsadapter.GenericDataStoreHelper`). Select the check box to use this data source in container managed persistence (CMP).

8. Click **Next** and select the Setup security aliases:

| Field Name                             | PostgreSQL Value           | MySQL Value             |
|--|----------------------------|-------------------------|
| Component-managed authentication alias | <node>/postgresql_jasperdb | <node>/mysql_jasperdb   |
| Mapping configuration alias            | DefaultPrincipalMapping    | DefaultPrincipalMapping |
| Container-managed authentication alias | <node>/postgresql_jasperdb | <node>/mysql_jasperdb   |

9. Click **Next**, review the summary information, and click **Finish**.

Next, define custom properties.

#### To define custom properties:

- In the list of JDBC data sources, select the check box for the newly created **jasperserver** data source, and click **Test Connection**.  
In the Messages area a success or failure message appears. The failure message gives you information about which custom properties you need to define. For example, in the case of PostgreSQL 9.0 and MySQL, a message indicates that the error is related to a null database name.
- Navigate to the jasperserver data sources General Properties page.
- In Additional Properties on the right side of the General Properties page of the jasperserver data source, click **Custom properties**.
- Scroll down the list of properties and select **databaseName**. Set the value to **jasperserver**.
- If you're using MySQL, create a new property called **url**. Enter the following value and save the change:  
`jdbc:mysql://localhost/jasperserver?useUnicode=true&characterEncoding=UTF-8`
- Click **Save directly to the master configuration**.

Next, if you installed sample databases, sugarcrm and foodmart, create data sources to use them.

#### To create optional sugarcrm and foodmart datasources:

- If you plan to run the sample reports, use the following values to create the foodmart and sugarcrm JNDI data sources:

| Field Name       | Value         |               |
|------------------|---------------|---------------|
| Data source name | foodmart      | sugarcrm      |
| JNDI name        | jdbc/foodmart | jdbc/sugarcrm |

- Click **Save directly to the master configuration**.

Next, deploy the WAR file in WebSphere as described in section [7.1.5, “Deploying the JasperReports Server WAR File in WebSphere,”](#) on page 71.

#### 7.1.4.2 Defining a JNDI Name and Sample Data Sources for DB2

##### To define the JDBC data source and expose it through JNDI:

- Click the name of the JDBC provider that you just created. For example, **DB2 Universal JDBC Provider**.
- Click Data sources under the Additional Properties of the JDBC provider details panel.
- To create a new data source, click New and the new data source wizard appears.
- Enter the data source name: `jasperserver`
- Enter the JNDI name: `jdbc/jasperserver`
- Click **Next**, choose **Select an existing JDBC provider**, then select **DB2 Universal JDBC Provider** from the drop-down list.

7. Click **Next** and enter these values:

| Field Name    | Value     |
|---------------|-----------|
| Driver type   | 4         |
| Database name | jsprsrvr  |
| Server name   | localhost |
| Port number   | 50000     |

8. Select **Use this data source in CMP** and click **Next**.
9. In the Setup security aliases page, enter the following value for Component-managed authentication alias:  
 <node>/db2admin\_user
10. Click **Next**, review the summary information, and click **Finish**.

Next, define custom properties.

**To define custom properties:**

- In the list of JDBC data sources, select the check box for the newly created **jasperserver** data source, and click **Test Connection**.
- In the Messages area a success or failure message appears. The failure message gives you information about which custom properties you need to define.
- Edit the following properties and add the ones that don't exist, then save the changes:

| Property Name                | Value   |
|------------------------------|---------|
| currentSchema                | JSPRSVR |
| fullyMaterializeLobData      | true    |
| fullyMaterializeInputStreams | true    |
| progressiveStreaming         | 2       |
| progressiveLocators          | 2       |

- Go back to the list of JDBC data sources, select the check box for the **jasperserver** data source, and click **Test Connection**.

Next, if you installed sample databases, sugarcrm and foodmart, create data sources to use them.

**To create optional sugarcrm and foodmart datasources:**

- If you plan to run the sample reports, use the following values to create the foodmart and sugarcrm JNDI data sources:

| Field Name                             | Value                |                      |
|--|----------------------|----------------------|
| Data source name                       | foodmart             | sugarcrm             |
| JNDI name                              | jdbc/foodmart        | jdbc/sugarcrm        |
| Component-managed authentication alias | <node>/db2admin_user | <node>/db2admin_user |

| Field Name    | Value     |           |
|---------------|-----------|-----------|
| Database name | foodmart  | sugarcrm  |
| Driver type   | 4         | 4         |
| Server name   | localhost | localhost |

|                             |          |          |
|-----------------------------|----------|----------|
| Port number                 | 50000    | 50000    |
| Use this data source in CMP | selected | selected |

| Property Name        | Value    |          |
|----------------------|----------|----------|
| currentSchema        | FOODMART | SUGARCRM |
| resultSetHoldability | 1        | 1        |

The other properties (fullyMaterializeLobData, fullyMaterializeInputStreams, progressiveStreaming, and progressiveLocators) are not needed for the sample databases.

- Click **Save directly to the master configuration**.

Next, deploy the WAR file in WebSphere as described in section [7.1.5, “Deploying the JasperReports Server WAR File in WebSphere,”](#) on page 71.

#### 7.1.4.3 Defining a JNDI Name and Sample Data Sources for Oracle

**Before defining the data source, change the name of the driver:**

- On the JDBC Providers page, click the newly created **Oracle JDBC Driver** to edit it.
- Locate the filename ojdbc14.jar and replace it with the filename ojdbc5-11.2.0.jar, then save the changes.

**To define the JDBC data source and expose it through JNDI:**

- Click the name of the JDBC provider that you just created. For example, **Oracle JDBC Driver**.
- Click Data sources under the Additional Properties of the JDBC provider details panel.
- To create a new data source, click New and the new data source wizard appears.
- Enter the data source name: `jasperserver`
- Enter the JNDI name: `jdbc/jasperserver`
- Click **Next**, choose **Select an existing JDBC provider**, then select **Oracle JDBC Driver** from the drop-down list.
- Click **Next** and enter the following values:

| Field Name                   | Value  |
|------------------------------|--|
| URL                          | <code>jdbc:oracle:thin:@localhost:1521:orcl</code> |
| Data store helper class name | Oracle11g data store helper                        |
| Use this data source in CMP  | selected   |

- Click **Next** and in Setup security alias, set Component-managed authentication alias to the following value:  
`<node>/jasperserver_user`
- Click **Next**, review the summary information and click **Finish**.

Next, define custom properties.

**To define custom properties:**

- In the list of JDBC data sources, select the check box for the newly created **jasperserver** data source, and click **Test Connection**.  
In the Messages area a success or failure message appears. The failure message gives you information about which custom properties you need to define.
- Navigate to the jasperserver data sources General Properties page.
- In Additional Properties on the right side of the General Properties page of the jasperserver data source, click **Custom properties**, and define properties.

Next, if you installed sample databases, sugarcrm and foodmart, create datasources to use them.

**To create optional sugarcrm and foodmart data sources:**

1. If you plan to run the sample reports, use the following values to create the foodmart and sugarcrm JNDI data sources:

| Field Name                             | Value                |                      |
|--|----------------------|----------------------|
| Data source name                       | foodmart             | sugarcrm             |
| JNDI name                              | jdbc/foodmart        | jdbc/sugarcrm        |
| Component-managed authentication alias | <node>/foodmart_user | <node>/sugarcrm_user |

2. Click **Save directly to the master configuration**.

Next, deploy the WAR file in WebSphere as described in section 7.1.5, “Deploying the JasperReports Server WAR File in WebSphere,” on page 71.

#### 7.1.4.4 Defining a JNDI Name and Sample Data Sources for SQL Server

**To define a JDBC provider:**

1. Click the name of the JDBC provider that you just created. For example, **Microsoft SQL Server JDBC Driver**.
2. Click Data sources under the Additional Properties of the JDBC provider details panel.
3. To create a new data source, click New and the new data source wizard appears.
4. Enter the data source name: `jasperserver`
5. Enter the JNDI name: `jdbc/jasperserver`
6. Click **Next**, choose **Select an existing JDBC provider**, then select **Microsoft SQL Server JDBC Driver** from the drop-down list.
7. Click **Next** and in Setup security alias, set Component-managed authentication alias to the following value:  
`<node>/sqlserver_jasperdb`
8. Click **Next**, review the summary information and click **Finish**.

Next, define custom properties.

**To define custom properties:**

1. In the list of JDBC data sources, select the check box for the newly created **jasperserver** data source, and click **Test Connection**.  
 In the Messages area a success or failure message appears. The failure message gives you information about which custom properties you need to define.
2. Navigate to the jasperserver data sources General Properties page.
3. In Additional Properties on the right side of the General Properties page of the jasperserver data source, click **Custom properties**, and define properties.

Next, if you installed sample databases, sugarcrm and foodmart, create datasources to use them.

**To create optional sugarcrm and foodmart data sources:**

1. If you plan to run the sample reports, use the following values to create the foodmart and sugarcrm JNDI data sources:
2. In the list of JDBC data sources, click the link for the newly created **jasperserver** data source.
3. Click **Save directly to the master configuration**.

Next, deploy the WAR file in WebSphere.

### 7.1.5 Deploying the JasperReports Server WAR File in WebSphere

**To deploy the JasperReports Server WAR file in WebSphere:**

1. In the Administrative Console, navigate to **Applications > New Application** and select **New Enterprise Application**.

JasperReports Server is a modern application, based on Java Servlet version 2.4, so you do not select the older, WebSphere V4-compliant application type.

2. Browse to `<js-install>/jasperserver-pro.war` on the local file system. Keep the default setting (**Fast path** selected) and click **Next**.
3. On the Select installation options page, accept all the default settings, and select **Save**. Click **Next**.
4. On the Map modules to servers page, make sure the JasperReports Server module is mapped to the cell, node, and server that you want. Click **Next**.
5. On the Map modules to servers page, select **jasperserver**. Click **Next**.
6. On the Map virtual hosts page, choose the **JasperServer UI application module**. Click **Next**.
7. In the Map context roots for Web modules, enter `jasperserver-pro`.
8. Click **Next**, review the summary information and start the installation process.
9. Click **Save directly to the master configuration**.

Next, set JVM Options. Required.

### 7.1.6 Setting JVM Options

#### To set the Java JVM Options:

For the JasperReports Server XML/A functionality to work, special Java JVM options need to be set to resolve class conflicts between the WebSphere and JasperReports Server web services implementation. JVM options also provide the optimal resources for running JasperReports Server.

#### To configure your Java JVM options:

1. Select **Enterprise Applications > jasperserver-pro\_war > Target specific application status > (server name)**.
2. Expand **Java and Process Management > Process Definition > Java Virtual Machine > Generic JVM arguments**.
3. In the text box named **Generic JVM Options**, paste in the following JVM options that explicitly specify JasperReports Server classes for AXIS and Xalan, as well as optimize JVM resources:



You can cut and paste these options from `<js-install>/samples/extra/java-settings/java-settings-websphere.txt`.

| Generic JVM Options on Windows |  |
|--------------------------------|--|
| Options for all databases      | -Dclient.encoding.override=UTF-8 -Xms1024m -Xmx2048m<br>-XX:PermSize=32m -XX:MaxPermSize=512m -Xss2m -XX:+UseConcMarkSweepGC<br>-XX:+CMSClassUnloadingEnabled<br>-Djavax.xml.soap.MessageFactory=org.apache.axis.soap.MessageFactoryImpl<br>-Djavax.xml.soap.SOAPConnectionFactory=org.apache.axis.soap.SOAPConnectionFactoryImpl<br>-Djavax.xml.soap.SOAPFactory=org.apache.axis.soap.SOAPFactoryImpl<br>-Djavax.xml.transform.TransformerFactory=org.apache.xalan.processor.TransformerFactoryImpl |
| For Oracle                     | set JAVA_OPTS=%JAVA_OPTS% -Doracle.jdbc.defaultNChar=true  |



| Generic JVM Options on Linux |  |
|------------------------------|--|
| Options for all databases    | -Dclient.encoding.override=UTF-8 -Xms1024m -Xmx2048m<br>-XX:PermSize=32m -XX:MaxPermSize=512m -Xss2m -XX:+UseConcMarkSweepGC<br>-XX:+CMSClassUnloadingEnabled<br>-Djavax.xml.soap.MessageFactory=org.apache.axis.soap.MessageFactoryImpl<br>-Djavax.xml.soap.SOAPConnectionFactory=org.apache.axis.soap.SOAPConnectionFactoryImpl<br>-Djavax.xml.soap.SOAPFactory=org.apache.axis.soap.SOAPFactoryImpl<br>-Djavax.xml.transform.TransformerFactory=org.apache.xalan.processor.TransformerFactoryImpl |
| For Oracle                   | export JAVA_OPTS="\$JAVA_OPTS -Doracle.jdbc.defaultNChar=true"   |



Setting the Oracle localization option, defaultNChar, can substantially impact the performance of JDBC queries. If you do not need to support UTF-8 for your Oracle database, you can omit this setting.

- Click **Save** on the console task bar.

#### To configure class loading:

- Select **Enterprise Applications > jasperserver-pro\_war > Class loading and update detection**.
- In the section **Class loader order** select **Classes loaded with local class loader first (parent last)**.
- In the section **WAR class loader policy** select **Single class loader for application**.
- Save directly to your master configuration.
- Restart WebSphere.

Next, start the jasperserver-pro application in WebSphere.

### 7.1.7 Starting and Restarting JasperReports Server

#### To start the jasperserver-pro application:

- Restart WebSphere.
- In the Administrative Console, navigate to: **Applications > Application Types > WebSphere Enterprise Application**.
- Select the check box next to the **jasperserver-pro** application, then click **Start** to start JasperReports Server.  
If you make configuration changes to your JasperReports Server instance, restart JasperReports Server.
- Log into JasperReports Server.

## 7.2 Logging into JasperReports Server

If JasperReports Server started up cleanly you should be able to login.

- Log in by going to the following URL:  
<http://<hostname>:9080/jasperserver-pro>  
Where <hostname> is localhost, a machine name, or an IP address. The login page should appear after some time to compile the necessary JSP files.
- Use the following administrative identity to log into the system:

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

If you logged in successfully, the JasperReports Server home page appears. If you have trouble logging in and get the following error message, you may be running at a WebSphere patch level that needs a further configuration:

Page cannot be found, HTTP 404 error

Refer to troubleshooting section [A.9.8, “Websphere Modifications,” on page 159](#).



The first time you log into JasperReports Server, you will be prompted to opt-in or opt-out of the JasperReports Server Heartbeat. For more information, refer to section [5.5.1, “JasperReports Server Heartbeat,” on page 38](#).

Refer to the *JasperReports Server User Guide* to begin adding reports and other resources to JasperReports Server.

## 7.3 Configuring Report Scheduling

The scheduled reporting feature of JasperReports Server allows reports to be run at pre-configured time intervals. Optionally, notification email can be sent to users to let them know that new reports are available.

### 7.3.1 Additional Fix for Scheduled Report with JNDI Data Source

Under the WebSphere application server, if you have a scheduled report that uses a JNDI data source you will need to make an additional configuration in order for the JNDI lookup to be resolved correctly.

There are special WebSphere specific configuration files included in the jasperserver-pro.war archive. These files are generated during the installation process. In order to enable this fix, you will need to rename the WebSphere specific files to remove the prefix webSphere. Two of the file rename operations will overwrite the existing configuration file names.

Rename: WEB-INF/webSphere-applicationContext-report-scheduling-wm.xml

To: WEB-INF/applicationContext-report-scheduling-wm.xml

Rename: WEB-INF/webSphere-js.quartz.base.properties

To: WEB-INF/js.quartz.base.properties (overwrite existing file)

Rename: WEB-INF/webSphere-js.scheduling.properties

To: WEB-INF/js.scheduling.properties (overwrite existing file)

#### Notes on this configuration change:

Since JasperServer 3.7 and 4.0, a new work manager class is used to run scheduled report jobs under WebSphere. The JNDI name of the work manager is configured in the WebSphere version of js.scheduling.properties. Also, for WebSphere, we need a different default value for the specified JNDI name (wm/default). This is also defined in the WebSphere version of js.scheduling.properties.

The number of threads that run report jobs is no longer configured in js.quartz.base.properties, but is instead provided by the work manager configuration.

### 7.3.2 Additional Change for Mail Server Authentication

If your mail server requires authentication, edit the applicationContext-report-scheduling.xml file after applying the changes above.

1. Extract the file from the WAR archive:

```
"%JAVA_HOME%\bin\jar" xf jasperserver-pro.war WEB-INF/applicationContext-report-scheduling.xml
```

2. Open the file for editing and locate the reportSchedulerMailSender bean.
3. Set the javaMailProperties key="mail.smtp.auth" value to true.
4. Save the file and replace it in the archive:

```
"%JAVA_HOME%\bin\jar" uf jasperserver-pro.war WEB-INF/applicationContext-report-scheduling.xml
```

5. Delete the WEB-INF directory that was created, along with the file it contains.

For more information about setting up report scheduling, refer to section [6.5, “Configuring Report Scheduling,” on page 58](#).

## 7.4 Updating XML/A Connection Definitions (Optional)

If you have loaded the JasperReports Server sample data, and you would like to run the Analysis XML/A examples, you will need to update the XML/A connection resources to use the correct web port.

The typical port used by WebSphere is 9080. Follow the procedure in section [6.6, “Updating XML/A Connection Definitions,” on page 61](#).

## 7.5 Troubleshooting your JasperReports Server Configuration

### 7.5.1 Startup Problems

The most common problems are errors in the database configuration. These problems are typically incorrect configurations within the database configuration files or in the application server configuration files. For information about resolving these types of errors, refer to troubleshooting section [Appendix A, “Troubleshooting,” on page 145](#).

### 7.5.2 Error Running Report

If you have trouble running reports in your new JasperReports Server instance, refer to troubleshooting section [A.8.11, “Error Running a Report,” on page 153](#). If you are having trouble running the MDX example Topic or SugarCRM OLAP view, you need to update the port for XML/A connections. See section [7.4, “Updating XML/A Connection Definitions \(Optional\),” on page 75](#).

### 7.5.3 Filter Error Using MySQL

The following error could be caused by an incorrect ampersand setting on your data source configuration:

```
Error 500: Filter [characterEncodingProxyFilter]: cold not be initialized
```

The data source line needs to have `&amp;` and not `&` in order to be evaluated correctly. That is, the URL you enter in the procedure to define the JDBC data source and expose it through JNDI should look like the following:

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

Note the ampersand string is `&amp;` and not `&`.

### 7.5.4 Error Creating Internationalized Name

If you encounter errors in JasperReports Server when creating resources with internationalized names, and you have an Oracle database, configure your Oracle JDBC driver. Set the Oracle-specific option listed in the tables of section [7.1.6, “Setting JVM Options,” on page 72](#).

### 7.5.5 Xerces Error

In earlier releases of JasperReports Server it was possible to find the following error in the WebSphere log:

```
SRVE0068E: Uncaught exception thrown in one of the service methods of the servlet:
jasperserver. Exception thrown: org.springframework.web.util.NestedServletException:
javax.xml.validation.SchemaFactoryFinder$ConfigurationError: Provider
org.apache.xerces.jaxp.validation.XMLSchemaFactory could not be instantiated:
org.apache.xerces.impl.dv.DVFactoryException: DTD factory class
org.apache.xerces.impl.dv.dtd.DTDDVFactoryImpl does not extend from DTDDVFactory.
```

Since around release 4.0, the xercesImpl jar used is version 2.7.1 and more recently 2.10.0.

The error shown above is caused by a conflict between the IBM JDK which is used by WebSphere and the xercesImpl-2.6.2 library which is bundled in older versions of JasperReports Server. There are two solutions:

- Remove the xercesImpl library from the following location:  
`<websphere>\profiles\AppSrv<NN>\installedApps\<node>\jasperserver-pro_war.ear\  
jasperserver-pro.war\WEB-INF\lib`
- Update the xercesImpl library to a new version if it is an old version.
- The newer xercesImpl jars do not cause this error.

### 7.5.6 OLAP View Fails With Exception

The following error could be caused by AspectJ requiring class loaders to be tried out in a specific order:

```
java.lang.NoSuchMethodError: org/aspectj/runtime/reflect/Factory.makeMethodSig(  
    java/lang/String;  
    ...)  
    org/aspectj/lang/reflect/MethodSignature;
```

Change the default class loader policy using the following procedure:

1. In the WebSphere Administrative Console, navigate to **Applications > (app-name) > Manage Modules > JasperServer UI application**.
2. Change the following setting:

| Property Name      | Value  |
|--------------------|--|
| Class loader order | Class loaded with local class loader first (parent last) |

3. Click **OK**.
4. Save the master configuration.
5. Restart the WebSphere server.

## CHAPTER 8    INSTALLING THE WAR FILE FOR WEBLOGIC

---

JasperReports Server supports deployment on the WebLogic Application Server, but requires its own database to store information such as users, organizations, and the repository. WebLogic users need the WAR file distribution to install JasperReports Server. Download the WAR file distribution from [Jaspersoft technical support](#) or contact your sales representative. The WAR file distribution comes in a file named `jasperreports-server-5.5-bin.zip`, the compressed ZIP format.

The WAR file distribution includes two sample databases containing data for optional demos. For evaluation, Jaspersoft recommends you install the sample databases. In a production environment, you typically don't install the sample databases. You create and initialize the required repository database and the optional sample databases before JasperReports Server is deployed in WebLogic. Deployment is performed by the WebLogic administrator using the WebLogic Administrative Console or `domain config.xml`.

This chapter contains the following sections:

- **Procedure for Installing the WAR File for WebLogic**
- **Setting Java Properties**
- **Configuring Other Database Connections**
- **Starting JasperReports Server**
- **Logging into JasperReports Server**
- **Configuring Report Scheduling**
- **Restarting JasperReports Server**
- **Updating XML/A Connection Definitions (Optional)**
- **Troubleshooting Your JasperReports Server Configuration**
- **Troubleshooting Problems with OLAP Views**

## 8.1 Procedure for Installing the WAR File for WebLogic

### To meet prerequisites for installing the WAR file for WebLogic:

1. Check that the Oracle/Sun Java JDK 1.6 or 1.7 is installed.
2. Check that the JAVA\_HOME system environment variable points to the JDK.
3. Install the PostgreSQL, MySQL, Oracle, SQL Server, or DB2 database.



The target database can be on a remote server.

### To install the WAR file for WebLogic:

1. Extract all files in jasperreports-server-5.5-bin.zip. Choose a top-level directory, such as C:\Jaspersoft on Windows or /home/<user> on Linux, for the extracted files.

Unpacking the ZIP file creates the directory jasperreports-server-5.5-bin.

2. Check that WebLogic is installed in the default location on your local machine.

If WebLogic is not installed in the default location, or if you encounter problems using the buildomatic scripts, set up the database manually, as described in section 6.1, “Setting JVM Options for Application Servers,” on page 41. After setting up the database manually, skip step 5 through step 8, and proceed to step 10.

3. (If you are using Oracle or MySQL, you can skip this step.) Copy your JDBC driver to Weblogic and set your classpath. For example, for PostgreSQL:

- a. Copy the JDBC jar from

<js-install>/buildomatic/conf\_source/db/postgresql/jdbc

to

<weblogic\_home>/server/lib

- b. Add the following line to your <weblogic\_home>/bin/setDomainEnv.cmd/sh file as follows:

```
set CLASSPATH=<weblogic_home>\server\lib\<jdbc-driver-name>.jar;%CLASSPATH% (Windows)
set CLASSPATH={pathToCommonsLangJar}\commons-lang-2.5.jar;%CLASSPATH% (Windows)
export CLASSPATH=<weblogic_home>/server/lib/<jdbc-driver-name>.jar:$CLASSPATH (Linux)
commons-lang-2.5.jar can be found in jasperserver WAR (WEB-INF/lib) (Linux)
```

- c. Save the file.

- d. Restart WebLogic using the startWebLogic.cmd/sh.

Note that the Oracle and MySQL JDBC drivers are included in recent versions of WebLogic by default.

4. If you are using Oracle, copy the database driver jar into the directory:  
.../Oracle/Middleware/user\_projects/domains/<user defined domain>/lib
5. Copy the .properties file for your database:
  - From — <js-install>/buildomatic/sample\_conf/
  - To — <js-install>/buildomatic
6. Rename the file you copied to default\_master.properties file.

7. Edit the default\_master.properties file to add the settings that are specific to your database and your application server. **Table 8-1** shown example property settings.



When appServerType = skipAppServerCheck, buildomatic skips the application servertype validation. Use this setting when installing JasperReports Server with WebLogic. Backslashes in appServerDir must be doubled, for example c:\WL\Application\_Server. Make sure there are no spaces in the appServerDir path.

**Table 8-1 Sample Values for the default\_master.properties File**

| Database   | Sample Property Values   |
|------------|--|
| PostgreSQL | <pre>appServerType=skipAppServerCheck appServerDir=[path to WebLogic application server] dbUsername=postgres dbPassword=postgres dbHost=localhost</pre>  |
| DB2        | <pre>appServerType=skipAppServerCheck appServerDir=[path to WebLogic application server] dbUsername=db2admin dbPassword=password dbHost=localhost</pre> <p>For DB2 8.x, change your deployed JDBC driver as described in section 6.4.3.1, “DB2 8.x Driver Example,” on page 54.</p>                        |
| MySQL      | <pre>appServerType=skipAppServerCheck appServerDir=[path to WebLogic application server] dbUsername=root dbPassword=password dbHost=localhost</pre>  |
| Oracle     | <pre>appServerType=skipAppServerCheck appServerDir=[path to WebLogic application server] sysUsername=system sysPassword=password dbUsername=jasperserver* dbPassword=password dbHost=hostname</pre>  |
| SQL Server | <pre>appServerType=skipAppServerCheck appServerDir=[path to WebLogic application server] dbUsername=sa dbPassword=sa dbHost=localhost</pre> <p>If your application server runs on Java 1.5, change your deployed JDBC driver as described in section 6.4.3.3, “SQL Server Driver Example,” on page 55.</p> |

\* The dbUsername must be the same as the Oracle user name.

8. Setup the database and optional sample databases using the buildomatic Ant scripts. Enter commands in the following table to call buildomatic Ant scripts:



Exception: For DB2, skip this step and perform [step 1](#) to [step 3](#) in section 6.3.4, “DB2,” on page 51, then go to the next step ([step 10](#)) of this procedure.

You call buildomatic Ant scripts from the command line using the following syntax:

Windows – js-ant <target-name>

Linux – `./js-ant <target-name>`

| Commands   | Description   |
|--|---|
| <code>cd &lt;js-install&gt;/buildomatic</code>   | Goes to the buildomatic directory.                      |
| <code>js-ant create-js-db</code><br><code>js-ant create-sugarcrm-db</code><br><code>js-ant create-foodmart-db</code> | Creates the JasperReports Server and sample databases.  |
| <code>js-ant load-sugarcrm-db</code><br><code>js-ant load-foodmart-db</code>   | (Optional) Loads sample data into the sample databases. |
| <code>js-ant init-js-db-pro</code><br><code>js-ant import-minimal-pro</code>   | Initializes database, loads core application data.      |
| <code>js-ant import-sample-data-pro</code>   | (Optional) Loads the demos that use the sample data.    |



On non-Linux Unix platforms, the `js-ant` commands may not be compatible with all shells. If you have errors, use the `bash` shell explicitly. For more information, see section [A.4, “Bash Shell for Solaris, IBM AIX, HP UX and FreeBSD,”](#) on page 147.

- Add the database driver to your classpath.
- In WebLogic, open an Administrative Console window, and navigate to **Services > Data Sources** or **Domain Configurations > Services > Data Sources**.
- Click **New** and then **Generic Data Source** for each of the data source columns in the following table, and enter the following values for a PostgreSQL database. You will need to click **Next** after entering the database driver and after **One-Phase Commit**.



To use a database other than PostgreSQL, configure the database connections using settings shown in section [8.3, “Configuring Other Database Connections,”](#) on page 84.

If you plan on using the sample databases (Foodmart and Sugar CRM), perform this step and the following step for each database.

| Parameter Name               | JasperReports Server   | Foodmart Example | Sugar CRM Example |
|------------------------------|--|------------------|-------------------|
| Name                         | JasperServerDataBase   | FoodmartDataBase | SugarcrmDataBase  |
| JNDI Name                    | JasperServerDataBase   | FoodmartDataBase | SugarcrmDataBase  |
| Database Type                | PostgreSQL   |                  |                   |
| Database Driver              | PostgreSQL Driver Versions: using <code>org.postgresql.Driver</code> |                  |                   |
| Supports Global Transactions | Selected   |                  |                   |
| One-Phase Commit             | Selected   |                  |                   |

- Set connection properties. Sample properties for a PostgreSQL database are:

| Parameter Name     | JasperReports Server | Foodmart Example | Sugar CRM Example |
|--------------------|----------------------|------------------|-------------------|
| Database Name      | jasperserver         | foodmart         | sugar crm         |
| Host Name          | localhost            |                  |                   |
| Port               | 5432                 |                  |                   |
| Database User Name | postgres             |                  |                   |



|                  |          |
|------------------|----------|
| Password         | postgres |
| Confirm Password | postgres |

13. Test the database connection:

- a. For SugarCRM and Foodmart, use the default connections:

jdbc:postgresql://localhost:5432/sugarcrm

jdbc:postgresql://localhost:5432/foodmart

- b. Change the URL for the jasperserver database to:

jdbc:postgresql://localhost:5432/jasperserver

14. Select targets, and ensure that **AdminServer** is set for all data sources.

15. Using the Java jar tool or an unzip tool, unpack the jasperserver-pro.war file. For example, enter these commands to use the Java jar tool:

```
cd <js-install-dir>
mkdir jasperserver-pro
cd jasperserver-pro
"%JAVA_HOME%/bin/jar" xvf ../jasperserver-pro.war
```

The jasperserver-pro.war file is unpacked to a folder.

16. Delete JARs to avoid conflict with WebLogic JARs:

- a. Go to the WEB-INF/lib directory:

```
cd jasperserver-pro/WEB-INF/lib
```

- b. Delete the following conflicting JARs, but save a copy of them to a temporary location:

jaxb-api-<ver>.jar

jaxb-impl-<ver>.jar

serializer-<ver>.jar

stax-api-<ver>.jar

xalan-<ver>.jar

xercesImpl-<ver>.jar

xml-apis-<ver>.jar

If you experience a problem with OLAP Views, you need these JARs for the workaround described in section [8.10](#), **“Troubleshooting Problems with OLAP Views,”** on page 87.

17. Update your Hibernate, Quartz, and Mail Server configuration:

- a. Open the file jasperserver-pro/WEB-INF/hibernate.properties for editing and locate the metadata.hibernate.dialect property. Set its value according to your database:

If you are using PostgreSQL, the information is already set correctly, so you can skip this step.

MySQL 5.1 metadata.hibernate.dialect=org.hibernate.dialect.MySQLInnoDBDialect

MySQL 5.5 metadata.hibernate.dialect=org.hibernate.dialect.MySQL5InnoDBDialect

DB2: metadata.hibernate.dialect=com.jaspersoft.ji.hibernate.dialect.DB2JICustomDialect

Oracle: metadata.hibernate.dialect=com.jaspersoft.ji.hibernate.dialect.OracleJICustomDialect

SQL Server metadata.hibernate.dialect=com.jaspersoft.ji.hibernate.dialect.SQLServerJICustomDialect

- b. Open the file jasperserver-pro/WEB-INF/js.quartz.properties for editing and locate the report.scheduler.web.deployment.uri property. Set its value to the hostname, port, and path where you intend to deploy JasperReports Server, for example:

report.scheduler.web.deployment.uri=http://localhost:7001/jasperserver-pro

If you are using PostgreSQL or DB2, set the Quartz delegate class in the `js.quartz.properties` file:

PostgreSQL `quartz.delegateClass=org.quartz.impl.jdbcjobstore.PostgreSQLDelegate`

DB2: `quartz.delegateClass=org.quartz.impl.jdbcjobstore.DB2v8Delegate`

All others: `quartz.delegateClass=org.quartz.impl.jdbcjobstore.StdJDBCDelegate`

- c. If you want to configure JasperReports Server to automatically schedule and email reports, enter your mail server information in the `js.quartz.properties` file. Modify all `report.scheduler.mail.sender.*` properties as necessary for your mail server.
18. If your mail server requires authentication, edit the `applicationContext-report-scheduling.xml` file in the same manner:
  - a. Open the `jasperserver-pro/WEB-INF/applicationContext-report-scheduling.xml` file for editing and locate the `reportSchedulerMailSender` bean.
  - b. Set the `javaMailProperties` key="mail.smtp.auth" value to `true`.
19. Now you can change to the `jasperserver-pro` folder and rearchive the `jasperserver-pro.war` file, using commands such as the following:

| Commands  | Description  |
|---|--|
| <code>cd ../../</code>  | Changes to the <code>jasperserver-pro</code> folder          |
| <code>mv ../jasperserver-pro.war ../BAK-jasperserver-pro.war</code>         | Renames the original <code>jasperserver-pro.war</code> file. |
| <code>"%JAVA_HOME%/bin/jar" cvf ../jasperserver-pro.war *</code>            | Rearchives the <code>jasperserver-pro.war</code> file.       |
| <code>cd ..</code><br><code>mv jasperserver-pro BAK-jasperserver-pro</code> | Renames the unneeded working folder to a backup location.    |



You now have a `jasperserver-pro.war` file that can be used for deploying to WebLogic.

20. Edit your WebLogic domain configuration file `<wl-domain>/config/config.xml`:



`<wl-domain>` is the path of the domain within WebLogic that contains your JasperReports Server deployment. For example `<weblogic>/samples/domains/wl_server`.

- a. Locate the `server` and `security-configuration` elements, and insert the following parameters:

```
<server>
...
  <stuck-thread-max-time>1200</stuck-thread-max-time>
  <listen-address></listen-address>
</server>

<security-configuration>
...
  <enforce-valid-basic-auth-credentials>false</enforce-valid-basic-auth-credentials>
</security-configuration>
```

- b. Check that the `stuck-thread-max-time` element appears above the `listen-address` element before the closing `</server>` tag.
21. Copy the file `<js-install>/buildomatic/conf_source/iePro/lib/xercesImpl-2.10.0.jar` to the folder `<wl-domain>/lib`.

22. Edit the `setDomainEnv` file to force WebLogic to use the correct version of the Joda time library, which is shipped with JasperReports Server. By default, the Joda time library is in:

```
<js-install>/buildomatic/conf_source/iePro/lib.
```

- ♦ Windows: Edit the `setDomainEnv.bat` file and add the following line:  
`set PATCH_CLASSPATH=${PATH TO NEW JODA LIBRARY}\joda-time-2.1.jar`
- ♦ Linux: Edit the `setDomainEnv.sh` file and add the following line:  
`export PATCH_CLASSPATH=${PATH TO NEW JODA LIBRARY}/joda-time-2.1.jar`

23. Set JVM options as described in section 8.2, “Setting Java Properties,” on page 83.

#### Deploy JasperReports Server to WebLogic:

1. Enable the **Lock & Edit** button:
  - a. Select the **Preferences** link at the top of the Admin console
  - b. On the **User Preferences** screen, scroll to the bottom of the screen and deselect **Automatically Acquire Lock** and **Activate Changes**.
  - c. Save.
2. In the Administrative Console, click the **Lock & Edit** button and navigate to **Deployments**.
3. On the Deployments page click the **Install** button.
4. Select the path to `<js-install>`. Click **Next**.
5. Leave the radio button selected for **Install this deployment as an application**. Click **Next**.
6. When prompted, enter the following parameter values:

| Parameter Name       | Example Value  |
|----------------------|--|
| Name                 | jasperserver-pro                                     |
| Security             | Custom Roles and Policies                            |
| Source accessibility | Use the defaults defined by the deployment's targets |

7. Review your choices and click **Finish**.
8. Click **Save**.

## 8.2 Setting Java Properties

Edit your WebLogic startup script to include the settings described in the following tables, according to your platform. Substitute the location of your JasperReports Server license file where necessary:

| WebLogic Startup Settings on Windows |                                   |
|--------------------------------------|-----------------------------------|
| Filename                             | <wl-domain>\bin\startWebLogic.cmd |

| WebLogic Startup Settings on Windows |  |
|--------------------------------------|--|
| Settings                             | set JAVA_OPTIONS=%JAVA_OPTIONS%<br>-Djs.license.directory=C:\<js-install>\<br>-Dfile.encoding=UTF-8<br>-Dcom.sun.xml.namespace.QName.useCompatibleSerialVersionUID=1.0<br>-Djavax.xml.soap.SOAPConnectionFactory=org.apache.axis.soap.SOAPConnectionFactoryImpl<br>-Djavax.xml.soap.MessageFactory=org.apache.axis.soap.MessageFactoryImpl<br>-Djavax.xml.soap.SOAPFactory=org.apache.axis.soap.SOAPFactoryImpl<br>-Xms1024m<br>-Xmx2048m<br>-XX:PermSize=128m<br>-XX:MaxPermSize=512m<br>-Xss2m |
| For Oracle *                         | set JAVA_OPTIONS=%JAVA_OPTIONS% -Doracle.jdbc.defaultNChar=true  |

\* Setting the Oracle localization option, defaultNChar, can substantially impact the performance of JDBC queries. If you do not need to support UTF-8 for your Oracle database, you can omit this setting.

| WebLogic Startup Settings on Linux |  |
|------------------------------------|--|
| Filename                           | <wl-domain>/bin/startWebLogic.sh   |
| Settings                           | export JAVA_OPTIONS="\$JAVA_OPTIONS<br>-Djs.license.directory=/home/<user>/weblogic/jasperlicense/<br>-Dfile.encoding=UTF-8<br>-Dcom.sun.xml.namespace.QName.useCompatibleSerialVersionUID=1.0<br>-Djavax.xml.soap.SOAPConnectionFactory=org.apache.axis.soap.SOAPConnectionFactoryImpl<br>-Djavax.xml.soap.MessageFactory=org.apache.axis.soap.MessageFactoryImpl<br>-Djavax.xml.soap.SOAPFactory=org.apache.axis.soap.SOAPFactoryImpl<br>-Xms1024m<br>-Xmx2048m<br>-XX:PermSize=128m<br>-XX:MaxPermSize=512m<br>-Xss2m |
| For Oracle *                       | export JAVA_OPTIONS="\$JAVA_OPTIONS -Doracle.jdbc.defaultNChar=true"   |

\* Setting the Oracle localization option, defaultNChar, can substantially impact the performance of JDBC queries. If you do not need to support UTF-8 for your Oracle database, you can omit this setting.

## 8.3 Configuring Other Database Connections

Use these settings to connect to a database other than PostgreSQL.

| Database Setting | MySQL        | Oracle       | DB2       | SQL Server   |
|------------------|--------------|--------------|-----------|--------------|
| Host             | localhost    | localhost    | localhost | localhost    |
| Name or SID      | jasperserver | Orcl         | jsprsvr   | jasperserver |
| User             | root         | jasperserver | db2admin  | sa           |
| Password         | password     | password     | password  | sa           |
| Port             | 3306         | 1521         | 50000     | 1433         |

|                        |                     |                        |                    |                          |
|------------------------|---------------------|------------------------|--------------------|--------------------------|
| Hibernate Dialect      | MySQLInnoDB Dialect | OracleJICustom Dialect | DB2JICustomDialect | SQLServerJICustomDialect |
| Quartz Driver Delegate | StdJDBCDelegate     | StdJDBCDelegate        | DB2v8Delegate      | StdJDBCDelegate          |

## 8.4 Starting JasperReports Server

1. In the Administrative Console, navigate to **Deployments**.
2. Select the **jasperserver-pro** application with a checkmark, and click **Start**.
3. In the Start Application Assistant page, click **Yes**.

## 8.5 Logging into JasperReports Server

If JasperReports Server started up cleanly you should be able to login.

1. Login by going to the following URL:

`http://<hostname>:7001/jasperserver-pro`

Where <hostname> could be localhost, a machine name, or an IP address. The login page should appear after some time to compile the necessary JSP files.

2. Use the following credentials to log into the system:

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

If you logged in successfully, the JasperReports Server home page appears.



The first time you log into JasperReports Server, you will be prompted to opt-in or opt-out of the JasperReports Server Heartbeat. For more information, refer to section [5.5.1, “JasperReports Server Heartbeat,” on page 38](#).

Refer to the *JasperReports Server User Guide* to begin adding reports and other resources to JasperReports Server.

## 8.6 Configuring Report Scheduling

The scheduled reporting feature of JasperReports Server allows reports to be run at pre-configured time intervals. Optionally, notification emails can be set to users to let them know that new reports are available.

For more information about setting up report scheduling, refer to section [6.5, “Configuring Report Scheduling,” on page 58](#).

## 8.7 Restarting JasperReports Server

If you made configuration changes to your JasperReports Server instance, restart JasperReports Server.

## 8.8 Updating XML/A Connection Definitions (Optional)

If you loaded the JasperReports Server sample data and would like to run the XML/A examples, update the XML/A connection resources to use the correct web port.

The typical port used by WebLogic is 7001. Follow the procedure in section [6.6, “Updating XML/A Connection Definitions,” on page 61](#).

## 8.9 Troubleshooting Your JasperReports Server Configuration

If you have problems running buildomatic scripts that set up the database, set up the database manually. For more information, see section [6.1, “Setting JVM Options for Application Servers,” on page 41](#).

### 8.9.1 Startup Problems

When trying to run a new JasperReports Server instance, most typical problems encountered are errors in the database configuration.

These problems are typically related to having incorrect configurations within the database configuration files or in the application server configuration files. For more information about resolving these types of errors, refer to troubleshooting section [Appendix A, “Troubleshooting,” on page 145](#).

### 8.9.2 Error Running Report

If you have trouble running reports in your new JasperReports Server instance, refer to troubleshooting section [A.8.11, “Error Running a Report,” on page 153](#).

## 8.10 Troubleshooting Problems with OLAP Views

If you are using WebLogic 10.3.2 or lower, and you encounter a 404 Error when attempting to change a data cube in an OLAP View, follow these steps to work around the problem:

1. Add the following java option to `startWebLogic.cmd` or `startWebLogic.sh`:

```
Windows:  set  JAVA_OPTS=%JAVA_OPTS% -Djavax.xml.transform.TransformerFactory=
           org.apache.xalan.processor.TransformerFactoryImpl

Linux:    export JAVA_OPTS="$JAVA_OPTS -Djavax.xml.transform.TransformerFactory=
           org.apache.xalan.processor.TransformerFactoryImpl"
```

2. Set `EXT_PRE_CLASSPATH` in `setDomainEnv.cmd`:

```
Windows:  set
           EXT_PRE_CLASSPATH=%DOMAIN_HOME%/lib/serializer-2.7.1.jar;
           %DOMAIN_HOME%/lib/xalan-2.7.1.jar;
           %DOMAIN_HOME%/lib/xercesImpl-2.10.0.jar;
           %DOMAIN_HOME%/lib/dom4j-1.6.1.jar;
           %DOMAIN_HOME%/lib/jdom-1.1.jar;
           %EXT_PRE_CLASSPATH%

Linux:    export
           EXT_PRE_CLASSPATH="$DOMAIN_HOME/lib/serializer-2.7.1.jar:
           $DOMAIN_HOME/lib/xalan-2.7.1.jar:
           $DOMAIN_HOME/lib/xercesImpl-2.10.0.jar:
           $DOMAIN_HOME/lib/dom4j-1.6.1.jar:
           $DOMAIN_HOME/lib/jdom-1.1.jar:
           $EXT_PRE_CLASSPATH"
```

3. Copy the JAR files listed in the previous step from the `jasperserver-pro.war/WEB-INF/lib` folder to `%DOMAIN_HOME%/lib` or `$DOMAIN_HOME/lib`.





## CHAPTER 9 OVERLAY UPGRADE

---

This chapter describes how to upgrade to JasperReports Server 5.5 using the new overlay upgrade package. This chapter contains the following sections:

- **Introduction to the Overlay Upgrade**
- **Upgrade Steps Overview**
- **Back Up Your JasperReports Server Instance**
- **Unpack the Overlay Upgrade Package**
- **Run the Overlay Upgrade**
- **Rerun the Overlay Upgrade**
- **Rollback Procedure**
- **Starting and Logging into JasperReports Server 5.5**
- **Additional Tasks to Complete the Upgrade**
- **Running Overlay Upgrade a Second Time**

### 9.1 Introduction to the Overlay Upgrade

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



Currently, the **Overlay Upgrade only supports Apache Tomcat as the application server**. 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

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 5.2 to 5.5 upgrade as an example.

## 9.2 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 5.5 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.

## 9.3 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_52_BACKUP` or `/opt/JS_52_BACKUP`.
2. Copy `<tomcat>/webapps/jasperserver-pro` 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:

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

## 9.4 Unpack the Overlay Upgrade Package

The overlay upgrade package comes in a file named: `jasperreports-server-overlay-5.5.zip`.

1. Download the overlay upgrade package from [Jaspersoft technical support](http://support.jaspersoft.com) (<http://support.jaspersoft.com>) or contact your sales representative.
2. Extract all files from `jasperreports-server-overlay-5.5.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>`

## 9.5 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 5.2 instance it might be similar to:  
C:\Jaspersoft\jasperreports-server-5.2\buildomatic\default\_master.properties  
/opt/jasperreports-server-5.2/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 Login to test the upgraded JasperReports Server  
If execution was successful, you will see BUILD SUCCESSFUL on the command line

## 9.6 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
```

## 9.7 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_52_BACKUP` or `/opt/JS_52_BACKUP`.
2. Run the following commands for PostgreSQL:  

```
cd /opt/JS_52_BACKUP  
pg_restore --username=postgres jasperserver < js-db-5.2-dump.sql
```

## 9.8 Starting and Logging into JasperReports Server 5.5

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

### 9.8.1 Clearing Your Browser Cache

Before you log in, make sure you and your end-users clear the browser cache. JavaScript files, which enable 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.

### 9.8.2 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 5.5. In the event of startup or login problems, refer to [Appendix A, “Troubleshooting,” on page 145](#)

## 9.9 Additional Tasks to Complete the Upgrade

For additional information about optional steps after the main upgrade steps are complete, see section [11.9, “Additional Tasks to Complete the Upgrade,” on page 107](#).

### 9.9.1 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.

## 9.10 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 10 UPGRADING FROM 5.2 TO 5.5

---

This chapter describes how to upgrade from JasperReports Server 5.2 to JasperReports Server 5.5.

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

This chapter contains the following sections:

- [Upgrade Steps Overview](#)
- [Back Up Your JasperReports Server Instance](#)
- [Preparing the JasperReports Server 5.5 WAR File Distribution](#)
- [Configuring Buildomatic for Your Database and Application Server](#)
- [Upgrading to JasperReports Server 5.5](#)
- [Starting and Logging into JasperReports Server 5.5](#)
- [Additional Tasks to Complete the Upgrade](#)
- [Old Manual Upgrade Steps: 5.2 to 5.5](#)

### 10.1 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 5.5 JasperReports Server WAR file distribution zip.
3. Run the js-upgrade script as described in [10.5, “Upgrading to JasperReports Server 5.5,” on page 98](#).

If your current instance of JasperReports Server has any custom modifications or extensions, keep track of these and re-integrate them into your 5.5 instance after upgrading.

### 10.2 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\_52\_BACKUP or /opt/JS\_52\_BACKUP.
2. Copy <tomcat>/webapps/jasperserver-pro 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 you 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 section [A.8.3, “Maximum Packet Size in MySQL,”](#) on page 151.

## 10.3 Preparing the JasperReports Server 5.5 WAR File Distribution

Use the buildomatic js-upgrade scripts included in the 5.5 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-5.5-bin.zip in the compressed ZIP format. Download the WAR file distribution from [Jaspersoft technical support](http://support.jaspersoft.com) (<http://support.jaspersoft.com>) or contact your sales representative.
2. Extract all files from jasperreports-server-5.5-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-5.5>
```

## 10.4 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 [“Bash Shell for Solaris, IBM AIX, HP UX and FreeBSD”](#) on page 147 for more information.

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

### 10.4.1 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.



### 10.4.1.1 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-5.5>/buildomatic/sample_conf/postgresql_master.properties |

2. Copy the file to `<js-install-5.5>/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 | <pre>appServerType=tomcat6 [tomcat7, tomcat6, jboss, glassfish2, glassfish3] appServerDir=c:\apache-tomcat-6.0.26 (for example) dbUsername=postgres dbPassword=postgres dbHost=localhost</pre> |

### 10.4.1.2 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-5.5>/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    | <pre>appServerType=tomcat6 [tomcat7, tomcat6, jboss, glassfish2, glassfish3] appServerDir=c:\Apache Software Foundation\Tomcat-6 (for example) dbUsername=root dbPassword=password dbHost=localhost</pre> |

### 10.4.1.3 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-5.5>/buildomatic/sample_conf/oracle_master.properties |

2. Copy the file to `<js-install-5.5>/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   | <pre>appServerType=tomcat6 [tomcat7, tomcat6, jboss, glassfish2, glassfish3] appServerDir=c:\Apache Software Foundation\Tomcat-6 (for example) dbUsername=jasperserver dbPassword=password dbHost=localhost</pre> |

## 10.5 Upgrading to JasperReports Server 5.5

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   |
|--|---|
| <code>cd &lt;js-install-5.5&gt;/buildomatic</code> |   |
| <code>js-upgrade-samedb.bat</code>                 | (Windows) Upgrade jasperserver-pro war file, upgrade jasperserver database to 5.5, add 5.5 repository resources into the database |
| <code>./js-upgrade-samedb.sh</code>                | (Linux) Upgrade jasperserver-pro war file, upgrade jasperserver database to 5.5, add 5.5 repository resources into the database   |

### 10.5.1 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-5.5>/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.

### 10.5.2 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-5.5>/buildomatic/logs/js-upgrade-<date>-<number>.log
```

### 10.5.3 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 [Appendix A, “Troubleshooting,” on page 145](#). The information in this section 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.

## 10.6 Starting and Logging into JasperReports Server 5.5

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

### 10.6.1 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.

### 10.6.2 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 5.5. In the event of startup or login problems, refer to [Appendix A, “Troubleshooting,” on page 145](#)

## 10.7 Additional Tasks to Complete the Upgrade

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

### 10.7.1 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.

### 10.7.2 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

### 10.7.3 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;
```

## 10.8 Old Manual Upgrade Steps: 5.2 to 5.5

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  |
|--|--|
| <code>cd &lt;js-install-5.5&gt;/buildomatic</code> |  |
| <code>js-ant upgrade-5.2-5.5-pro</code>            | Execute SQL script to upgrade database to 5.5.<br>Executes script <code>buildomatic/install_resources/sql/&lt;dbType&gt;/upgrade-&lt;dbType&gt;-5.2.0-5.5.0-pro.sql</code> |
| <code>js-ant import-minimal-for-upgrade-pro</code> | Loads themes and other core resources for 5.5.   |
| <code>js-ant import-sample-data-upgrade-pro</code> | (Optional) This step is optional. Loads the 5.5 sample data.   |
| <code>js-ant deploy-webapp-pro</code>              | Delete old 5.2 war file, deploy the 5.5 war file.  |

## CHAPTER 11 UPGRADING FROM 3.7 - 5.2 TO 5.5

---

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 or 5.1 to JasperReports Server 5.5. 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 5.2 to 5.5. However, we recommend you use the procedure in [Chapter 10, “Upgrading from 5.2 to 5.5,” on page 95](#).

In this chapter the examples shown will use JasperReports Server 5.1 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 5.5 WAR File Distribution](#)
- [Configuring Buildomatic for Your Database and Application Server](#)
- [Upgrading to JasperReports Server 5.5](#)
- [Starting and Logging into JasperReports Server 5.5](#)
- [Additional Tasks to Complete the Upgrade](#)
- [Old Manual Upgrade Steps](#)

## 11.1 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 5.1 data.
4. Download and set up the new 5.5 JasperReports Server WAR file distribution zip.
5. Run the js-upgrade script as described in section [11.7, “Upgrading to JasperReports Server 5.5,” on page 106](#).

If your current instance of JasperReports Server has any custom modifications or extensions, keep track of these and re-integrate them into your 5.5 instance after upgrading.

## 11.2 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: [Chapter 12, “Planning Your Upgrade from 4.7,” on page 109](#).

## 11.3 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_52_BACKUP` or `/opt/JS_52_BACKUP`.
2. Copy `<tomcat>/webapps/jasperserver-pro` 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 section [A.8.3, “Maximum Packet Size in MySQL,” on page 151](#).

## 11.4 Exporting Current Repository Data

You need to export your old repository data, for example your 5.1 repository data, using the JasperReports Server export utility. There are three ways to export:

- 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.

### 11.4.1 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 5.1 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 section [16.5.1.2, "Workaround for Export bug in the 5.1 Release,"](#) 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.

### 11.4.2 Using Buildomatic Scripts to Export Data

If you configured buildomatic and your `default_master.properties` file as described in section [16.3.1, "Creating a default\\_master.properties File,"](#) on page 138, you can export your repository data. For example, to export 5.1 repository data, use the following commands:

1. Navigate to the buildomatic directory:

```
cd <js-install-5.1>/buildomatic
```

2. Run buildomatic with the export target:

Windows: `js-ant.bat export-everything -DexportFile=js-5.1-export.zip`

Linux: `./js-ant export-everything -DexportFile=js-5.1-export.zip`



Note the location of this export file so that you can use it during the 5.5 upgrade process.

### 11.4.3 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-5.1>/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-5.1>/buildomatic`
2. Run the js-export script:

Windows: `js-export.bat --everything --output-zip js-5.1-export.zip`

Linux: `js-export.sh --everything --output-zip js-5.1-export.zip`



Note the location of this export file so that you can use it during the 5.5 upgrade process.

## 11.5 Preparing the JasperReports Server 5.5 WAR File Distribution

Use the buildomatic js-upgrade scripts included in the 5.5 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-5.5-bin.zip` in the compressed ZIP format. Download the WAR file distribution from [Jaspersoft technical support](http://support.jaspersoft.com) (<http://support.jaspersoft.com>) or contact your sales representative.
2. Extract all files from `jasperreports-server-5.5-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-5.5>`

## 11.6 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 “[Bash Shell for Solaris, IBM AIX, HP UX and FreeBSD](#)” on [page 147](#) for more information.

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

### 11.6.1 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`.

#### 11.6.1.1 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 | <code>&lt;js-install-5.5&gt;/buildomatic/sample_conf/postgresql_master.properties</code> |

2. Copy the file to `<js-install-5.5>/buildomatic`
3. Rename the file `default_master.properties`



- Edit `default_master.properties` for your database and application server:

| Database   | Sample Property Values  |
|------------|---|
| PostgreSQL | <pre>appServerType=tomcat6 [tomcat7, tomcat5, jboss, glassfish2, glassfish3] appServerDir=c:\Apache Software Foundation\Tomcat 6 dbUsername=postgres dbPassword=postgres dbHost=localhost</pre> |

#### 11.6.1.2 MySQL Example

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

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

| Database | Master Properties File  |
|----------|---|
| MySQL    | <code>&lt;js-install-5.5&gt;/buildomatic/sample_conf/mysql_master.properties</code> |

- Copy the file to `<js-install-5.5>/buildomatic`
- Rename the file `default_master.properties`
- Edit `default_master.properties` for your database and application server:

| Database | Sample Property Values  |
|----------|---|
| MySQL    | <pre>appServerType=tomcat6 [tomcat7, tomcat5, jboss, glassfish2, glassfish3] appServerDir=C:\Apache Software Foundation\Tomcat 6 dbUsername=root dbPassword=password dbHost=localhost</pre> |

#### 11.6.1.3 Oracle Example

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

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

| Database | Master Properties File   |
|----------|--|
| Oracle   | <code>&lt;js-install-5.5&gt;/buildomatic/sample_conf/oracle_master.properties</code> |

- Copy the file to `<js-install-5.5>/buildomatic`
- Rename the file to `default_master.properties`
- Edit `default_master.properties` for your database and application server:

| Database | Sample Property Values  |
|----------|---|
| Oracle   | <pre>appServerType=tomcat6 [tomcat7, tomcat6, jboss, glassfish2, glassfish3] appServerDir=c:\Apache Software Foundation\Tomcat-6 (for example) sysUsername=system sysPassword=password dbUsername=jasperserver dbPassword=password dbHost=localhost</pre> |

## 11.7 Upgrading to JasperReports Server 5.5

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  |
|---|--|
| <code>cd &lt;js-install-5.5&gt;/buildomatic</code>                | Change to buildomatic directory  |
| <code>js-upgrade-newdb.bat &lt;path&gt;\js-5.1-export.zip</code>  | (Windows) Upgrade jasperserver-pro war file, drop and recreate the database, import data file from previous version. |
| <code>./js-upgrade-newdb.sh &lt;path&gt;/js-5.1-export.zip</code> | (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 section [A.8.3, “Maximum Packet Size in MySQL,”](#) on page 151.



If you have auditing enabled, see section [A.12.2, “Include Audit Events on Upgrade,”](#) on page 162 for information on including auditing events.

### 11.7.1 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-5.5>/buildomatic
js-upgrade-newdb.bat test <path>/js-5.1-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.

### 11.7.2 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-5.5>/buildomatic/logs/js-upgrade-<date>-<number>.log
```

### 11.7.3 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 section [Appendix A, “Troubleshooting,”](#) on page 145. The information in this section 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.

## 11.8 Starting and Logging into JasperReports Server 5.5

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

### 11.8.1 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.

### 11.8.2 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 5.5. In the event of startup or login problems, refer to [Appendix A, “Troubleshooting,” on page 145](#).

## 11.9 Additional Tasks to Complete the Upgrade

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

### 11.9.1 Handling JasperReports Server Customizations

If you made modifications or customizations to the original JasperReports Server application, JasperReports Server 5.1 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.

### 11.9.2 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.

### 11.9.3 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

### 11.9.4 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;
```

## 11.10 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 5.1 upgrade as an example):

| Commands   | Description  |
|--|--|
| <code>cd &lt;js-install-5.5&gt;/buildomatic</code>   |  |
| <code>js-ant drop-js-db</code><br><code>js-ant create-js-db</code><br><code>js-ant init-js-db-pro</code> | This will delete and recreate your jasperserver db. Make sure your original database is backed up.   |
| <code>js-ant import-minimal-pro</code>   |  |
| <code>js-ant import-upgrade</code><br><code>-DimportFile=&lt;path-and-filename&gt;</code>                | The <code>-DimportFile</code> should point to the <code>&lt;path&gt;</code> and <code>&lt;filename&gt;</code> of the <code>js-5.1-export.zip</code> file you created earlier.<br>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. |
| <code>js-ant import-sample-data-upgrade-pro</code>   | (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.   |
| <code>js-ant deploy-webapp-pro</code>  | Delete the existing older war file, deploy the new war file.   |

## CHAPTER 12 PLANNING YOUR UPGRADE FROM 4.7

---

If you are upgrading from JasperReports Server 4.7 or earlier you should take a look at the information included in this chapter.

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

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

This chapter describes only those changes that can significantly impact your existing deployment.



These upgrade steps are for the WAR installation.

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:

- [Understanding the Changes in 4.7](#)
- [Migrating Ad Hoc Reports to Ad Hoc Views](#)
- [Upgrading Themes in 4.7](#)
- [Data Snapshots](#)
- [Understanding the Changes in 5.0](#)
- [Upgrading Preserving Custom Settings](#)
- [XML/A Connections to Servers with Multiple Organizations](#)

### 12.1 Understanding the Changes in 4.7

The following changes in 4.7 can significantly 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 5.0 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 persevered. 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 110** 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 5.0. See **“Upgrading Themes in 4.7” on page 113** 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 110** for more information on data snapshots.

## 12.2 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.

### 12.2.1 Changes to Ad Hoc

**Table 12-1** summarizes the differences between Ad Hoc reports and Ad Hoc views.

**Table 12-1 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.   |

**Table 12-1 Comparison of Ad Hoc Reports and Ad Hoc Views**

| Ad Hoc 4.5.1 and earlier   | Ad Hoc in 4.7  |
|--|--|
| 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.<br>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.   |

### 12.2.2 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 5.0 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

### 12.2.3 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 5.0 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.

## 12.2.4 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 5.0 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.

### 12.2.4.1 Working with Reports Modified in iReport or Jaspersoft Studio

During migration, the pre-5.0 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.

## 12.2.5 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.

### 12.2.5.1 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 113** for information on how to migrate these reports manually.

### 12.2.5.2 Finding Ad Hoc Reports in the Repository

Reports that did not migrate successfully show up in the 5.0 or newer 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.

### 12.2.5.3 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.

## 12.2.6 Troubleshooting

### 12.2.6.1 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**.

### 12.2.6.2 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-5.0 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.

## 12.3 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*.

### 12.3.1 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 12-2** shows the elements you need to modify and their default locations. The default values for these elements are in the `default.css` file.

**Table 12-2 Banner and Toolbar Settings**

| Element                                | Classname and Modification   | File                        | Notes   |
|--|--|-----------------------------|---|
| Banner                                 | <code>.banner</code><br>Give custom value to <code>height</code> .   | <code>containers.css</code> | Default value:<br><code>height: 26px</code>   |
| Banner Logo                            | <code>#logo</code><br>Give custom values to <code>height</code> and <code>width</code> that match the dimensions of your logo.   | <code>theme.css</code>      | Default values:<br><code>height: 20px</code><br><code>width: 115px</code>   |
| Banner Main Navigation                 | <code>.menu.primaryNav .wrap</code><br>Set <code>height</code> and <code>line-height</code> to the same measurement as <code>.banner</code> .  | <code>containers.css</code> | Default values:<br><code>height: 26px</code><br><code>line-height: 26px</code>  |
| Banner Main Navigation                 | <code>.menu.primaryNav .wrap.over</code><br><code>.menu.primaryNav .wrap.pressed</code><br>If you already have values for these defined in your alternate theme, then you need to change <code>height</code> and <code>line-height</code> to match the height of <code>.banner</code> .                            | <code>containers.css</code> | Not explicitly defined, but a value of 26px for both elements is cascaded from <code>.menu.primaryNav .wrap</code> . If the <code>over</code> and <code>pressed</code> effects fill the banner height after applying the modifications to <code>.menu.primaryNav .wrap</code> , then this step is not necessary.  |
| Banner Main Navigation Home icon       | <code>.menu.primaryNav #main_home .wrap &gt; .icon</code><br>Set <code>height</code> to be 2px shorter than the height of <code>.banner</code> . Set values for <code>width</code> and <code>background-position</code> to fit your image.   | <code>containers.css</code> | Default value:<br><code>background-position: left -166px</code> .<br>Height should be two pixels shorter than banner height. Height is not explicitly defined, but a value of 24px is cascaded from <code>.button .icon</code> in <code>buttons.css</code> . Width is not explicitly defined, but a value of 14px is cascaded from <code>.menu.primaryNav #main_home .wrap &gt; .icon</code> in the <code>containers.css</code> file. |
| Banner Main Navigation Item arrow icon | <code>.menu.primaryNav .node &gt; .wrap &gt; .icon</code><br>Set <code>height</code> to your desired value, with the maximum value being the same height measurement as the <code>.banner</code> element. Set <code>background-position</code> to a value that properly displays the default or your custom image. | <code>containers.css</code> | Default values:<br><code>height: 30px</code> ;<br><code>background-position: 0 -78px</code><br><code>background-position: 0 -78px (IE8-9)</code><br><code>background-position: 0 -79px (Ch+Saf)</code><br><br>If you are using Chrome or Safari, the classname that you apply background position to must be preceded by <code>"body:nth-of-type(1)"</code> .   |

**Table 12-2 Banner and Toolbar Settings**

| Element                                      | Classname and Modification  | File                        | Notes   |
|--|---|-----------------------------|---|
| Banner<br>Main Navigation<br>Item arrow icon | <code>.menu.primaryNav .wrap.over</code><br><code>.menu.primaryNav .wrap.pressed</code><br>Set <code>background-position</code> to a value that properly displays the default or your custom image. | <code>containers.css</code> | <code>background-position</code> is not explicitly defined. The value is cascaded from <code>.menu.primaryNav .node &gt; .wrap &gt; .icon</code><br>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<br>Metadata                           | <code>#metalinks li</code><br>Set <code>line-height</code> to the desired value that will vertically center it within the banner.   | <code>theme.css</code>      | Default value:<br><code>line-height: 9px</code>   |
| Banner<br>Search<br>container                | <code>#globalSearch.searchLockup</code><br>Set <code>margin-top</code> to desired value that will vertically center it within the banner.   | <code>controls.css</code>   | Default value:<br><code>margin-top: 3px</code>  |
| Body   | <code>#frame</code><br>Set a custom <code>top</code> value that positions the body of the application below the banner.   | <code>containers.css</code> | Default value:<br><code>top: 28px</code><br>This value needs to be greater than the value you apply to the height of <code>.banner</code> .   |
| Toolbar                                      | <code>.toolbar</code><br>Set custom height value that will fit your capsule buttons.  | <code>containers.css</code> | Default value:<br><code>height: 28px</code><br>This is only necessary if you have customized capsule buttons that differ in height from the default images.   |
| Footer                                       | <code>#frameFooter</code><br>Rename ID in stylesheet  | <code>containers.css</code> | The <code>#footer</code> element was renamed to <code>#frameFooter</code> . Change style rules for <code>#footer</code> to <code>#frameFooter</code> .  |

### 12.3.2 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 12-3** shows an example of an IE7 style rule that has been combined into the base style rules in 5.0 and newer.

: **Table 12-3 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)   |
|--|---|---|
| <pre>.stepIndicator .icon {     *top: 2px; }</pre> | <pre>.stepIndicator .icon {     margin: 0;     top: 1px;     margin-right: 3px; }</pre> | <pre>.stepIndicator .icon {     position: relative;     margin: 0px;     top: 1px;     *top: 2px; \\\* IE7 \\\     margin-right: 3px; }</pre> |

### 12.3.3 Images in JasperReports Server 4.7

Images for navigation, buttons, and backgrounds in the default theme have changed in version 5.0 and newer. 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 12-4** lists the images that have been added in the 4.7 default theme.

**Table 12-4 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 <code>.header</code> elements in columns and dialogs | Replaces <code>panel_dialog_header_sprite.png</code> image used for dialog box headers in 4.5.1. |
| 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 12-5** lists the images that have been modified for the 5.0 and newer default theme.

**Table 12-5 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 <code>.action.primary</code> 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 <code>.action.square</code> buttons                                | All icons have new shadow effect.                     |

**Table 12-5 Modified Images in 4.7 Default Theme**

| Image                            | Usage   | Notes  |
|----------------------------------|---|--|
| 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 <code>.button.capsule.indicator</code> .              |
| home_bkgd.png                    | Large background image on home page   | Image refactored.  |
| home_icons_sprite.png            | Large icons for <code>.action.jumbo</code> button on home and admin home pages  | Icons for <b>View Reports</b> and <b>Create Ad Hoc View</b> 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 12-6** lists the images that have not been affected by the refactoring of themes in 5.0 and newer.

**Table 12-6 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  |
| list_pressed_bkgd.png           | Background image for the top inner shadow that appears for the pressed state of a <code>.list.filter</code> list item |

**Table 12-6 Images With No Modifications in 4.7**

| 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 <code>.tabSet.horizontal.buttons</code>                                   |
| wait_animation_large.gif         | “Please wait” animation used in dialog boxes on page loads and in dashboard iframes when widgets are loading    |

**Table 12-7** shows images used in the earlier themes that have been removed in 5.0, as well as images that have been refactored by placing them in the file `inner_pagination_sprite.png`.

**Table 12-7 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 <code>wait_animation_large.gif</code>                  |
| 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    |  |
| 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 <code>inner_pagination_sprite.png</code> |
| first.gif                      | “Go to first” icon used for pagination controls inside report dashboard widgets                   | Image placed in sprite file <code>inner_pagination_sprite.png</code> |
| frame_background_sprite.png    | Gradient background image for the <code>#frame</code> element                                     |  |

Table 12-7 Images from Previous Default Theme Removed in 4.7

| Image                           | Usage   | Notes  |
|---------------------------------|---|--|
| frame_bkgd_corners.png          | Background image used for rounded corners with drop shadow on the <code>#frame</code> element             |  |
| frame_bkgd_edges_rl.png         | Drop shadow image used for right and left edges of the <code>#frame</code> element                        |  |
| frame_bkgd_edges_tb.png         | Drop shadow image used for top and bottom edges of the <code>#frame</code> element                        |  |
| frame_header_sprite.png         | Gradient background image with rounded corners used for the header of the <code>#frame</code> element.    | <code>#frame</code> 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 <code>inner_pagination_sprite.png</code> .  |
| last.gif                        | “Go to last” icon used for pagination controls inside report dashboard widgets                            | Image placed in sprite file <code>inner_pagination_sprite.png</code> . |
| 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 <code>.menu.context</code> elements         |  |
| menu_context_bkgd_edges_rl.png  | Background image used for right and left borders with drop shadow on <code>.menu.context</code> elements  |  |
| menu_context_bkgd_edges_tb.png  | Background image used for top and bottom borders with drop shadow on <code>.menu.context</code> elements  |  |
| menu_dropDown_bkgd_corners.png  | Background image used for rounded corners with drop shadow on <code>.menu.dropDown</code> elements        |  |
| menu_dropDown_bkgd_edges_rl.png | Background image used for right and left borders with drop shadow on <code>.menu.dropDown</code> elements |  |
| menu_dropDown_bkgd_edges_tb.png | Background image used for top and bottom borders with drop shadow on <code>.menu.dropDown</code> elements |  |
| menu_vertical_bkgd.png          | Background for <code>.menu.vertical</code>  |  |
| next-d.gif                      | Disabled state of <b>next</b> icon used for pagination controls inside report dashboard widgets           | Image placed in sprite file <code>inner_pagination_sprite.png</code> . |
| next.gif                        | <b>Next</b> icon used for pagination controls inside report dashboard widgets                             | Image placed in sprite file <code>inner_pagination_sprite.png</code> . |
| panel_bkgd_corners.png          | Background image of rounded corners with drop shadow used for the <code>.dialog.overlay</code> element    |  |

**Table 12-7 Images from Previous Default Theme Removed in 4.7**

| Image                                | Usage   | Notes |
|--------------------------------------|---|-------|
| panel_bkgd_edges_rl.png              | Background image of right and left borders with drop shadow used for the <code>.dialog.overlay</code> element   |       |
| panel_bkgd_edges_tb.png              | Background image of top and bottom borders with drop shadow used for the <code>.dialog.overlay</code> 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 <code>.info</code> element                    |       |
| panel_info_bkgd_edges_rl.png         | Background image of right and left borders with drop shadow used for the <code>.info</code> element             |       |
| panel_info_bkgd_edges_tb.png         | Background image of top and bottom borders with drop shadow used for the <code>.info</code> element             |       |
| panel_info_system_bkgd_corners.png   | Background image of rounded corners used for the <code>.info.system</code> element                              |       |
| panel_info_system_bkgd_edges_rl.png  | Background image of right and left borders used for the <code>.info.system</code> element                       |       |
| panel_info_system_bkgd_edges_tb.png  | Background image of top and bottom borders used for the <code>.info.system</code> element                       |       |
| panel_info_tooltip_bkgd_corners.png  | Background image of rounded corners used for the <code>.panel.tooltip.info</code> element                       |       |
| panel_info_tooltip_bkgd_edges_rl.png | Background image of right and left borders used for the <code>.panel.tooltip.info</code> element                |       |
| panel_info_tooltip_bkgd_edges_tb.png | Background image of top and bottom borders used for the <code>.panel.tooltip.info</code> element                |       |
| panel_inlay_bkgd_corners.png         | Background image of rounded corners with drop shadow used for the <code>.dialog.inlay</code> element            |       |
| panel_inlay_bkgd_edges_rl.png        | Background image of left and right borders with drop shadow used for the <code>.dialog.inlay</code> element     |       |
| panel_inlay_bkgd_edges_tb.png        | Background image of top and bottom borders with drop shadow used for the <code>.dialog.inlay</code> element     |       |
| panel_inlay_gradient_left.png        | Left background image with gradient and rounded corners used for the <code>.dialog.inlay .footer</code> element |       |



**Table 12-7 Images from Previous Default Theme Removed in 4.7**

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

## 12.4 Data Snapshots

As of 5.0, 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  $(100,000 \div 20) \times 2$  KB, or 10 MB.

Data snapshots are turned off globally by default in 5.0 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.

### 12.4.1 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.

## 12.5 Understanding the Changes in 5.0

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 [12.6, “Upgrading Preserving Custom Settings,” on page 122](#) 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 [12.7, “XML/A Connections to Servers with Multiple Organizations,” on page 122](#) for more information about this procedure.

## 12.6 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.

## 12.7 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*.

## CHAPTER 13 UPGRADING JASPERSERVER 3.5 OR EARLIER

---

### 13.1 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 5.5.

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 5.5.

### 13.2 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 (i.e. the originally installed version number) of JasperReports Server.

## CHAPTER 14 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 5.5 CP to 5.5 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 5.5 WAR File Distribution**
- **Configuring Buildomatic for Your Database and Application Server**
- **Upgrading to the Commercial Version of JasperReports Server 5.5**
- **Starting and Logging into JasperReports Server 5.5**
- **Re-Configuring XML/A Connections (Optional)**

### 14.1 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.

## 14.2 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.

### 14.2.1 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.

### 14.2.2 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 section [A.8.3, “Maximum Packet Size in MySQL,” on page 151](#).

## 14.3 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 section [14.5.1, “Example Buildomatic Configuration,” on page 127](#). 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.

## 14.4 Preparing the JasperReports Server 5.5 WAR File Distribution

Use the buildomatic scripts included in the commercial 5.5 WAR file distribution release package to carry out the upgrade. Follow these steps to obtain and unpack the commercial 5.5 WAR file distribution ZIP file:

1. The WAR file distribution comes in a file named `jasperreports-server-5.5-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-5.5-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>
```

## 14.5 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.

### 14.5.1 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`.

#### 14.5.1.1 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
```

#### 14.5.1.2 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
```

## 14.6 Upgrading to the Commercial Version of JasperReports Server 5.5

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<br>js-ant create-js-db<br>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.                                                                                                                                                                                                                               |
| js-ant import-upgrade<br>-DimportFile=<path>/js-cp-export.zip     | The -DimportFile argument should point to the js-cp-export.zip file you created earlier.<br>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 5.5 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 section [A.8.3, “Maximum Packet Size in MySQL,”](#) on page 151.

## 14.7 Starting and Logging into JasperReports Server 5.5

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 section [6.2, “Setting Up the JasperReports Server License,”](#) on page 45.

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 [11.9.2, “Clearing the Application Server Work Folder,” on page 107](#), [11.9.3, “Clearing the Application Server Temp Folder,” on page 107](#), and [11.9.4, “Clearing the Repository Cache Database Table,” on page 108](#).

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

### 14.7.1 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.

### 14.7.2 Logging into the Commercial Version of JasperReports Server 5.5

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 [Appendix A, “Troubleshooting,” on page 145](#).

## 14.8 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`



## CHAPTER 15 PASSWORD ENCRYPTION IN JASPERREPORTS SERVER

---

By default, password encryption is enabled in JasperReports Server and passwords are stored as cipher text in the database. System administrators can change the encryption algorithm and specify the salt key used to initiate the encryption algorithm.

This chapter describes the procedure to enable password encryption if you have a JasperReports Server instance without encryption turned on. For more information about encryption options, see the *JasperReports Server Administrator Guide*.

This chapter contains the following sections:

- **Backing Up Your JasperReports Server Database**
- **Stopping Your Application Server**
- **Running the Repository Export Utility**
- **Specifying Encryption Settings in the JasperReports Server WAR**
- **Specifying Encryption Settings for the Import Utility**
- **Recreating the JasperReports Server Database**
- **Importing Your Repository Data**
- **Starting the Application Server**
- **Logging into JasperReports Server**

### 15.1 Backing Up Your JasperReports Server Database

As a precaution, you must back up your `jasperserver` database in the event of a problem while enabling encryption.

To back up the default PostgreSQL database, go to the `<js-install>` directory and run the following command:

```
pg_dump -U postgres jasperserver > js-backup.sql
```

To back up DB2, Oracle, Microsoft SQL Server, and MySQL databases, refer to your database product documentation.

### 15.2 Stopping Your Application Server

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

### 15.3 Running the Repository Export Utility

The repository export utility writes the JasperReports Server repository objects to a set of XML and binary files. The output of the export operation is known as an export catalog.

To create the export catalog, go to the <js-install>/buildomatic directory and run the following commands. Note that there are two dashes (--) in front of the command options:

```
Windows: js-export.bat --everything --output-dir js-backup-catalog
Linux:   js-export.sh --everything --output-dir js-backup-catalog
```

For information about running the export utility, refer to [Chapter 16, “Configuring the Import-Export Utilities,” on page 137](#).

You may also run the export operation from the JasperReports Server UI. Refer to [“Running Import or Export from the UI” on page 141](#) for information on running export from the UI.

### 15.4 Specifying Encryption Settings in the JasperReports Server WAR

JasperReports Server uses the Spring configuration and security to enable and configure encryption. Use these options to set up strong encryption. This section is focused on the minimal configuration for enabling encryption.

1. Open the following file for editing:  
`<tomcat/webapps/jasperserver-pro/WEB-INF/ApplicationContext-security.xml`
2. In the definition of the `daoAuthenticationProvider` bean, there is a commented-out reference to the `passwordEncoder` bean. Look for the section of the XML file that starts with:  
`<bean id="daoAuthenticationProvider"`
3. Enable encryption in the `passwordEncoder` bean by modifying the `allowEncoding` property. Change the value from `false` to `true` so that it looks like this:  
`<property name="allowEncoding"><value>true</value></property>`
4. If the default DESede algorithm is used, the `secretKey` represents the salt key and must be 24 characters. By default, the `keyInPlainText` property is `true`, meaning the key can be in plain text to make it easier to enter, for example:  
`<property name="keyInPlainText"><value>true</value></property>`  
`<property name="secretKey"><value>jaspersoftInSanFrancisco</value></property>`



The text `jaspersoftInSanFrancisco` is 24 characters long, therefore the two properties above work with their default values. However, for better security, we recommend that they be changed.

5. The last two properties may be left unchanged. They are set to DESede by default. The default values are the following:  
`<property name="secretKeyAlgorithm"><value>DESede</value></property>`  
`<property name="cipherTransformation"><value>DESede/CBC/PKCS5Padding</value></property>`



The `secretKey`, `secretKeyAlgorithm`, and `cipherTransformation` property settings must be consistent with each other. For example, different algorithms expect different key lengths.

6. Save and close the file. Encryption is now enabled for the JasperReports Server application upon the next restart.

#### 15.4.1 Specifying Encryption Settings - Reference Table

The information in the [Table 15-1](#) is a summary of the available password encryption configuration options described in the previous section.

**Table 15-1 Password Encryption Configuration Options**

| Configuration File                          |
|---------------------------------------------|
| ...\WEB-INF\applicationContext-security.xml |

**Table 15-1 Password Encryption Configuration Options, continued**

| Property             | Bean                      | Description                                                                                                                                                                                                                                                                                                                                 |
|----------------------|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| passwordEncoder      | daoAuthenticationProvider | Comment this property out to disable the encryption of passwords.                                                                                                                                                                                                                                                                           |
| secretKey            | passwordEncoder           | The salt key to use as a pass phrase when encrypting passwords. This string is used to encrypt passwords. This value can be a simple string or a numeric representation that can be parsed by Integer.decode(). For example:<br>String:<br>This is my secret key<br>Numeric representation:<br>0xC8,0x43,0x29,0x49,0xAE,0x25,0x2F,0xA1,0xC1 |
| keyInPlainText       | passwordEncoder           | Determines whether the secret key is a simple string or a numeric representation. Set this parameter to TRUE if the secretKey is a string; set it to FALSE if the key is a numeric representation.                                                                                                                                          |
| secretKeyAlgorithm   | passwordEncoder           | The name of the algorithm to use, such as DESede.                                                                                                                                                                                                                                                                                           |
| cipherTransformation | passwordEncoder           | The name of the transformation, such as DES/CBC/PKCS5Padding.                                                                                                                                                                                                                                                                               |

The secretKey, secretKeyAlgorithm, and cipherTransformation must be consistent with each other. For example, if the secretKeyAlgorithm is DESede, the secretKey must be 24 bytes long. For more information about secretKey, secretKeyAlgorithm, and cipherTransformation, see Sun's `javax.crypto` documentation.

## 15.5 Specifying Encryption Settings for the Import Utility

To continue with the process of changing your encryption settings, you should add the same encryption settings to the import-export functionality as you did for the JasperReports Server WAR in the sections above.

1. Open the following configuration file for editing:  
`<js-install>/buildomatic/conf_source/iePro/applicationContext-security.xml`
2. This file contains the `passwordEncoder` bean definition, the same as in the JasperReports Server WAR, only by itself. Modify this file to specify the encryption settings as described in section 15.4, “[Specifying Encryption Settings in the JasperReports Server WAR](#),” on page 132.

## 15.6 Recreating the JasperReports Server Database

Next, drop your existing jasperserver database and recreate an empty jasperserver database.

### 15.6.1 Dropping and Recreating the Database in PostgreSQL

1. Change directory to `<js-install>/buildomatic/install_resources/sql/postgresql`.
2. Start `psql` using an administrator account such as `postgres`:  
`psql -U postgres`

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

```
drop database jasperserver;  
create database jasperserver encoding='utf8';  
\c jasperserver  
\i js-pro-create.ddl  
\i quartz.ddl
```

### 15.6.2 Dropping and Recreating the Database in MySQL

1. Change directory to <js-install>/buildomatic/install\_resources/sql/mysql.
2. Log into your MySQL client:  
`mysql -u root -p`
3. Drop the jasperserver database, create a new one, and load the jasperserver schema:

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

### 15.6.3 Dropping and Recreating the Database in Oracle

1. Change directory to <js-install>/buildomatic/install\_resources/sql/oracle.
2. Log into your SQLPlus client, for example:  
`sqlplus sys/sys as sysdba`
3. Drop the jasperserver database, create a new one, and load the jasperserver schema:

```
SQL> drop user jasperserver cascade;  
SQL> create user jasperserver identified by password;  
SQL> connect jasperserver/password  
SQL> @js-pro-create.ddl  
SQL> @quartz.ddl
```

### 15.6.4 Dropping and Recreating in the Database in Microsoft SQL Server

1. Change directory to <js-install>/buildomatic/install\_resources/sql/sqlserver.

2. Drop the jasperserver database, create a new one, and load the jasperserver schema using the SQLCMD utility:

```
cd <js-install>\buildomatic\install_resources\sql\sqlserver

sqlcmd -S ServerName -Usa -Psa
1> DROP DATABASE [jasperserver]
2> GO

1> CREATE DATABASE [jasperserver]
2> GO

1> USE [jasperserver]
2> GO

1> :r js-pro-create.ddl
2> GO

1> :r quartz.ddl
2> GO
```

## 15.7 Importing Your Repository Data

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

To import your backup catalog to the repository:

1. Change directory to <js-install>/buildomatic.
2. Run the import utility with the command for your platform. Note that there are two dashes (--) in front of the command options:

Windows: `js-import.bat --input-dir js-backup-catalog`

Linux: `js-import.sh --input-dir js-backup-catalog`

For information about running the import utility, see [Chapter 16, “Configuring the Import-Export Utilities,” on page 137](#).

## 15.8 Starting the Application Server

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

## 15.9 Logging into JasperReports Server

You can now log into JasperReports Server.

Enter your user ID and password in the same manner as you did before encryption was turned on. You can check the contents of the `JUser` table in the `jasperserver` database and examine the password column to see that the password is no longer stored in plain text.





## CHAPTER 16 CONFIGURING THE IMPORT-EXPORT UTILITIES

The import and export utilities add resources to or extract resources from the JasperReports Server repository. Typically, users export data from their previous instance and import it into their new installation when upgrading JasperReports Server. The import utility is also used at installation time to load the sample data into the repository.

This chapter covers the basic importing and exporting commands. For more information about command options, see the *JasperReports Server Administrator Guide*.

This chapter contains the following sections:

- [Introduction](#)
- [Import-Export Configuration Files](#)
- [Changing Your Configuration Settings](#)
- [Adding an Additional JDBC Driver](#)
- [Running Import or Export](#)

### 16.1 Introduction

As of release 5.0, the import-export functionality has been added the UI of JasperReports Server. This makes importing and exporting resources much more convenient for administrators. Using the UI to handle import and export is the recommended method. (Note: you must login as user `superuser` to use this UI.)

If you plan to use the command line import-export functionality, this chapter covers any additional pointers and/or steps that might be required to configure import-export. If you plan to use the JasperReports Server UI to handle your import-export tasks then this chapter does not apply to you.

You can import and export resources from the command line by running either of these types of scripts:

| Type         | Example Command                                                         | Description                            |
|--------------|-------------------------------------------------------------------------|----------------------------------------|
| Buildomatic  | <code>js-ant export-everything -DexportFile=js-catalog-exp.zip</code>   | Auto-configured buildomatic Ant script |
| Shell Script | <code>js-export.sh --everything --output-file=js-catalog-exp.zip</code> | Import-export shell scripts            |

Both types of scripts are located in:

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

You use the same commands, regardless of operating system, to run the buildomatic Ant scripts. The shell scripts, on the other hand, have different names:

| Windows Commands | Linux and Mac OSX Commands |
|------------------|----------------------------|
| js-import.bat    | js-import.sh               |
| js-export.bat    | js-export.sh               |

If you are running the `js-import.sh/bat` or `js-export.sh/bat` shell scripts, you might have to manually configure some databases as described in the following section.

## 16.2 Import-Export Configuration Files

The import-export utility files are in the buildomatic folder. Use or modify these files to make configuration changes.

| File or Location                                                    | Purpose                                                                                                   |
|---------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------|
| <js-install>/buildomatic/js-import.bat and .sh                      | Import scripts for Windows and Linux, respectively                                                        |
| <js-install>/buildomatic/js-export.bat and .sh                      | Export scripts for Windows and Linux, respectively                                                        |
| <js-install>/buildomatic/default_master.properties                  | File that you must edit (already in place if you installed from the binary installer)                     |
| <js-install>/buildomatic/build_conf/default/js.jdbc.properties      | Database and hibernate dialect settings file (put in place after you run <code>js-ant gen-config</code> ) |
| <js-install>/buildomatic/conf_source/iePro/log4j.properties         | log4j.properties file controls output logging levels                                                      |
| <js-install>/buildomatic/conf_source/iePro/applicationContext-*.xml | Spring configuration files                                                                                |
| <js-install>/buildomatic/conf_source/iePro/lib                      | All of the JasperReports Server jar files and the JDBC driver location                                    |

## 16.3 Changing Your Configuration Settings

If you are running the `js-import.bat/.sh` or `js-export.bat/.sh` shell scripts, then this section applies.

When you install JasperReports Server from the installer binary, the import and export shell scripts are automatically configured. However, if you are installing the WAR file distribution you must modify the `default_master.properties` configuration file to include your database settings and perform other configuration tasks.

### 16.3.1 Creating a default\_master.properties File

If you do not already have a `<js-install>/buildomatic/default_master.properties` file in place, create one. For instance, copy and rename `buildomatic/sample_conf/postgresql_master.properties` to `buildomatic/default_master.properties`.

Perform the following steps:

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

Perform **step 2** to **step 4** of the procedure “**To install the WAR file using js-install scripts:**” on **page 34** with the following exception for Oracle users.



Exception: To use the js-import and js-export scripts, Oracle users can set the sysUsername and sysPassword to the same name as dbUsername and dbPassword in the `default_master.properties`. The system user name and password are not required because js-import and js-export do not make changes to database schema.

2. Run: `js-ant clean-config gen-config`

### 16.3.2 Location of Properties Files and JDBC Driver

Paths and file names for properties files are:

- `<js-install>/buildomatic/build_conf/default/js.jdbc.properties`
- `<js-install>/buildomatic/build_conf/default/js.quartz.properties` (only for DB2 and PostgreSQL)

The JDBC driver file must be in this directory:

- `<js-install>buildomatic/conf_source/iePro/lib`

As of release 5.5, JDBC drivers for all five certified databases are found in the `lib` folder by default (DB2, MySQL, Oracle, PostgreSQL, and SQL Server).

### 16.3.3 Checking the js.jdbc.properties File

**Table 16-1** and **Table 16-2** contain sample settings for each database.

The `js.jdbc.properties` file is automatically configured by buildomatic. But you can check the file to make sure the settings are correct for your environment.

**Table 16-1 JDBC Settings in the js.jdbc.properties File**

| Database   | Sample Property Values                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PostgreSQL | <pre> metadata.hibernate.dialect= com.jaspersoft.hibernate.dialect.PostgresqlNoBlobDialect metadata.jdbc.driverClassName=org.postgresql.Driver metadata.jdbc.url=jdbc:postgresql://localhost:5432/jasperserver metadata.jdbc.username=<b>postgres</b> metadata.jdbc.password=<b>postgres</b> or metadata.jdbc.encryptedPassword=<b>encrypted-postgres</b> </pre>                                                                                                          |
| MySQL      | <pre> metadata.hibernate.dialect=org.hibernate.dialect.MySQLInnoDBDialect (MySQL 5.1) metadata.hibernate.dialect=org.hibernate.dialect.MySQL5InnoDBDialect (MySQL 5.5) metadata.jdbc.driverClassName=com.mysql.jdbc.Driver metadata.jdbc.url=jdbc:mysql://localhost:3306/ jasperserver?useUnicode=true&amp;characterEncoding=UTF-8 metadata.jdbc.username=root metadata.jdbc.password=<b>password</b> or metadata.jdbc.encryptedPassword=<b>encrypted-password</b> </pre> |

**Table 16-1 JDBC Settings in the js.jdbc.properties File, continued**

| Database   | Sample Property Values                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Oracle     | <pre> metadata.hibernate.dialect=com.jaspersoft.ji.hibernate.dialect.OracleJICustomDialect metadata.jdbc.driverClassName=oracle.jdbc.OracleDriver metadata.jdbc.url=jdbc:oracle:thin:@localhost:1521:orcl metadata.jdbc.username=jasperadmin metadata.jdbc.password=password or metadata.jdbc.encryptedPassword=encrypted-password </pre>                                                                                                                                                                                                                                                     |
| DB2        | <pre> metadata.hibernate.default_schema=JSPRSRVR metadata.hibernate.dialect=com.jaspersoft.ji.hibernate.dialect.DB2JICustomDialect metadata.jdbc.driverClassName=com.ibm.db2.jcc.DB2Driver metadata.jdbc.url=jdbc:db2://localhost:50000/jsprsrvr:driverType=4; fullyMaterializeLobData=true;fullyMaterializeInputStreams=true; progressiveStreaming=2;progressiveLocators=2;currentSchema=JSPRSRVR; metadata.jdbc.username=db2admin metadata.jdbc.password=password or metadata.jdbc.encryptedPassword=encrypted-password metadata.jndi=jdbc/jasperserver metadata.upperCaseNames=true </pre> |
| SQL Server | <pre> metadata.hibernate.dialect=org.hibernate.dialect.SQLServerJICustomDialect metadata.jdbc.driverClassName=com.microsoft.sqlserver.jdbc.SQLServerDriver metadata.jdbc.url=jdbc:sqlserver://localhost:1433;databaseName=jasperserver;SelectMethod=cursor metadata.jdbc.username=sa metadata.jdbc.password=sa or metadata.jdbc.encryptedPassword=encrypted-sa </pre>                                                                                                                                                                                                                         |

### 16.3.4 Checking the js.quartz.properties File

The `js.quartz.properties` file is automatically configured by `buildomatic`. But you can check the file to make sure the settings are correct for your environment.

This file is located at:

```
<js-install>/buildomatic/build_conf/default/js.quartz.properties
```

**Table 16-2 Quartz Settings in the js.quartz.properties File**

| Database   | Sample Property Values                                                                                         |
|------------|----------------------------------------------------------------------------------------------------------------|
| PostgreSQL | <pre> quartz.delegateClass=org.quartz.impl.jdbcjobstore.PostgreSQLDelegate quartz.tablePrefix=QRTZ_ </pre>     |
| DB2        | <pre> quartz.delegateClass=org.quartz.impl.jdbcjobstore.DB2v8Delegate quartz.tablePrefix=JSPRSRVR.QRTZ_ </pre> |
| Other DBs  | <pre> quartz.delegateClass=org.quartz.impl.jdbcjobstore.StdJDBCDelegate quartz.tablePrefix=QRTZ_ </pre>        |

## 16.4 Adding an Additional JDBC Driver

JDBC drivers are already pre-configured for all certified databases. If you would like to use a different JDBC driver, you can copy it to the following folder:

```
<js-install>/buildomatic/conf_source/iePro/lib
```

All Jaspersoft distributed JDBC drivers can be found at this location:

```
<js-install>/buildomatic/conf_source/db/<db-type>/jdbc
```

## 16.5 Running Import or Export

### 16.5.1 Running Import or Export from the UI

For complete information running import and export in the JasperReports Server UI, please refer to the *JasperReports Server Administrator Guide*.

#### 16.5.1.1 Short Summary of Import-Export Functionality in the JasperReports Server UI

As of Release 5.0, import-export functionality is supported in the UI. You must login as the administrative user `superuser` in order to access this functionality.

##### Export:

- **Export Single Set of Resources:** Navigate to `View > Repository`. From this page, which shows a tree view of the JasperReports Server repository, you can right click a folder (on the left side of the page), or a resource (on the right side of the page) to get a context menu that has an `export` option. You can click `export` and you will be prompted for a file name and location to save the export zip file.
- **Export All Resources:** Navigate to `Manage > Server Settings > Export` tab. This page will export all resources from the repository (similar to `js-export --everything`). You can click the `Export` button and you will be prompted for a file name and location to save the export zip file.
  - **Note:** If you export all resources from a repository with a large number of resources, the execution of this command in the UI may take a long while. It is recommended to export from the command line in the case of a large number of resources.



Warning: Export from the UI in Release 5.1 has a bug. To work around the bug, you can export from the command line, or edit the `export.zip` file created by the UI export. See below [16.5.1.2, “Workaround for Export bug in the 5.1 Release,”](#) for details

##### Import:

Import a zip file: Navigate to `Manage > Server Settings > Import` tab. Click the `Import Data File Browse` button to be prompted to choose a file from the file system. Then click the `Import` button to import resources.

#### 16.5.1.2 Workaround for Export bug in the 5.1 Release

In release 5.1 there is a known bug when exporting resources from the UI. There is incorrect metadata written to a file found inside the export zip file. To fix/workaround this problem you can export from the command line or update the incorrect metadata that is in the export zip file.

To update the incorrect metadata, you should create a temporary folder and unpack the `export.zip` file to this folder. In the root of the folder you will find a file named `index.xml`. You can edit the `index.xml` file:

Change: `name="jsVersion" value="5.0.0 CE"`

To: `name="jsVersion" value="5.1.0 PRO"`

Then rezip the `export.zip` file.

## 16.5.2 Running Import from Buildomatic

The import target for ant has the following syntax:

```
Windows:  js-ant import -DimportFile=<filename> [-DimportArgs="<import-options>"]
Linux and
Mac OSX:  ./js-ant import -DimportFile=<filename> [-DimportArgs="\<import-options>\"]
```

The imported file is handled as a ZIP archive if its name ends in `.zip`, otherwise it is handled as a directory. The `importArgs` argument is optional and can contain more than one import option. On Linux, all double quotation marks (") must be escaped with a backslash (\).



When performing a large import using `js-ant`, the server should be stopped (or put into a mode with reduced load) to avoid issues with caches, configuration, and security.

The following examples are typical import commands on Windows:

```
js-ant import-help-pro
js-ant import -DimportFile=my-reports.zip
js-ant import -DimportFile=my-datasources -DimportArgs="--update"
```

The following examples are typical import commands on Linux:

```
./js-ant import-help-pro
./js-ant import -DimportFile=my-reports.zip
./js-ant import -DimportFile=my-datasources.zip -DimportArgs="\--update\"
```

## 16.5.3 Running Export from Buildomatic

The export target for ant has the following syntax:

```
Windows:  js-ant export -DexportFile=<filename> -DexportArgs="<export-options>"
Linux and
Mac OSX:  ./js-ant export -DexportFile=<filename> -DexportArgs="\<export-options>\"
```

The export file format is a ZIP file or a set of files under a new directory name. If you specify the `.zip` extension for your output filename, a ZIP archive is created automatically. Otherwise, a directory with files and sub-directories is created as a uncompressed set of files.

The `exportArgs` argument requires double quotation marks (") and can contain more than one export option, as shown in these Windows examples:

```
js-ant export-help-pro
js-ant export -DexportFile=my-domains.zip
    -DexportArgs="--uris /organizations/organization_1/Domains"
js-ant export -DexportFile=my-reports-and-users.zip
    -DexportArgs="--uris /organizations/organization_1/reports
    --users jasperadmin|organization_1,joeuser|organization_1"

js-ant export -DexportFile=my-datasources
    -DexportArgs="--uris /organizations/organization_1/datasources
    --roles ROLE_USER"
js-ant export -DexportFile=js-everything.zip -DexportFile=js="--everything"
```

On Linux, all double quotation marks (“”) and other characters, such as the vertical bar (|), which separates login user and organization names must be escaped with a backslash (\). In addition, when listing user names, enclose the list in single quotation marks ('), as shown in this Linux example:

```
./js-ant export-help-pro
./js-ant export -DexportFile=my-reports-and-users.zip
-DexportArgs="--uris /organizations/organization_1/reports
--users 'jasperadmin\|organization_1,joeuser\|organization_1'\"
```

### 16.5.4 Running the Import-Export Shell Scripts

The import-export shell scripts are distinct from the buildomatic Ant import-export functionality. If you installed JasperReports Server using the WAR file distribution, you may need to configure the import-export shell scripts, as described in [Chapter 16, “Configuring the Import-Export Utilities,” on page 137](#).

If you installed JasperReports Server using the binary installer, the scripts are already properly configured.

These import-export shell scripts are available for Windows, Linux, and Mac:

- ♦ Windows:
  - <js-install>/buildomatic/js-export.bat
  - <js-install>/buildomatic/js-import.bat
- ♦ Linux and Mac:
  - <js-install>/buildomatic/js-export.sh
  - <js-install>/buildomatic/js-import.sh



When performing a large import, the server should be stopped (or put into a reduced load mode) to avoid issues with caches, configuration, and security.

In addition, Jaspersoft recommends you stop the server instance before running a large export operation.

Run the import-export shell scripts with the help option to list the other options that you can use. For example:

```
js-export.bat --help
js-import.sh --help
```





## APPENDIX A TROUBLESHOOTING

---

This appendix contains the following sections:

- **Binary Installer Freezes**
- **Error Running Buildomatic Scripts**
- **Unable to Edit Files on Windows 7**
- **Bash Shell for Solaris, IBM AIX, HP UX and FreeBSD**
- **Linux Installer Issue with Unknown Host Error**
- **Installation Error with Windows Path**
- **Mac OSX Issues**
- **Database-related Problems**
- **Application Server-related Problems**
- **License-related Errors**
- **Problems Importing and Exporting Data from the Repository**
- **Permissions Error in Overlay Upgrade with PostgreSQL and Bundled Jaspersoft Installation**

### A.1 Binary Installer Freezes

If you run the JasperReports Server installer on any platform and the installation fails, the following resources can help you find the source of the error.

#### A.1.1 Installer Log Files

If you run the JasperReports Server installer on any platform and there is an error, it is helpful to look at the log file created by the installer. This log file records the status and completion of installer operations. If your installer has had an explicit error, there may be a specific error message in the log. At a minimum, the log file should help narrow where the error has occurred even if there is not a specific error message.

You can find the installer log in the following locations:

Windows: <js-install>/installation.log  
Linux: <js-install>/installation.log  
Mac <js-install>/installation.log

If you have tried multiple installs, make sure you view the most recent install log file. Then, you can submit the installation.log to Jaspersoft Technical Support.

## A.1.2 Installer DebugTrace Mode

In addition, if you run the JasperReports Server installer on any platform and have a problem, you can run the installer a second time using the `--debugtrace` option. The `--debugtrace` option creates a binary output file that gives precise details about the execution of the installer and any problems encountered; this file can be analyzed by Jaspersoft Technical Support.

To use the `--debugtrace` option, you must run the installer from the command line and specify an output filename. The precise command depends on your platform (Linux, Windows, or Mac OSX). For example, you can execute the installer with a command similar to the following:

```
jasperreports-server-<ver>-linux-x64-installer.run --debugtrace install-trace-out.bin
```

When you run the installer in `--debugtrace` mode, the installer will take extra time to write the binary output file. The final size of the output file is approximately 10 mg. Contact Jaspersoft Technical Support to hand off of the binary file for analysis.

## A.2 Error Running Buildomatic Scripts

The buildomatic scripts depend on both Java and Apache Ant. There are two common configuration errors when attempting to do an installation using these scripts (if you are not using the included, bundled Apache Ant).

### A.2.1 Missing Java JDK

If you have the Java JRE (Java Runtime Environment) instead of the JDK, you will not have the additional utilities that are required. In particular, an error referring to the `tools.jar` might occur, as in the following message:

```
[exec] [ERROR] BUILD FAILURE
[exec] [INFO] -----
[exec] [INFO] Compilation failure
[exec] Unable to locate the Javac Compiler in:
[exec]   c:\Program Files\Java\jdk1.6.0_10\jre\..\lib\tools.jar
[exec] Please ensure you are using JDK 1.6 or above and
[exec] not a JRE (the com.sun.tools.javac.Main class is required).
[exec] In most cases you can change the location of your Java
[exec] installation by setting the JAVA_HOME environment variable.
```

The solution is to download and install the Sun Java JDK, labeled as the Java SE Development Kit on the Sun web site.

If you are upgrading JasperReports Server, you can also use the Java 1.6 JDK bundled in the previous version, as described in section [11.9.1, “Handling JasperReports Server Customizations,” on page 107](#).

### A.2.2 Forgot to Copy the File `ant-contrib.jar`

If you are using your own version of Ant and your Ant instance does not have the `ant-contrib.jar` in the `lib` directory, you will get an error similar to the following:

```
BUILD FAILED
c:\js-builds\jasperserver\buildomatic\install.xml:6:
```

Ant failed to create a task or type. To correct the error, copy `<js-install>/buildomatic/extra-jars/ant-contrib.jar` to your `<apache-ant>/lib` directory.

### A.2.3 Failure with '\$' Character in Passwords in Buildomatic Scripts

Ant is unable to accept more than one consecutive '\$' character in passwords in buildomatic scripts.

This issue only occurs when two dollar signs are used in a row. For example, “\$pas\$word\$” or “pas\$word\$” will not fail. If you have two consecutive dollar signs, escape each with 4 dollar signs. For example, if you use “pa\$\$word” you would need to set it as “pa\$\$\$\$\$\$\$\$word” in the configuration file. Once you do this, JRS will have all data connections set to “pa\$\$word”.

## A.2.4 Older Apache Ant Version

As of the release of JasperReports Server 4.0, Apache Ant version 1.8.1 or higher is required. There are improvements to error handling routines in the buildomatic js-install scripts which required the higher level of Ant. If you are using an older version of Ant, you will get an error similar to the following:

```
BUILD FAILED
c:\js-builds\jasperserver\buildomatic\install.xml:37:
Problem: failed to create task or type componentdef
```

To check your version of Ant and verify that it is at a high enough level, enter:

```
ant -version
```

If you have a lower version of Ant, check to see if it is set in your class path by entering:

```
echo %CLASSPATH%
```

To use the JasperReports Server version of Ant, update your CLASSPATH variable to point at the <js-install>/apache-ant/bin directory.

## A.3 Unable to Edit Files on Windows 7

In some cases, you may want to manually edit files under your C:/Jaspersoft directory during or after installation. For security reasons, Windows 7 doesn't allow normal processes to change files in many folders including the Program Files folder, for instance. When you attempt to edit these files, you may see an error such as the following:

You don't have permission to save in this location. Contact the administrator to obtain permission.

You can edit these files by running as administrator. For example, to edit these files with Notepad on Windows 7:

Click **Start > All Programs > Accessories**, right-click **Notepad**, and click **Run as administrator**.

## A.4 Bash Shell for Solaris, IBM AIX, HP UX and FreeBSD

To execute the js-install shell scripts described in Chapter 5 of this guide, the bash shell is required. The js-install and js-upgrade scripts which are found in the buildomatic folder are the following:

```
js-install.sh
js-upgrade-newdb.sh
js-upgrade-samedb.sh
```

The bash shell is not included by default in all Unix platforms. In the case where the bash shell is not available, it is necessary to download and install the bash shell. Bash shells are available for the following platforms:

```
Solaris
IBM AIX
HP UX
FreeBSD
```

Alternatively, you can manually run the same "buildomatic" Ant targets that are run by the js-install script. These Ant targets are listed in ["Troubleshooting Your JasperReports Server Configuration" on page 33](#).

Also, check that you have updated your local Ant to include the ant-contrib.jar which supports conditional logic in Ant. The ant-contrib.jar is found in the location below and it should be copied to your <ant\_home>/lib folder:

```
buildomatic/extra-jars/ant-contrib.jar
```

For updating your local Ant instance with the ant-contrib.jar see Section A.2.2 above ("Forgot to Copy the File ant-contrib.jar").

If you try and use the bundled ant that is included with the JasperServer WAR file Distribution ZIP package, you may get the same non-bash syntax error. You may get the error below, for example:

```
js-ant help-install
ANT_HOME=../apache-ant: is not an identifier
```

If you have the bash shell installed, you can try executing the js-ant command by calling bash explicitly, for example:

```
bash js-ant help-install
```

### A.5 Linux Installer Issue with Unknown Host Error

If a Linux server does not have proper hostname entries in the `/etc/hosts` file, it is possible to get installer errors.

The installer carries out an import operation in order to load the core, minimal data into the repository database. This import operation can fail if the host is not configured.

If the import operation fails during installation, the installation will also fail. However, there should be an `installation.log` in the root of the installation folder to help debug the problem. The `installation.log` is located here:

```
<js-install>/installation.log
```

An improperly configured hosts file typically causes the log, or error messages displayed on the console, to contain error messages such as these:

```
Caused by: java.net.NoRouteToHostException: No route to host
com.mysql.jdbc.exceptions.jdbc4.CommunicationsException: Communications link failure
ERROR Cache:145 - Unable to set localhost. This prevents creation of a GUID
java.net.UnknownHostException
org.quartz.SchedulerException: Couldn't get host name!
```

#### To fix the `/etc/hosts` file:

1. Include entries that look similar to these:

```
127.0.0.1    localhost.localdomain    localhost
172.17.5.0   myhost.mydomain.com      myhost
```

For instance:

```
127.0.0.1    localhost.localdomain    localhost
172.17.5.0   myhost.jaspersoft.com    myhost
```

2. Also, you can double check the file `/etc/sysconfig/network` (if it exists).

In this file it would be similar to the following:

```
HOSTNAME=myhost
```

3. After fixing the `/etc/hosts` file, reinstall JasperReports Server.

## A.6 Installation Error with Windows Path

If the length to the path to the war archive is longer than the maximum allowed by Windows, you will get an error message similar to:

```
BUILD FAILED
c:\jaspers\war_file_installations\war_mysql_500\jasperreports-server-5.0-
bin\buildomatic\bin\db-common.xml:871:
The following error occurred while executing this line:
c:\jaspers\war_file_installations\war_mysql_500\jasperreports-server-5.0-
bin\buildomatic\bin\import-export.xml:264:
The following error occurred while executing this line:
c:\jaspers\war_file_installations\war_mysql_500\jasperreports-server-5.0-
bin\buildomatic\bin\import-export.xml:158:
java.io.IOException: Cannot run program "C:\Program
Files\Java\jdk1.6.0_38\jre\bin\java.exe": CreateProcess error=206, The filename
or extension is too long
    at java.lang.ProcessBuilder.start(ProcessBuilder.java:460)
    at java.lang.Runtime.exec(Runtime.java:593)
```

You will need to move the war archive to reduce the path length.

More information is available from Microsoft at: [http://msdn.microsoft.com/en-us/library/windows/desktop/aa365247\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa365247(v=vs.85).aspx).

## A.7 Mac OSX Issues

### A.7.1 Problem Starting JasperReports Server on Mac

Jaspersoft has seen some issues caused by the improper shutdown of the Tomcat included with the JasperReports Server. This could be related to the machine being shutdown while Tomcat is running.

When the Tomcat scripts start Tomcat, they write a pid (Process ID) file to the Tomcat folder. Tomcat uses this to determine whether the Tomcat instance is already running. When Tomcat is shutdown, this pid file is removed. However, if the pid file is not removed on shutdown, Tomcat will fail to start up.

You may see this when you double-click the jasperServerStart.app startup. It will seem like JasperReports Server is starting up but it never actually starts up.

In order to recover from this issue, you will need to manually delete the pid file.

#### Delete catalina.pid using Finder:

1. Navigate to the <js-install>/tomcat/temp folder  
For instance: /Applications/jasperreports-server-<ver>/tomcat/temp
2. Delete catalina.pid

#### Delete the catalina.pid file using Terminal shell:

1. Open a Terminal shell (Finder > Go > Utilities > Terminal Icon)
2. Navigate to the <js-install>/tomcat/temp folder  
For instance: /Applications/jasperreports-server-<ver>/tomcat/temp
3. Enter the following command:  
rm catalina.pid

**To start and stop the PostgreSQL and Tomcat components separately from the command line shell:**

1. Open a Terminal shell (Finder > Go > Utilities > Terminal Icon).
2. Navigate to the <js-install> folder.  
For instance: /Applications/jasperreports-server-<ver>
3. To Start:  

```
./ctlscript start postgresql  
./ctlscript start tomcat
```
4. To shutdown:  

```
./ctlscript stop  
or  
./ctlscript stop tomcat  
./ctlscript stop postgresql
```

## A.7.2 Installation Error on Mac OSX 10.8 (Lion)

When trying to install Jaspersoft on Mac OSX 10.8 (Lion) you might see the pop-up message “jasperreports-server-5.2-osx-x86-installer.app can't be opened because it is from an unidentified developer. Your security preferences allow installation of only apps from the Mac App Store and identified developers.”

If you see that message, perform the following procedure:

1. Right-click or Control-click the application and choose **Open**.
2. Click the **Open** button at the next dialog warning to launch the installer.

## A.8 Database-related Problems

### A.8.1 Database Connectivity Errors

The most common problems encountered with a new JasperReports Server instance are database configuration problems. If the connection fails, perhaps the application server cannot find the driver for the data source. For example, in a default installation of JasperReports Server, Tomcat looks for data source drivers in <js-install>/apache-tomcat/lib. If the driver is not there, put a copy of the driver in this directory and restart Tomcat.

This section contains information that may help resolve other connectivity issues.

#### A.8.1.1 Testing the Database Connection

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

You can connect to your database using the database configuration settings that are found in JasperReports Server. This validates the database hostname, port, username, and password that are being used.

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

#### A.8.1.2 Logging into PostgreSQL

Run the PostgreSQL client from the command line and try to connect to the database. For example:

```
psql -U postgres jasperserver
```

#### A.8.1.3 Logging into MySQL

Run the MySQL client from the command line and try to log in directly using the `root` user, for example:

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

You are prompted for a password for the user you specified on the command line. Enter the appropriate password to login. The default password used in the sample configuration scripts is `password` (jasperadmin in 2.1 and earlier).

#### A.8.1.4 Logging into Oracle

Start SQL\*Plus and try to log into Oracle directly. Three users are created during installation:

- `jasperserver` - schema user for the JasperReports Server metadata.
- `sugarcrm` - schema user for the SugarCRM sample data.
- `foodmart` - schema user for the foodmart sample data.

To log in as each of these users, supply the password specified during installation.

#### A.8.1.5 Logging into Microsoft SQL Server

Run the `sqlcmd` and try to log into MSSQL Server directly. For example:

```
sqlcmd -S localhost\jasperserver -d jasperserver -U jasperadmin -P password
```

### A.8.2 Case Sensitive Collation in SQL Server

In Microsoft SQL Server, setting the collation to be case-sensitive is not supported. When collation is case-sensitive in SQL Server, column and table names are also treated as case-sensitive. This can happen when setting a locale that includes case-sensitive collation and will cause an error:

```
[sql] Failed to execute:
INSERT INTO JIUserRole (userId,roleId) select u.id, r.id
from JIUser u, JIRole r
where u.username = \'anonymousUser\' and r.roleName = \'ROLE_ANONYMOUS\'
[sql] com.microsoft.sqlserver.jdbc.SQLServerException: Invalid column name \'roleName\'
```

Use a different locale or remove the case-sensitivity setting.

### A.8.3 Maximum Packet Size in MySQL

If you are upgrading or importing into a MySQL database and your repository contains large objects such as images, you may see an error such as:

```
ERROR 1153 (08S01): Got a packet bigger than 'max_allowed_packet' bytes
```

The default `max_allowed_packet` on the MySQL server is 1M (one Megabyte = 1,048,576 bytes). The most effective fix is to change this value in the server configuration to accommodate the largest resource stored in your repository. The server configuration file is typically named `my.cnf` (or `my.ini`) and is located in the MySQL root directory, although this may vary. Change the configuration setting to a larger value, for example:

```
max_allowed_packet = 16M
```

For more information, see <http://dev.mysql.com/doc/refman/5.0/en/packet-too-large.html>.

After changing this value, restart the MySQL server. Then perform the upgrade or import step again.

#### A.8.4 Case Sensitivity for Table and Column Names

Some databases are case-sensitive with respect to table names and will consider “customer” and “Customer” to be two different tables. If JasperReports Server is using a case-sensitive database, it’s important that the table names specified in query strings in the JRXML file of a saved report match the actual table names found in the database. A mismatch may occur if you are transferring data from one database to another, which may cause the capitalization of table names to change.

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

Under Linux MySQL, table and column names are case-sensitive. Linux MySQL can be configured to be non-case-sensitive by setting the configuration parameter `lower_case_table_names` to 1 in the `my.ini` or `my.cnf` file. For more information search the MySQL documentation for a section about identifier case sensitivity.

Table and column names in Oracle and PostgreSQL are case-sensitive.

### A.8.5 PostgreSQL: Job Scheduling Error

If the Quartz settings under the PostgreSQL database have not been updated to specify the driver delegate class specific to PostgreSQL you will get errors when you try and run a scheduled report. The errors would look similar to the following:

```
Error while fetching Quartz runtime information
org.quartz.JobPersistenceException: Couldn't obtain triggers: Bad value for type int
org.postgresql.util.PSQLException: Bad value for type int
```

If you see this error you will need to check your Quartz properties file found at the following location:

```
<tomcat>/webapps/jasperserver-pro/WEB-INF/js.quartz.properties
```

Make sure that the following property does not have the standard driver delegate, but instead has the PostgreSQL specific driver delegate. It should look like the following for PostgreSQL:

```
quartz.delegateClass=org.quartz.impl.jdbcjobstore.PostgreSQLDelegate
```

### A.8.6 Oracle js-install Script Hangs with Oracle 10g

If you plan to run the `js-install` script when installing to an Oracle database instance, check for the appropriate JDBC driver version. The default JDBC driver used for Oracle tends to be specific for newer Oracle versions. If you are installing to an older version of Oracle then the newer JDBC driver can cause execution to hang.

The default JDBC driver for Oracle can be seen in your `buildomatic/default_master.properties` file:

```
<js-install>/buildomatic/default_master.properties           (final file name and location)
<js-install>/buildomatic/sample_conf/oracle_master.properties (original file)
```

The settings for the default JDBC driver for Oracle are:

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

The settings above are commented out. Nevertheless, this is the JDBC driver that is used by default. These settings are appropriate for an Oracle version 11.2 instance.

If you are using Oracle 10.2, remove the comments from these settings and change them to:

```
maven.jdbc.artifactId=ojdbc14
maven.jdbc.version=10.2.0
```

The location of the Oracle JDBC drivers is here:

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

### A.8.7 Performance Issues with Oracle JDBC Queries

Setting the Oracle database localization option `defaultNChar` to `true` can substantially impact the performance of JDBC queries. When `defaultNChar` is set to `true`, the database will implicitly convert all `CHAR` data into `NCHAR` when you access `CHAR` columns. If you do not need to support UTF-8 for your Oracle database, you can omit this setting.

The option you need and how to set it depends on your version of Java, your application server, and how it is deployed. For information about changing a JVM option setting for your particular environment, see your application server documentation.

To change this setting on Windows, enter a command such as the following at the command line:

```
set JAVA_OPTS=%JAVA_OPTS% -Doracle.jdbc.defaultNChar=false
```

To change this setting on Linux, enter a command such as the following at the command line:

```
export JAVA_OPTS="$JAVA_OPTS -Doracle.jdbc.defaultNChar=false"
```



## A.8.8 Using an Oracle Service Name

If your Oracle database is configured to use a service name instead of an Oracle system identifier (SID), set up the service name by updating your `default_master.properties` file before using `buildomatic`:

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

In `default_master.properties`, uncomment the `serviceName` property and enter your Oracle service name, for example:

```
serviceName=ORCL
```

When you are using an Oracle service name, make sure not to set the `SID` or `dbPort` in the `default_master.properties` file.

## A.8.9 Oracle Error on Upgrade when PL/SQL Not Enabled

If you are upgrading to JasperReports Server version 5.0 or later using the `js-upgrade-samedb.sh/bat` script, you can encounter an error if Oracle's Procedural Language (PL/SQL) is not enabled.

The upgrade script used to upgrade to a 4.7 database from a 4.5 database requires the PL/SQL language to be enabled. The script is located here:

```
buildomatic/install_resources/sql/oracle/upgrade-oracle-4.5.0-4.7.0-pro.plsql
```

The error you encounter might look something like the following:

```
[advanced-sql] PLS-00103: Encountered the symbol "end-of-file"
```

If your PL/SQL language is not enabled please consult the documentation for your Oracle database to enable PL/SQL.

## A.8.10 Error Running Scheduled Report

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

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

the problem may be the default size of the MySQL blob datatype. You can increase the size of this datatype by updating your `my.ini` or `my.cnf` MySQL configuration file with the following setting:

```
max_allowed_packet=32M
```

## A.8.11 Error Running a Report

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

One common problem with an individual report is the data source being used. To validate a data source connection:

1. Log into JasperReports Server as a user with administrative permissions and locate the report unit that returns errors.
2. Select the report and click the **Edit** button in the toolbar to identify the data source the report uses. The data source name is found on the fourth edit page.
3. Select this data source in the repository and click the **Edit** button in the toolbar.
4. Review the information specified for this data source.
5. Click the **Test Connection** button in order to validate the connection.

If the connection fails, perhaps the application server cannot find the driver for the data source. For example, in a default installation of JasperReports Server, Tomcat looks for data source drivers in `<js-install>/apache-tomcat/lib`.

6. Click **Save** or **Cancel** when you are done.
7. Test your report. If it still returns errors, edit the data source again and try checking other values, such as the port used by the database.

## A.8.12 Save Error with DB2 Database

When using the DB2 database as your repository database, it is possible to get errors when saving longer strings (such as over 50 characters) to data entry fields in the UI. An example would be saving a resource that has a name over 50 characters long. It is possible to get an error similar to the following:

Expected status code is 200, but was 400. Response body contained:

An unexpected exception has occurred

The problem, in this case, is that DB2 handles UTF-8 characters differently than other JasperSoft certified databases. The database columns holding these strings need to be made larger. The fix for this problem is targeted to the JasperReports Server release following the 5.5 release.

In the meantime, when DB2 is used as the repository database, there is a limitation on the number of characters that can be entered into UI fields.

## A.9 Application Server-related Problems

### A.9.1 Memory Issues Running Under Tomcat

If you experience problems related to the release of memory or to container tag pooling, the following steps might solve the problem:

1. Set the following parameter in the global \$CATALINA\_BASE/conf/web.xml:  
enablepooling = false
2. Restart Tomcat.

### A.9.2 Java Out of Memory Error

If you encounter a Java out of memory error, try increasing your Java heap size setting. See section [6.1, “Setting JVM Options for Application Servers,” on page 41](#). As a minimum, add `-Xms1024m -Xmx2048m` to your `JAVA_OPTS` setting.

This Java option is set within the application server, so you must set it then restart your application server.

### A.9.3 Configuration File Locations

JasperReports Server configuration properties are found in the following files, depending on your application server.

The following list shows the location of the properties for supported application servers:

|            |                                                                                                                              |
|------------|------------------------------------------------------------------------------------------------------------------------------|
| Tomcat:    | <code>&lt;tomcat&gt;/webapps/jasperserver-pro/META-INF/context.xml</code>                                                    |
|            | <code>&lt;tomcat&gt;/webapps/jasperserver-pro/WEB-INF/hibernate.properties</code>                                            |
|            | <code>&lt;tomcat&gt;/apache-tomcat/webapps/jasperserver-pro/WEB-INF/web.xml</code> (JNDI config)                             |
|            | <code>&lt;tomcat&gt;/apache-tomcat/config/Catalina/localhost/jasperserver-pro.xml</code> (delete: see below)                 |
| JBoss 5:   | <code>&lt;jboss&gt;/server/default/deploy/js-postgresql-ds.xml or js-oracle-ds.xml or js-&lt;database name&gt;-ds.xml</code> |
|            | <code>&lt;jboss&gt;/server/default/deploy/jasperserver-pro.war/WEB-INF/hibernate.properties</code>                           |
|            | <code>&lt;jboss&gt;/server/default/deploy/jasperserver-pro.war/WEB-INF/web.xml</code>                                        |
|            | <code>&lt;jboss&gt;/server/default/deploy/jasperserver-pro.war/WEB-INF/jboss-web.xml</code>                                  |
| GlassFish: | <code>&lt;glassfish&gt;/domains/domain1/autodeploy/jasperserver-pro.war/WEB-INF/hibernate.properties</code>                  |
|            | <code>&lt;glassfish&gt;/domains/domain1/autodeploy/jasperserver-pro.war/WEB-INF/js.quartz.properties</code>                  |
|            | <code>&lt;glassfish&gt;/domains/domain1/config/domain.xml</code>                                                             |

## A.9.4 Context.xml under Tomcat: Special Case

If you deploy multiple instances of JasperServer to Tomcat, the context.xml (database connection configuration) can be superseded by a file in this location: `<tomcat>/conf/Catalina/localhost/jasperserver-pro.xml` file. This is the case with some Tomcat versions before Tomcat 7.

When JasperServer is deployed, the context.xml will be copied to `<tomcat>/conf/Catalina/localhost/jasperserver-pro.xml` (Tomcat does this by default).

Now, if you make changes to your `<tomcat>/webapps/jasperserver-pro/META-INF/context.xml`, Tomcat will not “see” them. Instead, the `jasperserver-pro.xml` will be used. This is confusing, but is the way that Tomcat operates.

If you edit your context.xml to fix a database problem:

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

Remember to delete the `jasperserver-pro.xml` file:

```
<tomcat>/conf/Catalina/localhost/jasperserver-pro.xml    (delete this file)
```

## A.9.5 Tomcat 6 Installed Using apt-get

If you are installing JasperReports Server to an instance of Tomcat that has been installed using a package managers such as apt-get, yum, or rpm then you can use the CATALINA\_HOME and CATALINA\_BASE properties found in your default\_master.properties file.

Go to the section of the default\_master.properties that looks like this:

```
# Tomcat app server root dir
appServerDir = C:\\Program Files\\Apache Software Foundation\\Tomcat 7.0
# appServerDir = /home/devuser/apache-tomcat-7.0.26
# if linux package managed tomcat instance, set two properties below
# CATALINA_HOME = /usr/share/tomcat6
# CATALINA_BASE = /var/lib/tomcat6
```

and change it to the following:

```
# Tomcat app server root dir
# appServerDir = C:\\Program Files\\Apache Software Foundation\\Tomcat 7.0
# appServerDir = /home/devuser/apache-tomcat-7.0.26
# if linux package managed tomcat instance, set two properties below
CATALINA_HOME = /usr/share/tomcat6
CATALINA_BASE = /var/lib/tomcat6
```

Note that you must set both CATALINA\_HOME and CATALINA\_BASE.

## A.9.6 GlassFish Modifications

### A.9.6.1 Using a Custom Domain

If your application server is GlassFish and you’re using a custom domain, set up the following authentication information in the default\_master.properties:

```
# Glassfish domain name (default is domain1)
glassfishDomain=domain1

# Glassfish domain port (default is 4848), user (default is admin) and password.
# Uncomment and set up next parameters if you install JasperServer to the custom Glassfish
domain (not default)
#glassfishPort=4848
#glassfishUser=admin
#AS_ADMIN_PASSWORD=adminadmin
```

### A.9.6.2 Using GlassFish 3.1.0

There is a known issue with GlassFish 3.1.0 where Java JVM options are not properly set. This issue is fixed in GlassFish 3.1.1 and later.

#### To set the JVM options in GlassFish 3.1.0:

1. Open this buildomatic property file:  
`<js-install>/buildomatic/default_master.properties`
2. Add the `glassfishPort` property as follows:  
`glassfishPort=4848`

### A.9.6.3 Can't Upload Files on GlassFish 3.1.2

There is a known issue with file upload on GlassFish 3.1.2 which prevents you from uploading files to the JasperReports Server (GLASSFISH 18446). This issue is resolved in the 3.1.2.2 release, which is a microrelease that resolves this and other critical issues.

### A.9.6.4 Requests to Single Permissions REST2 Service fail on GlassFish

Requests to Single Permissions REST2 service are failing on GlassFish with the following error:

```
400 Invalid URI: Encoded slashes are not allowed by default. To enable encodedslashes, set the property com.sun.grizzly.util.buf.UDecoder.ALLOW_ENCODED_SLASH to true
```

To fix this issue, perform the following command:

```
./bin/asadmin create-jvm-options -  
Dcom.sun.grizzly.util.buf.UDecoder.ALLOW_ENCODED_SLASH=true
```

## A.9.7 JBoss Modifications

### A.9.7.1 JBoss 7 Startup Error

JBoss7 has a default startup time period. If your JBoss 7 takes longer than 60 seconds to startup or deploy, you may receive the following error:

```
"(DeploymentScanner-threads - 1) Did not receive a response to the deployment operation within the allowed timeout period [60 seconds]. Check the server configuration file and the server logs to find more about the status of the deployment".
```

To fix this, you need to increase your `deployment-timeout` setting as follows:

1. Change to the JBoss standalone configuration directory.  
`cd <jboss>/standalone/configuration`
2. Open the `standalone.xml` file.
3. Look for the `<subsystem xmlns="urn:jboss:domain:deployment-scanner:1.1">` element, for example:  

```
<subsystem xmlns="urn:jboss:domain:deployment-scanner:1.1">  
  <deployment-scanner path="deployments" relative-to="jboss.server.base.dir" scan-  
    interval="5000"/>  
</subsystem>
```
4. Edit this to add or set the attribute `deployment-timeout` to the desired amount of time in seconds, for example:  

```
<subsystem xmlns="urn:jboss:domain:deployment-scanner:1.1">  
  <deployment-scanner path="deployments" relative-to="jboss.server.base.dir" scan-  
    interval="5000" deployment-timeout="600"/>  
</subsystem>
```
5. Save the file.

On server restart, your system will have the specified amount of time to start up.

### A.9.7.2 JBoss Large INFO Log Message on Drill-through

JBoss has an internal mechanism to track and log information on unclosed JDBC connections. JasperSoft OLAP Views leaves a connection open for performance reasons when doing a drill-through. In this case, JBoss puts a large INFO level message into the server.log.

**To silence this INFO message, perform these steps:**

1. Open the JBoss log4j configuration file for editing:  
`<jboss>/server/default/conf/jboss-log4j.xml`
2. Set the logging level for the `CachedConnectionManager` class to the following value:

```
<category name="org.jboss.resource.connectionmanager.CachedConnectionManager">
  <priority value="WARN"/>
</category>
```

### A.9.7.3 JBoss 5.0.1 and 5.1.x Error

With JBoss 5.0.1 and 5.1.x, you might see the following error:

```
org.jboss.xb.binding.JBossXBRuntimeException: Failed to create a new SAX parser
Caused by: java.lang.ClassCastException
```

This error might have occurred with older releases of JasperReports Server. Newer releases use a jar newer than 2.7.1 so this problem should not occur. Additionally, the `xercesImpl-*.jar` is automatically deleted when installing using the buildomatic scripts.

This is a class conflict with the `xercesImpl-2.7.1.jar` in JasperReports Server. To correct it, delete the following file:

```
<jboss>/server/default/deploy/jasperserver-pro.war/WEB-INF/lib/xercesImpl-*.jar
```



When running the buildomatic scripts to deploy to JBoss, the `xercesImpl-*.jar` file is automatically deleted in order to fix this problem.

#### A.9.7.4 AttachmentStore Error in JBoss 5.1

With JBoss 5.1 you might see the following error:

```
DEPLOYMENTS IN ERROR:
```

```
Deployment "AttachmentStore" is in error due to: java.lang.IllegalArgumentException: Wrong
arguments. new for target java.lang.reflect.Constructor expected=[java.net.URI]
actual=[java.io.File]
```

This is a known JBoss issue. The resolution is to edit the file `jboss-5.1.0.GA\server\default\conf\bootstrap\profile.xml` to include `java.io.File`:

```
<bean name="AttachmentStore"
class="org.jboss.system.server.profileservice.repository.AbstractAttachmentStore">
<constructor><parameter class="java.io.File"><inject
bean="BootstrapProfileFactory" property="attachmentStoreRoot"
/></parameter></constructor>
<property name="mainDeployer"><inject bean="MainDeployer" /></property>
<property name="serializer"><inject bean="AttachmentsSerializer" /></property>
<property name="persistenceFactory"><inject bean="PersistenceFactory"
/></property>
</bean>
```

#### A.9.7.5 Using a Non-default JBoss Profile

If your application server is JBoss and you're using a profile other than the default, you need to set the `jboss.profile` property before running the `js-install` script in section 5.2, [“Installing the WAR File Using js-install Scripts,” on page 34](#):

1. Open this buildomatic property file:

```
<js-install>/buildomatic/build_conf/default/app.srv.properties
```

2. Uncomment the `jboss.profile` property and change the profile name as follows:

from

```
# jboss.profile = default
```

to

```
jboss.profile = <your_profile>
```

#### A.9.7.6 Using JBoss with Non-Latin Characters

If your application server is JBoss, and your organization is created with non-Latin characters, you will need to edit the `standalone.xml` configuration file.

1. Edit `<jboss-home>/standalone/configuration/standalone.xml`

2. Add after `<extensions>` tag new tag:

```
<extensions>
.....
</extensions>

<system-properties>
  <property name="org.apache.catalina.connector.URI_ENCODING"
value="UTF-8"/>
  <property
name="org.apache.catalina.connector.USE_BODY_ENCODING_FOR_QUERY_STRING"
value="true"/>
</system-properties>
```

### A.9.7.7 JBoss 4.2 XML/A Connection Fix

JBoss 4.2 includes the JBossWS service as a standard, default feature. JasperReports Server has web services support for XML/A connections.

The web services classes in JasperReports Server and JBoss can conflict and cause the following error when attempting to utilize a JasperReports Server XML/A connection:

```
javax.xml.soap.SOAPException: Unable to create message factory for
SOAP: org.jboss.ws.core.soap.MessageFactoryImpl
```

To prevent the web services class conflict, set the special Java JVM options for JBoss 4.2, as described in section [6.1.1, “Tomcat and JBoss JVM Options,”](#) on page 41.

## A.9.8 Websphere Modifications

### A.9.8.1 Page Not Found Error on Login

This error is seen during a WebSphere installation when the user attempts to log into JasperReports Server. After typing in a correct user ID and password, the user sees an error page: Page cannot be found, HTTP 404

Some WebSphere versions or fix packs have modified code that processes web server filters incorrectly. Components with the /\* url pattern get affected by this. JasperReports Server uses the Spring framework for authentication and it is mapped using a filter chain with the /\* url pattern. You need to set a special property that WebSphere provides to solve this problem.

#### To solve the Page Not Found Error on Login:

1. Login into WebSphere Administrative Console.
2. Navigate to **Application Servers > <server> > Web Container Settings > Web Container > Custom Properties.**
3. Create a new property with the following attributes:
 

```
name: com.ibm.ws.webcontainer.invokefilterscompatibility
value: true
```
4. Save the master configuration.
5. **Restart** the WebSphere server.

## A.9.9 Disabling User Session Persistence in Application Servers

JasperReports Server stores non-serializable data in its user sessions, which can cause errors after restarting your application server:

```
Exception loading sessions from persistent storage
Cause: java.io.NotSerializableException ...
```

The errors appear in the JasperReports Server log when users log in after the application server has been restarted. The errors do not appear to users, and they have no impact on JasperReports Server operations.

Because JasperReports Server user sessions are not persistent, you can configure your application server to disable persistence and avoid the error. For example, in Apache-Tomcat 5.5, 6, and 7 edit the file <tomcat>/conf/context.xml and locate the following lines:

```
<!-- Uncomment this to disable session persistence across Tomcat restarts -->
<!--
<Manager pathname="" />
-->
```

Remove the comment markers from lines 2 and 4 above, then restart Apache-Tomcat for the change to take effect. For other application servers, refer to the product documentation.

## A.9.10 Session Error Using JasperReports Server and Tomcat 7

On some versions of Tomcat 7, a session error might occur while running reports, with the log error “A request has been denied as a potential CSRF attack.” This is due to a known conflict between security settings in Direct Web Remote library (DWR) 2.x and some versions of Tomcat 7.0.x:

- Tomcat 7 sets httpOnly on session ID cookies to safeguard against cross-site scripting (XSS) attacks.
- DWR 2.x uses session ID cookies to safeguard against cross-site request forgery (CSRF).

To work around this problem, you must modify these safeguards by doing one of the following:

- Disabling httpOnly for cookies in Tomcat

**OR**

- Allowing requests from other domains in DWR

For more information on the security impact and relative risks of these two choices, see, for example, the Cross-site Scripting and Cross-site Request Forgery pages at the [Open Web Application Security Project \(OWASP\)](http://www.openwasp.org/).

### A.9.10.1 Disabling httpOnly for Cookies in Tomcat

The application server that hosts JasperReports Server handles the session cookie. To prevent malicious scripts on a client from accessing the session cookie, and thus the user connection, Tomcat 7 is set to use httpOnly cookies. This tells the browser that only the server may access the cookie, not scripts running on the client. When enabled, this setting safeguards against XSS attacks.

You can disable this by setting httpOnly in the file <tomcat>/conf/context.xml:

```
<Context useHttpOnly="false">
...
</Context>
```

### A.9.10.2 Allowing Requests from Other Domains in DWR

DWR is a server-side component used for Input Controls. By default, DWR uses session ID cookies to prevent against cross-site request forgery. You can disable the protection in DWR by setting the `crossDomainSessionSecurity` parameter for the `dwr` servlet in the file <tomcat>\webapps\jasperserver-pro\WEB-INF\web.xml:

```
<servlet>
  <servlet-name>dwr</servlet-name>
<servlet-class>org.directwebremoting.spring.DwrSpringServlet</servlet-class>
  ...
  <init-param>
    <param-name>crossDomainSessionSecurity</param-name>
    <param-value>false</param-value>
  </init-param>
</servlet>
```



## A.10 License-related Errors

### A.10.1 License Not Found Errors

Normally, the JasperReports Server installer includes an evaluation license file that you replace with a commercial license file, as described in section 2.10.3, “[Installing a New License File](#),” on page 23. If JasperReports Server returns an error after you replace the license file, the most likely causes are:

- You did not clear your application server’s work directory, as explained in section 2.10.3. Delete the work directory, restart the application server, and try logging into JasperReports Server again.
- The `Djs.license.directory` property in your application server startup environment is incorrectly set:

For example, in Windows the correct setting looks like this:

```
-Djs.license.directory=<js-install>
```

In Linux, the correct setting looks like this:

```
-Djs.license.directory=/opt/jasperreports-server-5.0
```

The specified directory must contain the license file, named `jasperserver.license`. The property is typically set for your application server in the environment startup script. It must contain the location of your license file, which is typically in the `<js-install>` directory:

Tomcat: `<tomcat>/bin/setclasspath.bat/.sh` or `bin/setenv.bat/.sh`

JBoss: `<jboss>/bin/run.bat` or `.sh`

### A.10.2 License Not Found or License Corrupt Error with Tomcat as a Service

If you have an existing Tomcat running as a service under Windows, the installer attempts to make the proper updates so that the JasperReports Server license file is found at application server startup time. If the installer is unsuccessful, be sure that you took the steps described in section 2.10.4, “[License File for Existing Tomcat as Windows Service](#),” on page 24.

## A.11 Problems Importing and Exporting Data from the Repository

### A.11.1 Exporting a Repository That Contains UTF-8

The following errors may happen when you have international characters in repository objects, for example, in user IDs.

#### A.11.1.1 Error During Export

Upgrading typically requires doing an export operation on your database. If you are using MySQL and getting this null pointer exception, it may be due to an incorrect character in the file `js.jdbc.properties`:

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

Check the URL in this file in `<js-install>buildomatic/build_conf/default/`; it should look like the following:

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

Note the ampersand `&` character. It is incorrect if it appears as `&amp;`. The `&amp;` is only correct in an HTML or XML context. It is incorrect in a properties file.

### A.11.1.2 Error During Export from Repository on Oracle

Oracle requires a specific JVM property to handle UTF-8 characters properly. When this error happens, the export is empty and an error occurs when attempting to compress the result:

```
ERROR ExporterImpl:129 - java.util.zip.ZipException: ZIP file must have at least one entry
```

If you have stored your repository database on an Oracle RDBMS, modify the last line of both `<js-install>/buildomatic/js-export.*` files as follows:

```
From: java -classpath ...
```

```
To: java -Doracle.jdbc.defaultNChar=true -classpath ...
```

## A.12 Problems with Upgrade

### A.12.1 DB2 Script Error on Upgrade

If you are upgrading from 4.7 to 5.0, and you use the `js-upgrade-samedb.bat/sh` script, it is possible to get an error related to an inadequate `PAGESIZE` setting. The recommended minimal `PAGESIZE` setting for both 4.7 and 5.0 is 16384. If the `PAGESIZE` setting is less than this you can get an error such as the following:

```
com.ibm.db2.jcc.am.SqlException:
DB2 SQL Error: SQLCODE=-670,SQLSTATE=54010, SQLERRMC=4005;
or
com.ibm.db2.jcc.am.SqlSyntaxErrorException:
DB2 SQL Error: SQLCODE=-286, SQLSTATE=42727, SQLERRMC=8192;DB2ADMIN,
DRIVER=4.11.77
```

If you get this error, recreate your 4.5 database using a command such as the following:

```
CREATE DATABASE JSPRSVR USING CODESET UTF-8 TERRITORY US PAGESIZE 16384
```

This may require reloading your database from your database backup and rerunning the upgrade procedure after `PAGESIZE` has been changed.

### A.12.2 Include Audit Events on Upgrade

If you have auditing enabled and you run upgrade using `js-upgrade-newdb.bat/sh`, audit events are not imported by default. To import audit events, you need to run an additional command after the `js-upgrade-newdb` script completes. To do this, change to the `buildomatic` directory:

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

Then run one of the following commands (for example):

```
js-import.bat --input-zip=js-my-export-all.zip --include-audit-events (Windows)
```

```
js-import.sh --input-zip=js-my-export-all.zip --include-audit-events (Linux)
```

or

```
ant import -DimportFile=js-my-export-all.zip -DimportArgs="--include-audit-events"
```

These commands reimport all resources from the specified export file. Existing resources will not be overwritten and the audit event will be added.



When using either import utility, the server must be stopped to avoid issues with caches, configuration, and security.

## A.13 Installing on Windows XP if Memory Error

With some instances of Windows XP, we have found that JasperReports Server is not able to cleanly startup after running the binary installer (i.e. `jasperreports-server-<ver>-windows-x86-installer.exe`). The problem has been that Windows XP is not providing enough contiguous memory for JasperReports Server to successfully startup.

When this is the case, there will be an error message seen in the Apache Tomcat log (`<js-install>/apache-tomcat/logs/Catalina.out`) similar to this:

Not enough reserve enough space for object heap

The solution to this issue is to either decrease the Java memory requested by the Tomcat application, or to install to a 64 bit Windows machine (which is able to map more system memory).

To decrease the Java memory requested by the Tomcat service put into place by the binary installer, you can utilize the following steps:

- ♦ Edit the file `<js-install>/apache-tomcat/bin/service.bat`
- ♦ Look for the line (toward the end of the file) that defines the Java memory settings
  - ♦ `-Xms512M; -Xmx1024M`
- ♦ Change this value to the following
  - ♦ `-Xms256M; -Xmx512M`
- ♦ Open a Windows Command shell, Start > Run > cmd
  - ♦ `cd C:\Jaspersoft\jasperreports-server-<ver>\apache-tomcat\scripts`
  - ♦ (Note: after you run the following command the CMD shell will disappear)
  - ♦ `serviceinstall.bat REMOVE`
- ♦ Next, make sure that your PostgreSQL database is running
  - ♦ Go to the Windows Services Panel, and check the status of `jasperreportsPostgreSQL`
  - ♦ Click to start the PostgreSQL service
- ♦ Open a new Windows Command shell, Start > Run > cmd
  - ♦ `cd C:\Jaspersoft\jasperreports-server-<ver>\apache-tomcat\scripts`
  - ♦ (Note: after you run the following command the CMD shell will disappear)
  - ♦ `serviceinstall.bat INSTALL`
  - ♦ The command above will install the Tomcat service and leave Tomcat running
- ♦ Log into JasperReports Server to see that everything is working cleanly
  - ♦ `http://localhost:8080/jasperserver-pro`

## A.14 Property File Updates

### A.14.1 Owasp.CsrfGuard.properties - Security File

There is a set of property files that help control the settings for the JasperReports Server functionality that prevents Cross-Site Request Forgery (CSRF). The key files are described in the *JasperReports Server Administrator Guide*.

One of these properties files refers to the JasperReports Server “webapp name”, but it is not currently automatically updated by the installation procedure. This file is at the following location:

`<appserver-path>/jasperserver-pro/WEB-INF/esapi/Owasp.CsrfGuard.properties`

If you change the name of your webapp from the default of “`jasperserver-pro`”, then you will also need to manually update the `Owasp.CsrfGuard.properties` file.

So, if you modify your `default.master.properties` like so:

```
# webAppNamePro = jasperserver-pro
webAppNamePro = jasperserver-inst2
```

And then do an installation (using the `js-install.sh` scripts), you will need to edit `Owasp.CsrfGuard.properties` like so:

Change:

```
org.owasp.csrfguard.NewTokenLandingPage=/jasperserver-pro/login.html  
org.owasp.csrfguard.action.Redirect.Page=/jasperserver-pro/login.html
```

To:

```
org.owasp.csrfguard.NewTokenLandingPage=/jasperserver-inst2/login.html  
org.owasp.csrfguard.action.Redirect.Page=/jasperserver-inst2/login.html
```

This issue is planned to be fixed in the next JasperReports Server release.

## A.15 Permissions Error in Overlay Upgrade with PostgreSQL and Bundled Jaspersoft Installation

If you are using the overlay upgrade procedure with Postgres as your database and you installed an earlier version of JasperReports Server, you might see an HTTP Status 404 error. The jasperserver-por.log file will show something like:

```
ERROR ContextLoader,Thread-1:307 - Context initialization failed  
org.springframework.beans.factory.BeanCreationException: Error creating bean with name  
'proMaintenanceSchedulerTriggers' defined in ServletContext resource [/WEB-INF/  
applicationContext-audit.xml]: Cannot resolve reference to bean 'heartbeatTrigger' while  
setting constructor argument with key [1];
```

The occurs because if you use the Installer, in the default master properties file we have the user postgres and in context.xml we have the user jasperdb.

If this occurs, perform the following procedure:

1. Back up your existing master properties.
2. Change the DB user in the master properties to jasperdb.
3. Run the overlay upgrade.