

Loftware Web Services Installation Guide

Version 2.1.1

© 2009. All rights reserved. Version 2.1.1

Loftware, LLM, Loftware Label Design, Loftware Print Server, LPS, Loftware Connector, Global Marking Solutions, I-Push, and I-Pull are all registered trademarks of Loftware, Inc. Loftware WebAccess, LWA, and Loftware Web Services are trademarks of Loftware, Inc.



LOFTWARE[™]
ENTERPRISE LABELING SOLUTIONS

166 Corporate Drive, Portsmouth, NH 03801 U.S.A.

Tel: (603) 766-3630 Fax: 603) 766-3631

sales@loftware.com www.loftware.com

This page intentionally left blank

Contents

Contact Loftware	5
Professional Services.....	5
Technical Support.....	5
Customer Service.....	5
Traditional Mail.....	5
Technical Support	6
Premium Annual Support Contract.....	6
Before Calling Support.....	6
Licensing, Warranty, and Support	8
Introduction: Loftware Web Services Components	10
Loftware Web Services.....	10
Security Administration User Interface.....	11
Loftware Web Services Installation Options	12
Installation options.....	12
Section 1: Configuring Prerequisites	14
System Requirements.....	14
Third-Party Tools.....	14
Configuration Notes.....	14
About installing and configuring Apache Tomcat 6.0.18.....	15
Deploying Apache Axis2 1.4.1.....	16
Set Postgres Environment Variable in Windows.....	17
Set Postgres Environment Variable in Unix/Linux.....	17
Section 2: Installing Loftware Web Services on One Computer	18
What you need.....	18
Prepare the installation files.....	18
Run the LWSinstall Script.....	19
About Upgrading Loftware Web Services.....	20
Section 3: Installing Loftware Web Services Components Individually	22

About setting up the RBAC database using PostgreSQL.....	22
About installing Software Web Services on Unix/Linux.....	23
About installing Software Web Services on Windows.....	24
Section 4: Uninstalling Web Services.....	26
Uninstall Software Web Services.....	26
Uninstall Software Web Services Administration.....	27
Section 5: Configuring Authentication.....	28
About using the LDAPConfigurationTool to configure LDAP.....	28
About LDAP Authentication Modes.....	29
Generate encrypted passwords.....	31
Update the LDAPConfigurationTool.properties file.....	31
Create the LDAP Configuration Files.....	32
Update the security_settings.xml file.....	32
Verify the securityAuthentication.config file.....	33
Update the LDAPConnectionTemplate.xml file.....	33
Section 6: Configuring Software Web Services.....	36
Logging.....	36
Authentication.....	36
Cache Service.....	37
Web Services.....	37
Administration.....	37
About Configuring Logging in Software Web Services.....	37
Recreating the Public and Private Key Files.....	39
Customizing Software Web Services.....	40
Configuring the Cache Service in Software Web Services.....	46
About Configuring Software WebAccess Administration.....	48
Section 7: Accessing Software Web Services Administration.....	50
Access Software Web Services Security Administration.....	50

Professional Services

For consultation, implementation services, training or product optimization please contact Loftware's Professional Services Group.

Phone +1.603.766.3630 x209

E-mail psg@loftware.com

Technical Support

For installation and configuration questions, please contact Loftware's Technical Support department. Visit www.loftware.com for Loftware's technical support policies.

Phone +1.603.766.3630 x402

Fax +1.603.766.3635

E-mail techsupport@loftware.com

Customer Service

For licensing, product information, and ordering questions, please contact Loftware's Customer Service department. Please have your Serial Number and Registration information available, so we can provide service to you quickly and efficiently.

Phone +1.603.766.3630 x401

Fax +1.603.766.3631

E-mail customerservice@loftware.com

Traditional Mail

If you need to contact Loftware via traditional mail, FEDEX, UPS, or other mail service carriers, please use Loftware's shipping address.

Loftware, Inc.
166 Corporate Drive
Portsmouth NH 03801
U.S.A.

Software licenses purchased directly from Loftware include the first year of Technical Support. This initial 12-month support period starts on the day the product is shipped and invoiced from Loftware's factory. When needed, support recipients during this period are eligible to receive unlimited telephone support, access to software upgrades and enhancements and speak with our Systems Analysts.

Premium Annual Support Contract

To ensure uninterrupted telephone support as well as access to the latest software upgrades and enhancements, make sure all your software licenses remain under a Loftware Support Contract. After your first year of ownership, you will be sent a notice to renew your support contract. Please refer to Loftware's website for additional information on this very important topic, or if you prefer, call Loftware's Customer Service Department for more information.

During the one-year Support Contract period, Contract Subscribers have access to the following services:

1. Unlimited Technical Support Incidents
2. Access to Loftware's Professional Services Group
3. Automatically eligible to download software upgrades and service packs from our website
4. Automatic e-mail notification when new versions of software become available
5. When necessary, access to senior Loftware technical support staff, via phone and e-mail
6. Guaranteed software license replacement for accidentally damaged or malfunctioning hardware keys

Before Calling Support

Loftware has highly trained technicians available to help you with your labeling system. Technical support calls are not accepted until all of the following Technical Support requirements are met:

1. Your product is registered. If you have not registered your software, you may do so at <http://loftware.com> or via fax by using the form included with your software.
2. There is a Support Contract in place that covers the specific license in question.

3. You have checked the user's guide(s) for your answer. If you do not have the User's Guides, both of the guides or various chapters of each can be downloaded in PDF format from our web site, or read on-line. User manuals are also on the Loftware CD.
4. You have checked the Loftware's Knowledge Base articles on our <http://loftware.com>. Hundreds of frequently asked questions and typical problems are documented there in easy to read articles.
5. If you suspect that your problem is hardware related, try to first determine if it is a problem with your PC, Network, or printer and contact the appropriate company. Loftware does not sell or service any hardware products.
6. Have your serial number and version number of the product you are using ready. These numbers can be obtained by accessing the Help | About menu of the label design mode.
7. Think about how you are going to efficiently explain the problem prior to speaking with a technician. The better the description, the quicker the solution and/or resolution to your problem.
8. If this is a follow up call to a previous incident, please have the incident number ready.

Phone 603-766-3630 x402

Fax 603-766-3635

E-mail techsupport@loftware.com

Licensing, Warranty, and Support

The following documents are available in the “Documents” folder included with the CD-ROM or Internet download of the Loftware Software:

- Loftware End User License Agreement
- Loftware Third Party Terms and Conditions
- Loftware Software Services Support Agreement

This page intentionally left blank

Introduction: Loftware Web Services Components

Loftware Web Services

Loftware Web Services enables you to build custom applications in JAVA, .Net, and other technologies that integrate with the Loftware Print Server.

The Loftware Web Services Component includes the following components.

LoftwarePrintWS

The LoftwarePrintWS, a Web Service, enables access to label formats and label print operations using a browser. This allows the Loftware Web Services clients to send print jobs to LPS and access printers, labels and data sources.

Being compliant with current web service standards, the LoftwarePrintWS uses HTTP / HTTPS and SOAP (Simple Object Access Protocol), and publishes its services using WSDL (Web Services Description Language), an XML description of its data formats.

LoftwareAdminWS

The LoftwareAdminWS provides role-based access control to the security database and to user and license management.

Cache Service

The primary function of the Loftware Print Server (LPS) is to process print requests. It also returns a list of printers or labels to On Demand Print. These lists are large and retrieving them from LPS can impact system performance.

Loftware WebAccess includes a cache service to store printer and device information. This information is retrieved from LPS at regular intervals. The cache service improves performance and offloads commonly used LPS requests from web service clients.

Security Service

Loftware WebAccess includes a security component for user authentication and authorization. The Security Service provides role-based security by requesting credentials from a user (name, password, domain) to determine the activity or information that the user is authorized to access.

Security Database

The security service's Role Based Access Control (RBAC) database uses a custom database schema to maintain your list of authorized users and groups, as well as the specific Loftware services entities, roles, and permissions that you create.

Security Administration User Interface

You use the Loftware Web Services Security Administration user interface to control access to the LoftwarePrintWS. See "About Configuring Loftware WebAccess Administration" on page 48 for information on setting up Loftware Web Services Security Administration.

To control access to the print service, you configure rules, roles, users, and groups. In Loftware Web Services Security Administration:

- You create rules to define groups of devices and actions such as editing and printing labels.
- You use roles that are defined by rules.
- You create users that have roles, and belong to groups.
- You build groups that have roles, and contain users.

For detailed information on Loftware Web Services Security Administration see the *Loftware WebAccess Security Administration User's Guide* available in the Documents folder of your installation CD or download and from the Downloads section of www.ftware.com.

Loftware Web Services Installation Options

This guide describes the installation and configuration of Loftware Web Services.

Note: Loftware Web Services supports only the Apache Tomcat Web Server.

All references to *\$tomcat* in this guide indicate the root folder of the Apache Tomcat Web Server installation.

For example

Windows:
C:\Program Files\Apache Software Foundation\Tomcat 6.0.

Unix:
/usr/local/tomcat

Installation options

You can install all the components of Loftware Web Services on a single computer, or you can install each component on a different computer. See "Introduction: Loftware Web Services Components" on page 10

To install all components on one computer, you can use the LWSInstall script. See "Installing Loftware Web Services on One Computer" on page 18 for instructions on installing Loftware Web Services on a single computer.

See "Installing Loftware Web Services Components Individually" on page 22 for instructions on installing each component of Loftware Web Services on a different computer.

This page intentionally left blank

Section 1: Configuring Prerequisites

Loftware Web Services can be installed on Unix/Linux and Windows systems. The following must be installed and configured before the Loftware Web Services components are installed:

- An accessible Loftware Print Server version 9.5.x or greater with the required licenses. We recommend that you install Loftware Web Services and the Loftware Print Server on different computers.

Note: Loftware Web Services works with a single LPS.

System Requirements

System requirements are dependent on the number of concurrent users you have. Loftware's Professional Services Group (PSG) will help determine your system requirements.

Third-Party Tools

Loftware Web Services requires the following third-party software on the application server:

- Java Runtime Environment (JRE) 1.6_03(www.java.com)
- Apache Tomcat 6.0.18 See "Install Tomcat 6.0.18" on page 15.
- Apache Axis2 1.4.1 See "Deploying Apache Axis2 1.4.1" on page 16
- PostgreSQL 8.3.3 See "Set up the RBAC Database" on page 22
- LDAP - LDAP is not required. However, if LDAP is going to be used, you need permission to get into the existing LDAP server, or you will need to set up your own LDAP server.
- Internet Explorer Version 7 or Firefox Version 2 or 3 to access the Loftware Security Administration user interface.

Configuration Notes

- Loftware recommends that you install and configure Loftware Web Services on a clean computer (Except for the third-party tools listed, which should be installed before the Loftware Web Services components.)

- Software does not support Software Web Services if it is installed in a clustered-server environment. However, Software Web Services can communicate with a Software Print Server that is installed in a clustered-server environment.
- Software Web Services does not support labels created in versions of the Software Print Server prior to 9.0. To use a pre-9.0 label in Software Print Server, open the label in a 9.x version of Software Label Manager, and save it. Labels must have a Label Description. Check the Label Properties tab of Software Label Manger Media Setup.

About installing and configuring Apache Tomcat 6.0.18

For complete instructions on installing, configuring, and deploying in Tomcat, refer to the documentation for Tomcat 6.0.18.

Note: All references to **\$tomcat** indicate the root folder of the Apache Tomcat installation. In Windows the default installation path is, C:\Program Files\Apache Software Foundation\Tomcat 6.0. In Unix/Linux the default installation path is /usr/local/tomcat.

Install Tomcat 6.0.18

1. Install Apache Tomcat version 6.0.18
2. Create an administrative user in Tomcat.

Note: If you install Tomcat using the Windows installer, you can create an administrative user using the Install Wizard.

For example

To manually create an administrative user, add the following to the tomcat-users.xml file located in \$tomcat/conf:

```
<user username="administrator" password="password" roles="manager"/>
```

The section of tomcat-users.xml may look like the following:

```
<tomcat-users>
  <role rolename="manager"/>
  <user username="administrator" password="password"
roles="manager"/>
</tomcat-users>
```

Logging into the Apache Tomcat Manager console using these credentials allows you to stop, start, redeploy, or undeploy the Software WebAccess web applications.

Configure Tomcat Settings

Use the following instructions to configure Tomcat with Software Web Services.

1. Open **\$tomcat\bin\tomcat6w.exe**.

Note: See "Configure Tomcat Settings Alternative" on page 16 if you are starting Tomcat using catalina.bat.

2. Select the Java tab, and add the following, as two separate lines, to Tomcat's Java Options.

```
-Djava.security.auth.login.config=securityAuthentication.config
-XX:MaxPermSize=128m
```

3. Select the Startup tab, and add "\bin" to the end of the Working Path.
4. Select the Shutdown tab, and add "\bin" to the end of the Working Path.
5. Click **OK**.

Configure Tomcat Settings Alternative

If you will not use tomcat6w.exe to start Tomcat, you can configure Tomcat settings on a Unix/Linux or Windows systems using the following procedure.

1. Open the *\$tomcat*\bin\catalina.bat(sh) file.
2. Verify that the following lines appears in the catalina.bat(sh) file located in *\$tomcat*\bin:

Windows

```
set LOFTWARE_OPTS=-XX:MaxPermSize=128m
-Djava.security.auth.login.config=securityAuthentication.config
```

Unix/Linux

```
export LOFTWARE_OPTS='-XX:MaxPermSize=128m
-Djava.security.auth.login.config=securityAuthentication.config'
```

3. Add the LOFTWARE_OPTS setting to each of the java command lines at the end of catalina.bat(sh).

For example (Windows)

```
%_EXECJAVA% %JAVA_OPTS% %CATALINA_OPTS% %LOFTWARE_OPTS% %DEBUG_OPTS%
-Djava.endorsed.dirs="%JAVA_ENDORSED_DIRS%"
-classpath "%CLASSPATH%" -Dcatalina.base="%CATALINA_BASE%"
-Dcatalina.home="%CATALINA_HOME%"
-Djava.io.tmpdir="%CATALINA_TMPDIR%" %MAINCLASS% %CMD_LINE_ARGS%
%ACTION%
```

Deploying Apache Axis2 1.4.1

1. Download Apache Axis2 / Java Version 1.4.1 from <http://ws.apache.org/axis2/>.
2. Stop Tomcat.
3. Copy the axis2.war file to *\$tomcat*/webapps folder.

For example

Windows:
C:\Program Files\Apache Software Foundation\Tomcat 6.0

Unix:
/usr/local/tomcat

Start Tomcat

Start Tomcat. Tomcat automatically deploys the axis2.war files on startup.

Set Postgres Environment Variable in Windows

1. Right-click on the My Computer icon, select **Properties**.
2. Select the “Advanced” tab
3. Click **Environment Variables**.
4. Select **Path** from the **System Variables** section and edit the path.
5. Add a “;” and then the path to the bin directory where Postgres is installed (the psql.exe file should be in this directory)
6. Open a new command prompt and type “path” – verify the path just added is returned.

Set Postgres Environment Variable in Unix/Linux

1. Open a terminal
2. Type “which psql” and verify that it returns a path

Example:

This is OK

```
# which psql
/export/home/postgres/8.3/bin/psql
```

This is not OK

```
# which psql
no psql in /usr/bin /usr/ucb /etc /usr/sbin . /export/home/jdk1.6.0_
03/bin /export/home/postgres/8.3/bin
```

3. If you receive the “no psql” result, then you need to add it to your path.
4. Type “export PATH=\$PATH:/export/home/postgres/8.3/bin (assuming this is the path to the postgres bin directory on your Unix/Linux system)
5. Type “echo \$PATH” and verify that the path to postgres is listed in the returned set of paths.

Section 2: Installing Loftware Web Services on One Computer

You can install all Loftware Web Services components on a single computer by running the LWS Install script. This script performs the following installation and configuration tasks:

- Deploys Loftware WebAccess, Web Services and Administration in Tomcat.
- Installs the Loftware RBAC Database, and creates the database user.
- Adds the LOFTWARE_HOME environment variable to the Tomcat context.xml file.

What you need

The install script requires the following files, found on the Loftware Web Services CD, on both Windows and Unix:

- LoftwareWebAccessInstall.properties
- LWSInstall.bat (Windows)
- LWSInstall.sh (Unix)
- lwaservices.zip
- LWAInstall.jar
- jdom.jar
- loftwareadmin.zip

Prepare the installation files

- Copy the contents of the Loftware Web Services CD to a temporary folder on your system.

Update the LoftwareWebAccessInstall.properties file

1. Edit the LoftwareWebAccessInstall.properties file using a text editor such as Notepad in Windows or VI in Unix.
2. Set the InstallDatabase property to true or false.

Note: The LWAInstall script will not reinstall the RBACDatabase or user role if it exists. To reinstall the Database you must first uninstall the database, drop the database user role, and then reinstall the database and user role.

3. Note the DatabaseProduct property. Software Web Services supports PostgreSQL.
4. Set the DatabaseInstallPath to the location of PSQL.

Note: The `SoftwareWebAccessInstall.properties` file contains default database install paths for Windows and Unix. You can choose the default path for your operating system by removing the # character before the `DatabaseInstallPath` property, and entering a # character in front of the `DatabaseInstallPath` for the other operating system.

5. Note the Web server settings. Software Web Services supports Tomcat Web Server with Axis2 Web Services engine.
6. Set the WebServerInstallPath to the location where you installed your web server.

Note: The `SoftwareWebAccessInstall.properties` file contains default web server install paths for Windows and Unix. You can choose the default path for your operating system by removing the # character before the `WebServerInstallPath` property, and entering a # character in front of the `DatabaseInstallPath` for the other operating system.

7. Set the LPSIPAddress to the IP Address of the Software Print Server that you want Software Web Services to connect. If the Software Print Server is installed on a cluster, enter the IP address of the Software Print Server Virtual Server.

Run the LWSinstall Script

1. Stop Tomcat.
2. Set an environment variable called PGPASSWORD for the password of the PostgreSQL database `postgres` login role.

For example

Open a shell prompt in Unix, and enter:

```
setenv PGPASSWORD password
```

Open a command line in Windows, and enter:

```
C:> Set PGPASSWORD=password
```

3. Run the LWSinstall script from the temporary folder you created in "Prepare the installation files" on page 18.

For example

Open a shell prompt in Unix, and enter:

```
./LWSinstall.sh
```

Open a command line in Windows, and enter:

```
LWSTemp\> LWSinstall.bat
```

Note: The LWSinstall script creates a log file called `lwaInstall.log` in the temporary folder you created. If there are errors during installation, you can use this file to determine the cause.

4. Start Tomcat. The `lwaservices.war` and `loftwareadmin.war` files are extracted to the `$tomcat/webapps` folder.

About Upgrading Loftware Web Services

When you upgrade to a new version of Loftware Web Services you essentially replace one or more WAR files on the application server.

The process is similar to a new installation, except for the following:

- You must use different settings in `LoftwareWebAccessInstall.properties`.
- You must stop the application server (Tomcat).
- You must delete the WAR files and application folders that you are replacing.

Upgrade Loftware Web Services

1. Prepare the installation files. See "Installing Loftware Web Services on One Computer" on page 18.
2. Modify `LoftwareWebAccessInstall.properties`. "Update the `LoftwareWebAccessInstall.properties` file" on page 18. Use the following values:
 - Set `InstallDatabase` to `false`
 - Set `WebServerInstallPath` to the Tomcat directory
 - Set `LPSIPAddress` to the IP Address for the Loftware Print Server you want to connect to Loftware Web Services.
3. Stop Tomcat.
4. Delete the following WAR files from the `$tomcat\webapps` folder.
 - `lwaservices.war`
 - `loftwareadmin.war`
5. Delete the following application folders from the `$tomcat\webapps` directory.
 - `lwaservices`
 - `loftwareadmin`
6. Delete the `$tomcat\work\Catalina\localhost\loftwareadmin` folder.
7. Run `LWSInstall.bat` or `LWSInstall.sh`. See "Run the LWSinstall Script" on page 19.

This page intentionally left blank

Section 3: Installing Loftware Web Services Components Individually

You can install each component of Loftware Web Services on a different computer. Use the instructions in this section if your Loftware Web Services configuration requires installation on separate computers. For installations on Windows computers see "About installing Loftware Web Services on Windows" on page 24 For installations on Unix/Linux computers see "About installing Loftware Web Services on Unix/Linux" on page 23.

About setting up the RBAC database using PostgreSQL

This section describes the steps that the database administrator performs to set up the RBAC database in PostgreSQL.

What you need

You must install the following on the RBAC database system.

- PostgreSQL 8.3.3
- A copy of the Loftware security schema `loftrbac_schema.sql`
- RBAC Database user role

Set up the RBAC Database

Note: For complete instructions on creating databases and roles in PostgreSQL, see the documentation installed with PostgreSQL.

1. Start the PostgreSQL pgAdmin III tool.
2. Connect to your PostgreSQL Database Server.
3. Create a database called "loftrbac"
4. Create a role (user) called "loftwaredbuser" with password "ldbu\$pwd"
5. Open command prompt or shell, and navigate to `loftrbac_schema_pg.sql`.
6. Run `postgresql_install_directory\8.3\bin\psql -U postgres loftrbac < loftrbac_schema_pg.sql`

About installing Software Web Services on Unix/Linux

To install a Software Web Services environment on a Unix/Linux server you extract the installation files, prepare the Web Application Environment, and then deploy the WAR files in the application server.

Extract the installation files in Unix

Before deploying Software Web Services in Unix, you must prepare the installation files.

1. Copy and expand the `lwaservices.tar.gz` file to a temporary folder on the server where you want Software Web Services to reside.
2. Copy and expand the `loftwareadmin.tar.gz` file to a temporary folder on the server where you want Software Web Services Administration to reside. The extracted files should include `loftwareadmin.properties` and `loftwareadmin.war`.

Preparing the Web Application Environment in Unix

Before deploying Software Web Services in Unix, you must prepare the web application environment on each of the servers where a Software Web Services component will reside:

1. Under the `$tomcat` folder, create a folder called `Software`. Under the `Software` folder, create a "conf" folder and a "logs" folder.
2. Configure the `SOFTWARE_HOME` Environmental variable.
 - a. Edit `$tomcat/conf/context.xml`.
 - b. Between the `<context></context>` elements, add the following tag where `SOFTWARE_PATH` is the path to the `Software` directory created in step 1:


```
<Environment name="SOFTWARE_HOME" value="SOFTWARE_PATH" type="java.lang.String" override="false"/>
```

Deploy Software Web Services on Unix/Linux

1. Stop Tomcat.
2. Run the `lwaservicesinstall.sh` script with `$tomcat` and `axis2` as parameters.

For example

Open a shell prompt in Unix/Linux and enter:

```
./lwaservicesinstall.sh $tomcat $tomcat/webapps/axis2
```

3. Restart Tomcat.

Note: Apache Tomcat will automatically expand the war file under the `lwaservices` directory structure when you start Tomcat.

The Software Web, Cache, and Security services are installed on a single Tomcat instance. The installation process also creates security keys. See "Recreating the Public and Private Key Files" on page 39.

Deploy Software Web Services Administration on Unix

1. Stop Tomcat.
2. Copy the `loftwareadmin.properties` file to `SOFTWARE_HOME/conf`.
3. Copy the `loftwareadmin.war` file to the `$tomcat/webapps` directory.
4. Restart Tomcat.

Note: Apache Tomcat will automatically expand the war file under the `loftwareadmin` directory structure when you start Tomcat.

About installing Software Web Services on Windows

To install the Software Web Services environments on a Windows server you extract the installation files, prepare the Web Application Environment, and then deploy the WAR files in the application server.

Extract the installation files in Windows

Before deploying Software Web Services in Windows, you must prepare the installation files:

1. Copy and unzip the `lwaservices.zip` file to a temporary folder on the server where you want Software Web Services to reside.
2. Copy and unzip the `loftwareadmin.zip` file to a temporary folder on the server where you want Software Administration to reside. The extracted files should include `loftwareadmin.properties` and `loftwareadmin.war`.

Preparing the Web Application Environment in Windows

Before deploying Software Web Services in Windows, you must prepare the web application environment on each of the servers where a Software Web Services component will reside:

1. Under the `$tomcat` folder, create a folder called `Software`. Under the `Software` folder, create a `conf` folder and a `logs` folder.
2. Configure the `SOFTWARE_HOME` Environmental variable.
 - a. Edit `$tomcat/conf/context.xml`.
 - b. Between the `<context></context>` elements, add the following tag:


```
<Environment name="SOFTWARE_HOME" value="SOFTWARE_PATH" type="java.lang.String" override="false"/>
```

 where `SOFTWARE_PATH` is the path to the `Software` directory created in step 2 above.

Deploy Software Web Services on Windows

1. Stop Tomcat.
2. Run the LWSInstall.bat script with *\$tomcat* and your *axis2* folder as parameters.

For Example

Open a command window in Windows, and enter:

```
C:\LWATemp\SoftwareWebAccess\WebServices>lwaservicesinstall.bat "$tomcat" "$tomcat\webapps\axis2"
```

Note: Put quotation marks around the *\$tomcat* parameter and the full path to the axis2 install folder.

3. Restart Tomcat.

Note: Apache Tomcat will automatically expand the war file under the lwaservices directory structure when you start Tomcat.

The Software Web, Cache, and Security services are installed on a single Tomcat instance. The installation process also creates Security keys. See "Recreating the Public and Private Key Files" on page 39.

Deploy Software Web Services Administration on Windows

1. Stop Tomcat.
2. Copy the loftwareadmin.properties file to *SOFTWARE_HOME*\conf.
3. Copy the loftwareadmin.war file to the *\$tomcat*\webapps directory.
4. Restart Tomcat.

Note: Apache Tomcat will automatically expand the war file under the loftwareadmin directory structure when you start Tomcat.

Section 4: Uninstalling Web Services

You can uninstall Loftware Web Services from the Web Server by removing the WAR files and the associated web application folders and files.

Uninstall Loftware Web Services

Use the following instructions to remove Loftware Web Services from a Web Server.

1. Stop the Tomcat web server.
2. Delete the `lwaservices.war` file from the `$tomcat\webapps` directory.
3. Delete the `lwaservices` directory from the `$tomcat\webapps` directory.
4. Delete the `lwaservices` directory structure from the `$tomcat\work\Catalina\localhost` directory.
5. Delete the following files from the `$tomcat\webapps\axis2\WEB-INF\services` directory.
 - `LoftwareAdminWS_I18N.properties`
 - `LoftwarePrintWS_I18N.properties`
 - `LoftwareWebServices.properties`
 - `LoftwareWebServices.aar`
6. Delete the following files from the `$tomcat\webapps\axis2\WEB-INF\classes` directory.
 - `LDAPConnectionTemplates.xml`
 - `security_settings.xml`
7. Delete the following files from the `$tomcat\webapps\axis2\WEB-INF\lib` directory.
 - `ClientCommWDProtocol.jar`
 - `loftwareCommon.jar`
 - `loftwareSecurity.jar`
 - `loftwareSecurityCommon.jar`
8. Delete the following files from the `$tomcat\bin` directory.

- auditrecord.xsd
 - auditrecorddetail.xsd
 - auditrecordsearchcontext.xsd
 - CreateKeys.bat
 - CreateKeys.sh
 - filter.xsd
 - group.xsd
 - key.private.xml
 - key.public.xml
 - LDAPConfigurationTool.bat
 - LDAPConfigurationTool.properties
 - LDAPConfigurationTool.sh
 - opensaml.properties
 - role.xsd
 - rule.xsd
 - securityAuthentication.config
 - sort.xsd
 - user.xsd
 - serviceProvider.publicKeys.xml
9. Refer to the “Configure Tomcat Settings” section of the installation guide and remove the specified configuration settings from either the Tomcat Configuration UI or catalina.bat.

Uninstall Software Web Services Administration

Use the following instructions to remove Software Web Services Administration from a Web Server.

1. Stop the Tomcat Web Server.
2. Delete the “loftwareadmin.war” file from the *\$tomcat*\webapps directory.
3. Delete the “loftwareadmin” directory structure from the *\$tomcat*\webapps directory.
4. Delete the “loftwareadmin” directory structure from the *\$tomcat*\work\Catalina\localhost directory.
5. Delete the “loftwareadmin.properties” file from the *\$tomcat*\Software\conf directory.
6. If the Software Print Web application is not installed on the same server, delete the Software directory from *\$tomcat*.

Section 5: Configuring Authentication

You can configure Loftware Web Services to work with your existing security domain using Lightweight Directory Access Protocol (LDAP).

To configure LDAP with Loftware Web Services, you complete the following steps:

- Create the default LDAPConfigurationTool.properties file
- Select LDAP Authentication Mode
- Generate encrypted passwords
- Update the LDAPConfigurationTool.properties file
- Create the LDAP Configuration Files
- Update the security_settings.xml file
- Verify the SecurityAuthentication.config file
- Update the LDAPConnectionTemplate.xml file
- Update User Interface Properties files

About using the LDAPConfigurationTool to configure LDAP

You can use the LDAPConfigurationTool to configure the Loftware Web Services connection to an LDAP compliant directory service. You run the LDAPConfigurationTool from a command line.

You can use the LDAPConfigurationTool for the following:

- Generate encrypted password strings.
- Verify the ability to authenticate a user.
- Create the securityAuthentication.config file.
- Create the configuration settings for the security_settings.xml file.
- Create the LDAPConnectionTemplate.xml file.

Parameters

You use parameters with the LDAPConfigurationTool to accomplish different tasks. You can apply the following parameters when you run the LDAPConfigurationTool.

- -h - Displays help for the tool
- -c - Creates the default LDAPConfigurationTool.properties file
- -e, *string* - Creates an encrypted string for the passwords that must be included in Software Web Services configuration files.

LDAPConfigurationTool.properties

To use the LDAPConfigurationTool to configure LDAP, you first create the LDAPConfigurationTool.properties file. The LDAPConfigurationTool uses the LDAPConfigurationTool.properties file to get LDAP settings. Before running the LDAPConfigurationTool script to generate configuration files, you edit the settings in LDAPConfigurationTool.properties.

Note: The LDAPConfigurationTool.properties file contains detailed instructions and explanations of the test and LDAP settings that the properties file controls. Read these instructions before changing any settings or using the LDAPConfigurationTool.

Create the default LDAPConfigurationTool.properties file

The first time you run the LDAPConfigurationTool, you must create the LDAPConfigurationTool.properties file.

1. Open a command line and navigate to the directory containing the LDAPConfigurationTool script (.bat in Windows, .sh in Unix).
2. Run the tool with the -c parameter. The default LDAPConfigurationTool.properties file is created in the Tomcat bin directory.

For example

```
C:\Program Files\Apache Software Foundation\Tomcat 6.0\bin>
LDAPConfigurationTool -c
```

About LDAP Authentication Modes

An Authentication Mode is the method that Software Web Services should use to connect to a LDAP directory service. Software Web Services supports the SearchFirst and AuthenticationOnly authentication modes. To configure Software Web Services to use your LDAP directory service, you must choose an authentication mode.

Note: For more information on LDAP authentication see the LdapLoginModule documentation at <http://java.sun.com>.

SearchFirst

This mode searches your LDAP directory to determine a user's distinguished name. Use this mode if you do not know a user's distinguished name at login.

To use SearchFirst mode with Loftware Web Services, you must supply a domain account's credentials. These credentials are used to query your LDAP using a search filter and return each user's distinguished name when they log in to Loftware Installation Guide

Note: Loftware recommends using an account with a password that does not expire. For example, the account used for the Loftware Print Server service. This prevents authentication failure when the password expires.

When you use SearchFirst Authentication Mode, you must supply domain specific information in the following files:

- security_settings.xml
- securityAuthentication.config
- loftwareadmin.properties

AuthenticationOnly

This mode attempts to authenticate using the supplied username and password. Use this mode if you know a user's distinguished name at login or if there are few distinguished name templates to check.

To use AuthenticationOnly mode with Loftware Web Services, you must create a distinguished name template to use in processing login requests. When you use AuthenticationOnly Authentication Mode, you must supply domain specific information in the following files:

- security_settings.xml
- securityAuthentication.config
- LDAPConnectionTemplate.xml
- loftwareadmin.properties

Select LDAP Authentication Mode

You choose an authentication mode by specifying which mode you want to use in the LDAPConfigurationTool.properties file.

Note: You must create the LDAPConfigurationTool.properties file before specifying an authentication mode. See "Create the default LDAPConfigurationTool.properties file" on page 29.

Use SearchFirst Mode

1. Open the LDAPConfigurationTool.properties file.
2. Uncomment the searchFirst AuthenticationMode setting.

```
AuthenticationMode searchFirst
#AuthenticationMode authenticationOnly
```

3. Follow the instructions in the LDAPConfigurationTool.properties file for setting up the Administrator DN.

- Use the LDAPConfigurationTool with the *-e string* parameter to generate an encrypted password for the AdministratorPassword and the TestUserPassword setting

Use AuthenticationOnly mode

1. Open the LDAPConfigurationTool.properties file.
2. Uncomment the AuthenticationOnly AuthenticationMode setting.

```
#AuthenticationMode searchFirst
AuthenticationMode authenticationOnly
```

3. Follow the instructions in the LDAPConfigurationTool.properties file for setting up a DN template.
4. Optionally, enter a name for the XML file that will store the DN template for the templateConfigFileName.
5. Enter your LDAP settings for the different parts of the distinguishedNameTemplate setting.

Generate encrypted passwords

After creating the default LDAPConfigurationTool.properties file, you must create encrypted password strings to include in the properties file.

1. Open a command line, and navigate to the location of the LDAPConfigurationTool script.
2. Run the LDAPConfigurationTool script with the *-e* parameter, and the test user's password. An encrypted string appears in the command window.

For example

Open a command prompt in Windows, and enter:

```
C:\Program Files\Apache Software Foundation\Tomcat 6.0\bin>
LDAPConfigurationTool -e password
```

3. Open the LDAPConfigurationTool.properties file.
4. Copy the string from the command line, and paste it over the default TestUserPassword property in the LDAPConfigurationTool.properties file.
5. Repeat steps 2-3 with the administrator password, and paste the encrypted string over the default AdministratorPassword property in the LDAPConfigurationTool.properties file.
6. Save the LDAPConfigurationTool.properties file.

Update the LDAPConfigurationTool.properties file

Note: The LDAPConfigurationTool.properties file contains detailed instructions and explanations of the test and LDAP settings that the properties file controls. Read these instructions before changing any settings or using the LDAPConfigurationTool.

1. Open the LDAPConfigurationTool.properties file in a text editor such as Notepad or VI.
2. Complete the parameters in the Test user information section with the credentials of a known user.
 - Use the LDAPConfigurationTool with the `-e, string` parameter to generate an encrypted password for the TestUserPassword setting.
3. Indicate whether or not to generate a new securityAuthentication.config file. If you set to `false`, there must be an existing securityAuthentication.config file.
4. Provide Directory Service settings.
5. Save the LDAPConfigurationTool.properties file.

Create the LDAP Configuration Files

1. Open a command prompt and navigate to the location of the LDAPConfigurationTool.
2. Run the LDAPConfigurationTool script with no parameters.

For example

```
C:\Program Files\Apache Software Foundation\Tomcat 6.0\bin> LDAPConfigurationTool
```

The LDAPConfigurationTool performs the following, using the current setting from the LDAPConfigurationTool.properties file:

- Attempts to authentication the test user. The results displayed indicate if the user was authenticated.
- If you set the `useExistingFile` setting to `false`, creates a new SecurityAuthentication.config file.
- Creates a TestConfigSettings.xml file.
- If you are using AuthenticationOnly mode, creates the LDAPConnectionTemplate.xml file.

Update the security_settings.xml file

1. Open the TestConfigSettings.xml file created when you ran the LDAPConfigurationTool.
2. Copy the text between the `<properties>` tags.

Note: Variable values appear in this example. **DO NOT** copy and paste the text from this example directly to your file. Change the variables to match your configuration before using this example text.

For example

If you are using SearchFirst mode, the entry will look like the following:

```
<entry key="yourdomain.com.UseLDAPSearch">true</entry>
<entry key="yourdomain.com
.LDAPAdminPassword">AXNbK9HSsQrWPY9KRiN/5Q==</entry>
```

If you are using AuthenticationOnly mode, the entry will look like the following:

```
<entry key="yourdomain.com.UseLDAPSearch">>false</entry>
<entry
key="LDAPconnectionTemplateFile">LDAPConnectionTemplates.xml</entry>
```

3. Open the `$tomcat\webapps\axis2\WEB-INF\classes\security_settings.xml` file.
4. Paste the copied setting from the `TestConfigSettings.xml` file at the end of the `security_settings.xml` file before the `</properties>` tag. Save the file.

Verify the securityAuthentication.config file

Each time you run the `LDAPConfigurationTool`, a file called `securityAuthentication.config` that contains domain information is created.

1. Verify the domain information in the `securityAuthentication.config` file.

For example

Each renamed `.config` file will contain an entry like the following.

```
yourdomain.com {
com.sun.security.auth.module.LdapLoginModule required
    debug = true
    storePass=true
    usefirstPass=true
    useSSL=false
    authIdentity="{USERNAME}"
    userProvider="ldap://ldapserver";
};
```

2. Save the `securityAuthentication.config` file.

TIP: If you rename the `securityAuthentication.config` file after creation, change the settings in `LDAPConfigurationTool.properties`, and run `LDAPConfigurationTool` again, you can create additional domain descriptions to later combine in a single `securityAuthentication.config` file.

Update the LDAPConnectionTemplate.xml file

If you are using `AuthenticationOnly` mode to connect to an LDAP directory service, each time you run the `LDAPConfigurationTool` a file called `LDAPConnectionTemplate.xml` that contains your domain's DN template is created.

1. Verify the domain information in the `LDAPConnectionTemplate.xml` file.

For example

```
<entry key="yourdomain.com">CN=-givenname- -surname-
,CN=Users,DC=user,DC=com</entry>
<entry key="default">-username-</entry>
```

2. Save the `LDAPConnectionTemplate.xml` file.
3. Copy the `LDAPConnectionTemplate.xml` file to `$tomcat\webapps\axis2\WEB-INF\classes`.

Verify the securityAuthentication.config file

TIP: If you change the settings in LDAPConfigurationTool.properties, and run LDAPConfigurationTool again; you can create additional domain templates to later add to the LDAPConnectionTemplate.xml file you copied to **\$tomcat**\webapps\axis2\WEB-INF\classes.

This page intentionally left blank

Section 6: Configuring Loftware Web Services

After installing Loftware Web Services, you must configure it for your particular environment.

The following configuration and properties files are used by Loftware Web Services. For changes in these files to take effect, you must restart the application server after making changes.

Logging

Log4j.properties

This file stores the logging settings used by LoftwarePrintWS and LoftwareAdminWS. See "About Configuring Logging in Loftware Web Services" on page 37

Log4j.xml

This file stores the logging settings used by the Cache Service and Security Service. See "About Configuring Logging in Loftware Web Services" on page 37

Authentication

security_settings.xml

Configuration settings for authentication and authorization. See "Security_settings.xml" on page 41

securityAuthentication.config

Configuration file for setting up the Authentication provider information for JAAS. See "securityAuthentication.config" on page 45

key.private.xml

The private key file for the Authentication component. See "Recreating the Public and Private Key Files" on page 39

key.public.xml

The public key file for the Authentication component.

serviceProvider.publickeys.xml

The public key configuration file for the trusted authentication sources. (This file contains the information in key.public.xml plus any other authentication provider IDs from other systems.)

LDAPConnectionTemplates.xml

LDAP connection definitions for your security domains. See "LDAPConnectionTemplates.xml" on page 45

Cache Service

SocketConfig.properties

The properties file that defines the connection to the cache service. See "SocketConfig.properties" on page 46

cache_system.properties

The properties file that defines the cache service. See "Cache_system.properties" on page 47

Web Services

LoftwareWebServices.properties

The properties file that defines the connection between Loftware WebAccess and an LPS Print Server. See "Loftware Web Services Properties" on page 40

Administration

loftwareadmin.properties

The properties file that defines certain attributes in the Administration client. See "Loftware Admin Properties" on page 48

About Configuring Logging in Loftware Web Services

Loftware Web Services uses the Log4J logging utility. Log4j is an Apache Software Foundation tool primarily used for logging and debugging. Refer to <http://logging.apache.org/log4j/1.2/index.html> for more information. Log4j.properties is the property file containing the logging info.

The Loftware Web Services logging system supports multiple logging types including: Informational, Debug, Warnings, and Errors.

For Loftware Web Services running under the Axis2 framework, log files are created in *\$tomcat* /Loftware/logs. The Web Services log file is called Loftware WebServices.log.

Configure Log4j.properties

1. Open the `$tomcat\webapps\axis2\WEB-INF\classes\log4j.properties` file.
2. Add the following to the bottom of the `log4j.properties` file:

Note: If you copy and paste the text in this example directly into a file from a PDF, you may need to adjust the line breaks in the pasted text. Each `log4j.appender...` setting should be one line in a file. There should be no line breaks (carriage returns) in the text that follows an "=" sign. Start each `log4j.appender...` setting on a new line.

```
#LOFTWARE_LOGFILE appender for Loftware Web Services
log4j.logger.com.loftware=WARN, LOFTWARE_LOGFILE
log4j.appender.LOFTWARE_LOGFILE=org.apache.log4j.RollingFileAppender
log4j.appender.LOFTWARE_LOGFILE.MaxFileSize=20MB
log4j.appender.LOFTWARE_LOGFILE.MaxBackupIndex=20
log4j.appender.LOFTWARE_
LOGFILE.File=${catalina.home}/Loftware/Logs/LoftwareWebServices.log
log4j.appender.LOFTWARE_LOGFILE.Append=true
log4j.appender.LOFTWARE_
LOGFILE.layout=org.apache.log4j.PatternLayout
log4j.appender.LOFTWARE_LOGFILE.layout.ConversionPattern=%d{yyyy-MM-
dd HH:mm:ss.SSSS}|%5p|%t|%c %x|%m%n
```

Log4j.xml

The following is an example of a `log4j.xml` file. `log4j.xml` is deployed to `$tomcat/webapps/lwaservices/WEB-INF/classes`.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE log4j:configuration SYSTEM "log4j.dtd">

<log4j:configuration xmlns:log4j="http://jakarta.apache.org/log4j/">

  <appender name="CONSOLE" class="org.apache.log4j.ConsoleAppender">

    <layout class="org.apache.log4j.PatternLayout">
      <param name="ConversionPattern"
        value="%p [%t] %c{1}.%M(%L) | %m%n" />
    </layout>

  </appender>

  <appender name="LOGFILE" class="org.apache.log4j.FileAppender">
    <param name="File"
      value="${catalina.home}/Loftware/logs/LoftwareServices.log" />
    <param name="Append" value="true" />
    <layout class="org.apache.log4j.PatternLayout">
      <param name="ConversionPattern"
        value="%d{dd MMM yyyy HH:mm:ss,SSSS} %p [%t] %c{1}.%M(%L) | %m%n" />
    </layout>
  </appender>

  <logger name="org.apache">
    <level value="WARN" />
  </logger>
```

```

<logger name="com.loftware">
<level value="ERROR" />
</logger>

<root>
<level value="WARN" />
<appender-ref ref="CONSOLE" />
<appender-ref ref="LOGFILE" />
</root>

</log4j:configuration>

```

Create certificate keystore

1. Create the SSL Keystore by executing the following command, where `%JAVA_HOME%` is the root folder of the JDK1.6 installation:

```
%JAVA_HOME%\bin\keytool -genkey -alias loftware -keyalg RSA -
keystore $tomcat/conf/loftware.keystore
```

This command will initiate the following prompt/response sequence.

Prompt	Response
Enter keystore password:	loftware
Re-enter new password:	loftware
What is your first and last name? [Unknown]:	Loftware Inc
What is the name of your organizational unit? [Unknown]:	Loftware Inc
What is the name of your organization? [Unknown]:	Loftware Inc
What is the name of your City or Locality? [Unknown]:	Portsmouth
What is the name of your State or Province? [Unknown]:	New Hampshire
What is the two-letter country code for this unit? [Unknown]:	US
Is CN=Loftware Inc, OU=Loftware Inc, O=Loftware Inc, L=Portsmouth, ST=New Hampshire, C=US correct? [no]	yes

2. Proceed to *Enable SSL in Tomcat*.

Recreating the Public and Private Key Files

When you install Software Web Services, the install script automatically creates the public and private key files. Use the following process if you need update the files. For example, if you need to change the name of the computer on which Software Web Services is installed.

1. Locate the CreateKeys.bat script. (This file was copied to `$tomcat\bin` during installation of Software Web Services.)

Note: This script assumes that the jar files are in `$tomcat\webapps\axis2\WEB-INF\lib\`. You may need to modify the classpath setting in the script to match your environment.

2. Run the bat file. The following files should be created in `$tomcat\bin`.
 - `key.public.xml`
 - `key.private.xml`
 - `serviceProvider.publicKeys.xml`

Key.public.xml

This is the public key for the Authentication provider. This key should be distributed to the Authorization provider for which it will be used in conjunction with. The information in this file is copied to the `ServiceProvider.publicKeys.xml` file.

Key.private.xml

This is the private key for the Authentication provider. This key provides the private key information for signing secure tokens issued by the Authentication service. This file should **NOT** be distributed. It should be kept secured against all users except the administrator.

ServiceProvider.publickeys.xml

The information in this file is used by the Authorization component of security. This file provides the list of trusted Authentication providers. The public key is used to verify that the security token is from a trusted provider and that it is valid.

Customizing Software Web Services

You can customize Software Web Services by making changes to the `SoftwareWebServices.properties` file.

Software Web Services Properties

The Software Web Services contains a properties file that is read in when the service starts up. The information is cached and used for each call to LPS. `SoftwareWebServices.properties` is deployed to `$tomcat\webapps\axis2\WEB-INF\services`. You can configure the following settings:

- `LPS_ADDRESS` – The address of the LPS install.
- `LPS_PORT` – The port of the LPS install
- `LPS_SOCKET_TIMEOUT` – The amount of time, in milliseconds, before the socket will time out and close the connection.
- `LPS_REQUEST_TIMEOUT` – The amount of time, in milliseconds, Software Web Access will wait for LPS to respond before timing out.

- LPS_REQUEST_SPIN – The number of times to immediately recheck for data before waiting. The higher the value the more attempts will be made.
- LPS_REQUEST_INTERVAL – The amount of time, in milliseconds, to wait before rechecking for data.
- WS_KEEPLIVE_INTERVAL – How often, in seconds, a signal is sent to LPS to retain the web service's license.
- WS_LICENSERETRY_INTERVAL – How frequently, in seconds, the web service should retry if no license is given from LPS.
- WS_FORCESHUTDOWN_WAITTIME – How soon, in seconds, connection threads should be forced to shutdown if they are not closed by the webservice.
- db.jdbc.url - The address of the RBAC database install.

LoftwareWebServices.properties contains the following default information:

Note: Variable values appear in this example. **DO NOT** copy and paste the text from this example directly to your file. Change the variables to match your configuration before using this example text.

Example LoftwareWebServices.properties file

```
#LoftwareWebServices properties
LPS_ADDRESS=127.0.0.1
LPS_PORT=2723
LPS_SOCKET_TIMEOUT=5000
LPS_REQUEST_TIMEOUT=5000
LPS_REQUEST_SPIN=0
LPS_REQUEST_INTERVAL=25

#Web Service Licensing
WS_KEEPLIVE_INTERVAL=30
WS_LICENSERETRY_INTERVAL=30
WS_FORCESHUTDOWN_WAITTIME=60

#Auditing properties
db.jdbc.driver=org.postgresql.Driver
db.jdbc.url=jdbc:posrtgresql://rbacIPAddress:5432/loftbac
db.jdbc.pool=true
```

Security_settings.xml

The security_setting.xml file includes properties for specifying settings for the security and application databases. The security_settings.xml file is located in the *\$tomcat* \webapps\axis2\WEB-INF\classes folder and the *\$tomcat*\webapps\lwaservices\WEB-INF\classes. You can generate certain settings for this file using the LDAPConfigurationTool.

Note: All modifications to settings in security_settings.xml must be within the <properties> tag.

Setting	Description
AuthorizationService	This service interacts with the cache service and

Setting	Description
	provides requested security authorization information.
BindAddress	The address where the service is running.
Port	The port to bind to.
ServerTimeout	How often to allow the server to check for shutdown commands
ClientThreadCount	How many clients threads to spin up. This is a performance setting and is tied directly to the socket backlog.
ClientThreadClass	Name of the class that implements the processing behavior for the connecting clients

RulesDatabase Setting	Description
RulesCaching	Sets whether the rules will be pulled from the cache service or the database.

Rule Engine Setting	Description
RuleEngineProviderUri	URI to the rules engine provider.
RulesDatabaseDSLFile	Domain Specific Language file location.

Authentication parameters Settings	Description
PublicKeysFile	The file containing the list of public keys from trusted providers.
LocalLoginType	The tag name of the local provider as displayed in the JAAS configuration file (securityAuthentication.config file).
LDAPConnectionTemplateFile	The file containing the LDAP connection configuration settings.
PublicKeyFile	The file containing the public key for authentication.
PrivateKeyFile	The file containing the private key for authentication.
SecureTokenExpirationMs	The duration of time for the issued token to be valid.
db.jdbc.url	The connection string of the authentication database.
db.jdbc.username	The username used to login. This setting is optional.
db.jdbc.password	The password to login. This setting is optional.
db.jdbc.driver	The driver instance to use for the connection.

Note: These settings should be the same as the settings in the securityAuthentication.config file.

Note: Variable values appear in this example. **DO NOT** copy and paste the text from this example directly to your file. Change the variables to match your configuration before using this example text.

Example security_settings.xml file

```

<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<!DOCTYPE properties (View Source for full doctype...)>
<properties version="1.0">
  <!-- Authorization -->
  <!-- Security Service and Client Connection information -->
  <entry
key="AuthorizationService.BindAddress">192.168.10.87</entry>
  <!-- local address for which the service should bind and the
address
      for the client to connect -->

  <entry key="AuthorizationService.Port">2725</entry>
  <!-- port that the security service is listening on, and the port
      for the client to connect -->

  <entry key="AuthorizationService.ServerTimeout">3000</entry>
  <!-- Socket timeout for listening for connections -->

  <entry key="AuthorizationService.ClientThreadCount">5</entry>
  <!-- amount of client connections to handle at once -->

  <entry key="AuthorizationService.ClientThreadClass">
com.software.security.service.thread.SecurityRequestThread</entry>
  <!-- class to use for handling connection requests -->

  <!-- Rules engine wrapper usage for figuring out what type of
rules engine to use. -->
  <entry key="RuleEngineProvider">org.drools.jsr94.rules.
RuleServiceProviderImpl</entry>

  <!-- The rule engine provider implementation -->

  <entry key="RuleEngineProviderUri">http://drools.org/</entry>
  <!-- The rule engine provider uri -->

  <!-- Rules engine Domain Specific Language file to use as a
resource -->
  <entry key="RulesDatabaseDSLFile">LoftwareStandard.dsl</entry>

  <!-- Security Service - determines how to retrieve the rules
information -->
  <entry key="RulesCaching">true</entry>
  <!--RulesDatabaseReader - configuration for reading rules from the
database if
      RulesCaching is false -->
  <!-- <entry
key="RulesDatabase">jdbc:postgresql://localhost:3306/loftrbac</entry>
  -->
  <!-- The driver to use for connecting to the database --> -
  <!-- <entry key="RulesDatabaseLogin">engineer</entry> -->
  <!-- The database connection string -->
  <!-- <entry key="RulesDatabasePassword">engloftware</entry> -->

```

```

<!-- The database account user name -->
<!-- <entry
key="RulesDatabaseDriver">org.postgresql.Driver</entry> -->
<!-- The database account password -->
<!-- information for determining where to get the public keys for
signature verification -->

<entry key="PublicKeysFile">ServiceProvider.publickeys.xml</entry>
<!-- The public key file store for consumers i.e. security service
-->
<!-- information for determining where to get the public keys for
signature verification -->

<entry key="RoleAccessControlProvider">
  com.software.security.authentication.RBACAccessControl</entry>
<!-- The public key file store for consumers i.e. security service
-->
<!-- Authentication -->
<!-- information for determining where to get the public and
private keys
for signing -->

<entry key="PublicKeyFile">key.public.xml</entry>
<!-- The public key file for the authentication service -->

<entry key="PrivateKeyFile">key.private.xml</entry>
<!-- The private key file for the authentication service -->
<!-- JAAS configurations to use for authentication of local and
system users -->

<entry key="LocalLoginType">CustomLoftRBAC</entry>
<!-- The JAAS configuration to use for checking the user locally -
->
<!-- n/a uses domain name now.<entry
key="SystemLoginType">LDAPSample</entry> -->
<!-- The JAAS configuration to use for checking the user on a
system i.e. LDAP -->

<entry
key="LDAPConnectionTemplateFile">LDAPConnectionTemplates.xml</entry>
<!-- The file holding the LDAP DN/search templates -->
<!-- Setting for specifying how long the security token should be
valid -->

<entry key="SecureTokenExpirationMs">3600000</entry>
<!-- RBAC connection information -->

<entry key="db.jdbc.driver">org.postgresql.jdbc.Driver</entry>
<!-- The driver to use for connecting to the database -->

<entry
key="db.jdbc.url">jdbc:postgresql://localhost:3306/loftfbac</entry>
<!-- The database connection string -->

<entry key="db.jdbc.username">engineer</entry>
<!-- The database account user name -->

```

```
<entry key="db.jdbc.password">engloftware</entry>
<!-- The database account password -->
</properties>
```

securityAuthentication.config

This configuration file is required by the Java Authentication and Authorization Service (JAAS) for configuring the proper login module to use for authentication. The LDAP component needs to match the local directory service information. This file can be updated using the LDAPConfigurationTool.

Example SecurityAuthentication.config file

```
/** Domain configuration settings */
/** Contains the local authentication provider as well as */
/** the various domain configurations. */

CustomLoftRBAC {
    com.loftware.security.authentication.LoftwareRBACLoginModule
    required debug=true
    failOnPassword="false";
};

yourdomain.com {
    com.sun.security.auth.module.LdapLoginModule required
        debug = true
        storePass=true
        usefirstPass=true
        useSSL=false
        authIdentity="{USERNAME}"
        userProvider="ldap://ldapserver";
};

yourdomain2.com {
    com.sun.security.auth.module.LdapLoginModule required
        debug = true
        storePass=true
        usefirstPass=true
        useSSL=false
        authIdentity="{USERNAME}"
        userProvider="ldap://ldapserver2";
};
```

LDAPConnectionTemplates.xml

If you are using AuthenticationOnly mode to connect to your LDAP directory service, you need to specify a distinguished name template in the LDAPConnectionTemplates.xml file. LDAPConnectionTemplates.xml is deployed to *\$tomcat/webapps/axis2/WEB-INF/classes*. The following tokens can be used to configure your domains:

LDAP Token	RBAC User table value
-username-	Username
-domain-	Domain
-givenname-	First_name
-surname-	Last_name
-email-	Email

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE properties SYSTEM
"http://java.sun.com/dtd/properties.dtd">
<properties>
  <entry key="yourdomain.com">CN=-givenname- -surname-
,CN=Users,DC=yourdomain,DC=com</entry>
  <entry key="default">-username-</entry>
</properties>
```

Configuring the Cache Service in Software Web Services

The primary function of the Software Print Server (LPS) is to process print requests. It also returns a list of printers or labels to On Demand Print. These lists are large and retrieving them from LPS could impact system performance.

Software Web Services includes a cache service to store printer and device information. This information is retrieved from LPS at regular intervals. The cache service improves performance and offloads common LPS requests from On Demand Print.

SocketConfig.properties

The settings in SocketConfig.properties define the connection information to connect to the Cache service. The information in the SocketConfig.properties file must match the information in the security_settings.xml file located in the *\$tomcat\webapps\lwaservices\WEB-INF\classes* folder.

The SocketConfig.properties file is located in the *\$tomcat\webapps\lwaservices\WEB-INF\classes* folder.

Example SocketConfig.properties file

```
## These settings are being used by the CacheManager System and
Cache Client
CacheManagerSystem.BindAddress=127.0.0.1
CacheManagerSystem.Port=2726

## These settings are being used by the CacheManager System only
CacheManagerSystem.ServerTimeout=3000
CacheManagerSystem.ClientThreadCount=5
Cache-
Man-
agerSystem.ClientThreadClass=com.software.caching.CacheManagerRequestThread
```

Cache_system.properties

The Software Web Services cache service contains a properties file for configuration of the service. This file is deployed to *\$tomcat*\webapps\lwaservices\WEB-INF\classes. You can configure the following settings:

- LpsCacheSource.PollingInterval – How often, in minutes, the cache is refreshed from LPS.
- LpsCacheSource.LpsAddress – The address of the LPS install.
- LpsCacheSource.LpsPort – The port of the LPS install
- LpsCacheSource.LpsSocketTimeout – The amount of time, in milliseconds, before the socket will time out and close the connection.
- LpsCacheSource.LpsRequestTimeout – The amount of time, in milliseconds, Software Web Services will wait for LPS to respond before timing out.
- LpsCacheSource.LpsRequestSpin – The number of times to immediately recheck for data before waiting. The higher the value the more attempts will be made.
- LpsCacheSource.LpsRequestInterval – The amount of time, in milliseconds, to wait before rechecking for data.
- RbacCacheSource.PollingInterval - How often, in minutes, the cache is refreshed from LPS.
- RbacCachSource.db.jdbc.url - The address of the RBAC database install.

Cache_system.properties contains the following default information:

Note: Variable values appear in this example. **DO NOT** copy and paste the text from this example directly to your file. Change the variables to match your configuration before using this example text.

Example Cache_system.properties file

```
## Define cache sources (This section should not be modified)

CacheSource.RbacCacheSource=com.loftware.caching.RbacCacheSource
CacheSource.LpsCacheSource=com.loftware.caching.LpsCacheSource

## Define Data type mappings for cache sources (This section should
not be modified)

DataType.LpsCacheSource.Label=LABELS
DataType.LpsCacheSource.Printer=DEVICES
DataType.RbacCacheSource.Rule=RULES

## Define Data type mappings for client requests (This section
should not be modified)

DataType.GET_LABELS_REQ=LABELS
DataType.GET_DEVICES_REQ=DEVICES
DataType.GET_RULES_REQ=RULES

## LpsCacheSource properties

LpsCacheSource.PollingInterval=10
LpsCacheSource.LpsAddress=127.0.0.1
```

```
LpsCacheSource.LpsPort=2723
LpsCacheSource.LpsSocketTimeout=5000
LpsCacheSource.LpsRequestTimeout=5000
LpsCacheSource.LpsRequestSpin=0
LpsCacheSource.LpsRequestInterval=25

## RbacCacheSource properties

RbacCacheSource.PollingInterval=10
RbacCacheSource.db.jdbc.driver=org.postgresql.Driver
RbacCacheSource.db.jdbc.url=jdbc:postgresql://rbacIPAddress
:5432/loftrbac
RbacCacheSource.db.jdbc.pool=true
```

Configure the cache service

1. Open the `$tomcat\webapps\lwaservices\WEB-INF\classes\SocketConfig.properties` file.
2. Specify the BindAddress and Port that the cache system is running on. If you are connecting to the service on the local computer, specify the exact IP Address.

For example

```
CacheManagerSystem.BindAddress=localhost
CacheManagerSystem.Port=2726
```

3. Modify the configuration information in the `$tomcat\webapps\axis2\WEB-INF\classes\cache_system.properties` file. Set the LpsCacheSource and the JDBC connection strings.

For example

```
LpsCacheSource.LpsAddress=192.168.60.12
RbacCacheSource.db.jdbc.url=
jdbc:postgresql://192.168.60.12:3306/loftrbac
```

About Configuring Software WebAccess Administration

You can customize Software WebAccess Administration by making changes to the `loftwareadmin.properties` file.

Software Admin Properties

The `loftwareadmin.properties` file, located in `$tomcat/loftware/conf`, contains parameters that you can configure to customize the Software Web Services Administration web interface. (`$tomcat/loftware` is the folder defined in the `LOFTWARE_HOME` system environment variable.)

Example loftwareadmin.properties file

```
# This properties file is used to configure admin ui settings
com.lof-
tware.s-
ecurity.admin.service=http://localhost:8080/axis2/services/LoftwareAdminWS
```

```
# Domains to display for user logins - Add additional Domains using
comma delimited String
com.loftware.security.admin.domain=None

#####
# System Administrator contact phone number
#####
admin.contact.number=

#####
#
# Http Protocol Version Configuration
# Valid Settings: HTTP/1.0 or HTTP/1.1
# ADVANCED USERS ONLY
#
# Note: This settings affects the protocol
# version used in SOAP requests and has serious
# performance implications if changed
#
#####
com.loftware.security.admin.httpProtocolVersion=HTTP/1.0
```

Section 7: Accessing Loftware Web Services Administration

You can access the Loftware Web Services Administration component using the following browsers:

- Microsoft Internet Explorer version 7
- Mozilla Firefox version 2 or 3

Access Loftware Web Services Security Administration

Note: By default, the communication port assigned to the Loftware Web Services components is 8080, your system may be configured to use a different port.

You can access the Loftware Web Services Security Administration application by entering the URL, `http://<webserver-ipaddress:port>/loftwareadmin`. The Loftware Web Services Security Administration login page appears.

For detailed information on Loftware Web Services Security Administration see the *Loftware WebAccess Security Administration User's Guide* available in the Documents folder of your installation CD or download and from the Downloads section of www.loftware.com.

This page intentionally left blank