

Liferay Portal 4 - Installation Guide

Joseph Shum

Alexander Chow

Redmond Mar

Jorge Ferrer

Mark Wong

Charles May

Liferay Portal 4 - Installation Guide

Joseph Shum
Alexander Chow
Redmond Mar
Jorge Ferrer
Mark Wong
Charles May

1.4.3

Copyright © 2000, 2007 Liferay Inc.

Table of Contents

Preface	vi
1. Installation Options	1
2. Installing a Liferay Bundle	2
1. Quick Installation	2
2. Customizing the bundle installation	3
2.1. Database Configuration	3
2.2. Further Configuration	3
3. Detailed Installation Procedure	5
1. Application Servers	5
1.1. Geronimo 1.1 with Tomcat 5.0.28/5.5.17	5
1.2. JBoss 4.03sp1/4.04/4.05 with Jetty 5.1.1	7
1.3. JBoss 4.03sp1/4.04/4.05 with Tomcat 5.0.28/5.5.17	9
1.4. Jetty 5.1.1	12
1.5. Tomcat 5.0.28/5.5.17	15
1.6. Resin 3.0.19	17
1.7. Websphere 6.0.2.5	19
2. Databases	34
2.1. Oracle	34
4. Configuring Liferay Portal Paths	35
5. Integration with External Systems	36
1. Mail Servers	36
1.1. Washington IMAP+Sendmail	36
1.2. Cyrus IMAP+Postfix	40
1.3. Dovecot+Postfix	46
1.4. Microsoft Exchange	55
2. LDAP Integration	55
2.1. Installing Apache Directory Server	56
2.2. Installing LDAP Browser	56
2.3. Inputting User in LDAP Browser	56
2.4. Integration	58
3. Chat Portlet	60
4. CAS Server	61
5. Installation of Workflow services	64
5.1. Installation	64
5.2. Testing the installation	64
5.3. Configuration of the jBPM database	65
6. Alfresco	65
6. Multiple Portal Instances	67
7. Conclusion	68

List of Tables

3.1.	23
-----------	----

Preface

Intended audience. This installation guide for Liferay Portal 4 is still a work in progress and will be updated frequently with new content. Contributions are welcome. It explains from the simplest installation methods intended for people reviewing Liferay to the most elaborate ones that will serve those wanting to install Liferay in servers. All Liferay supported servers and databases are covered.

Liferay version. This guide has been written for Liferay 4.3. Some details might be slightly different for previous 4.x versions. Do not expect it to be accurate for even older versions.

Related documents. If this is not what you are looking for consider the following related documents

- Liferay Portal 4 - Customization Guide
- Liferay Portal 4 - Portal Administration Guide
- Liferay Portal 4 - Development in the Extension Environment

More information and support. If you are looking for help for a specific issue we invite you to use our community forums: <http://www.liferay.com/web/guest/devzone/forums> [<http://forums.liferay.com>] to ask your questions. We also offer professional support services (support@liferay.com [<mailto:support@liferay.com>]) where your company will be assigned a Liferay developer ensuring your questions are answered promptly so that your project is never compromised. Purchased support always gets first priority. This business model allows us to build a company that can contribute a great portal to the open source community. If your company uses Liferay, please consider purchasing support. Liferay has an extremely liberal license model (MIT, very similar to Apache and BSD), which means you can rebundle Liferay, rename it, and sell it under your name. We believe free means you can do whatever you want with it.

Chapter 1. Installation Options

Liferay Portal can be installed through different methods depending on your specific needs. The installation options can be grouped into three main types:

- *Using an open source bundle*: the bundles are the easiest and fastest installation method. They include an embedded database so all it takes to do the installation is to install a Java SE runtime environment, unzip, and run the bundle. It is recommended for people that want to review Liferay's functionality or for those who need to have Liferay working quickly.
- *Detailed installation procedure*: explains how to install the portal in an existing application server or one that has been installed using vendor packages. This option is available for all the supported application servers. It is recommended for production environments and deployment to a proprietary app. server.
- *Using the ext environment*: this option provides a full development environment to extend the functionality provided by Liferay. It is recommended for installation in the personal computers of developers doing the customization. It will not be covered in this guide. Refer to the developer guides for more information.

Tip

Liferay 4.3 makes installation easier than ever, because it creates the tables it needs along with example data the first time it starts. All you need to do is to create the database and configure Liferay's datasource to point to it.

Note

Liferay formerly provided two editions, Enterprise and Professional, but this differentiation has been removed in version 4.3. Now there is only one edition called Liferay Enterprise Portal that works in all the supported servlet containers and full application servers. This edition does not use EJBs internally but does fully support them for custom portlets and applications.

Chapter 2. Installing a Liferay Bundle

Liferay Portal is distributed with the following bundle options for servlet containers and full JavaEE application servers:

- Geronimo+Tomcat
- Glassfish
- JBoss+Jetty 4.0
- JBoss+Tomcat 4.0
- JBoss+Tomcat 4.2
- Jetty
- JOnAS+Jetty
- JOnAS+Tomcat
- Pramati
- Resin
- Tomcat 5.5 for JDK 1.4
- Tomcat 5.5 for JDK 5.0
- Tomcat 6.0

Choose your preferred bundle and download it from the downloads page [<http://www.liferay.com/web/guest/downloads>]. Then follow the following steps described next.

1. Quick Installation

Once you have downloaded the bundle you can have Liferay Portal running very quickly following these steps:

1. Download and install JavaSE 5.0 if you have not done so already.

Note: If you picked the *Tomcat 5.5 for JDK 1.4* bundle, use the JavaSE 1.4 version instead. Make sure you have defined the `JAVA_HOME` environment variable.

2. Unzip the bundle to your applications directory. For example: `c:\apps` in Windows or `/usr/local/` in Linux or UNIX variants.
3. Go to the `bin` directory inside the application server directory and run it using the provided script. The name of the script will depend on the application server and you will have to use the version provided for your operating system. Names are usually pretty intuitive such as `run.bat`, `startup.bat`, `jonas.bat`...etc. When in a Unix environment, the batch file to start the server will end with the extension `sh` instead of `bat` and it is necessary to make the script executable by running **`chmod +x filename.sh`**. It is often necessary to run the executable from the directory where it resides.

4. Once the application server has finished initialization (the first time it might take a few minutes) open your browser and go to `http://localhost:8080` (assuming local installation otherwise change localhost with the host name or IP).

The bundle comes with an embedded HSQL database loaded with sample data from the public website of Liferay. You can try Liferay as an anonymous user by navigating your local version the public website.

5. Login as an administrator using the following data:

- User: `test@liferay.com`
- Password: `test`

Congratulations! you now have a running copy of Liferay. Refer to the end user documentation for more information about how to use the administration functionalities and the bundled portlets. Refer to the development and customization documentation to learn how to adapt Liferay to your needs. The next section provides information

2. Customizing the bundle installation

Once the bundle is up and running you can start using Liferay Portal, but before opening the service to general public some configuration is recommended. This configuration will allow an optimized performance when multiple concurrent users access the portal and activates all of Liferay Portal features by integration with external systems

2.1. Database Configuration

The default bundle installation of Liferay Portal uses an embedded database. While this is a good method to have it up and running fast for reviewing or developing, it has several drawbacks:

- Only one user can access it at a time. This is because the data is stored on a file on disk and HSQL locks it when doing changes.
- The data is stored inside the application server and might be lost on redeployment.

Fortunately, Liferay has great support for a good number of production-ready databases, and it is easy to configure Liferay to use them. The exact instructions will depend on the application server and database, but can be summarized as:

1. Create the database in your DBMS manager of choice
2. Configure a DataSource named LiferayPool in your Application Server or servlet container to point to the recently created database
3. Start Liferay. It will automatically create the necessary tables and populate them with example data.

Refer to the detailed installation for instructions on how to configure MySQL or other databases.

2.2. Further Configuration

The bundled configuration connects to a mail SMTP and IMAP server installed in the same machine as

the application server (referred to as localhost in the configuration). It also assumes that certain paths are present in the system.

Refer to chapters 4 and 5 for more information on how to change the system defaults. Also read *Liferay Portal 4 - Customization Guide* for a complete reference of the configuration options provided by Liferay Portal.

Chapter 3. Detailed Installation Procedure

1. Application Servers

This chapter contains detailed instructions for installing Liferay Portal using its WAR distribution. This allows system administrators to deploy Liferay in existing application server installations. It is recommended to have a good understanding of the deployment procedure of Java EE applications in the application server of choice.

Please note that while Liferay Portal supports a wide range of databases, for brevity this section assumes MySQL as the database. To use other databases, please substitute that database JDBC driver and URL to those required by your database. Consult the databases section below for special instructions for some databases.

Note

The following instructions assume an installation on a local machine. When installing to a server just change `localhost` with the host name or IP of the server.

1.1. Geronimo 1.1 with Tomcat 5.0.28/5.5.17

1. Download and install Geronimo/Tomcat [<http://geronimo.apache.org/downloads.html>] into your preferred directory. From now on, the directory where you installed Geronimo will be referred to as `GERONIMO_HOME`.
2. Download and install JDK 5 [<http://java.sun.com/javaee/downloads/index.jsp>]. Set an environment variable called `JAVA_HOME` to point to your JDK directory.
3. Download MySQL from www.mysql.com [<http://www.mysql.com>] and install.
4. Download the WAR for the latest available version of Liferay Portal 4.3 [<http://www.liferay.com/web/guest/downloads/additional>].

5. Create a database for Liferay. For example:

```
create database lportal character set utf8;
```

Liferay will automatically create the tables and populate it the first time it starts.

6. Edit `GERONIMO_HOME\bin\geronimo.bat`

insert at line 219:

```
set JAVA_OPTS=-Xms128m -Xmx512m -Dfile.encoding=UTF8  
-Duser.timezone=GMT
```

7. Download the Portal 4.3 Dependencies [<http://www.liferay.com/web/guest/downloads/additional>].
8. Point browser to `localhost:8080/console` to enter Administration Console.
 - Login in as **User:** system and **Password:** manager
9. Click **Common Libs** under Services.

- Click **Browse**, find portal-kernel.jar and add
- **Group:** Liferay
- **Artifact:** Portal-kernel
- **Version:** enter version number of jar
- **Type:** Jar
- Click **Install**

10.Repeat the last step for each of the libraries in the dependencies ZIP file.

11.Click **Database Pools** under Services

- Click **Using the Geronimo database pool wizard**
- **Name of Database Pool:** LiferayPool
- **Database Type:** MySql
- Click **Next**
- **Driver Jar:** click **Download a Driver** and select **MySQL Connector/J3.0.17**
- Click **Next**
- **DB User Name:** <none>
- **DB Password:** <none>
- **Port:** 3306 (default)
- **Host:** localhost
- **Database:** lportal
- Click **Next**
- Click **Test Connection**
- Click **Deploy**

12.Click **Deploy New** under Applications

- **Archive:** Browse for liferay-portal-4.3.x.war (substitute x with the version you've downloaded)
- Click **Install**

13.Click Web App WARs

- **Uninstall** geronimo/welcome-tomcat/1.1/car
- **Start** -default/liferayportal/xxxxxxx.../war

14.Open your browser to <http://localhost:8080>. Click on My Liferay at the upper right hand corner to enter the login screen. Your login is test@liferay.com and your password is test.

1.2. JBoss 4.03sp1/4.04/4.05 with Jetty 5.1.1

1. Download and install JDK 5 [<http://java.sun.com/javaee/downloads/index.jsp>] . Set an environment variable called `JAVA_HOME` to point to your JDK directory.
2. Download MySQL from www.mysql.com [<http://www.mysql.com>] and install.
3. Download the latest version of the liferay-portal-4.3.x.war [<http://www.liferay.com/web/guest/downloads/additional>].
4. Download or create yourself a package of JBoss + Jetty.

Note: Previously Jetty provided a SAR file that could be deployed along with some configuration in a regular JBoss. As of now that file is no longer available for download. Refer to Jetty's home page [<http://www.mortbay.org/>] or Jetty's wiki [<http://docs.codehaus.org/display/JETTY/Jetty+Wiki>] for detailed instructions on how to integrate it with JBoss.

5. Create file `$JBOSS_HOME/server/default/deploy/liferay-ds.xml` with following content:

```
<?xml version="1.0"?>
<datasources>
  <local-tx-datasource>
    <jndi-name>jdbc/LiferayPool</jndi-name>
    <connection-url>
jdbc:mysql://localhost/lportal?useUnicode=true&characterEncoding=UTF-8
    </connection-url>
    <driver-class>com.mysql.jdbc.Driver</driver-class>
    <user-name></user-name>
    <password></password>
    <min-pool-size>0</min-pool-size>
  </local-tx-datasource>
</datasources>
```

6. Go to `$JBOSS_HOME/server/default/lib/` and create new directory `ext`. Download `mysql-connector-java-{$version}-bin.jar` [<http://www.mysql.com/products/connector/j/>] and copy to this directory. (This is the JDBC connector for MySQL, for other databases, go to appropriate website to download.)
7. Create a database for Liferay. For example:

```
create database lportal character set utf8;
```

Liferay will automatically create the tables and populate it the first time it starts.

8. Download Liferay's Portal 4.3 Dependencies [<http://www.liferay.com/web/guest/downloads/additional>].
 - a. Unzip to `$JBOSS_HOME/server/default/lib/ext`.
9. Set mail properties by replacing the contents of `$JBOSS_HOME/server/default/deploy/mail-service.xml` with:

```
<?xml version="1.0"?>
<server>
  <mbean code="org.jboss.mail.MailService"
```

```

name="jboss:service=MailSession">
  <attribute
name="JNDIName">mail/MailSession</attribute>
  <attribute name="User">nobody</attribute>
  <attribute name="Password">password</attribute>
  <attribute name="Configuration">
    <configuration>
value="imap" />
      <property name="mail.store.protocol"
      <property
name="mail.transport.protocol" value="smtp" />
      <property name="mail.imap.host"
value="localhost" />
      <property name="mail.pop3.host"
value="localhost" />
      <property name="mail.smtp.host"
value="localhost" />
    </configuration>
  </attribute>
</mbean>
</server>

```

10. Configure JAAS. Edit `$JBOSS_HOME/server/default/conf/login-config.xml` and comment out the entire XML for policy 'other' in lines 140-156.

```

<!--<application-policy name = "other">-->
  ...
  <!--<authentication>
    <login-module code =
"org.jboss.security.auth.spi.UsersRolesLoginModule"
    flag = "required" />
  </authentication>
</application-policy>-->

```

11. Deploy liferay-portal-4.3.x.war.

- a. Create directory `$JBOSS_HOME/server/default/deploy/liferay-portal.war`
- b. Unzip `liferay-portal-4.3.x.war` to directory
- c. Go to `JBOSS_HOME/server/default/deploy/liferay-portal.war/lib`
 - i. Move `dom4j.jar`, `jaxen.jar` to `JBOSS_HOME/lib`
 - ii. Move `commons-collections.jar` to `JBOSS_HOME/server/default/lib`
- d. Remove `hibernate3.jar`, `jboss-hibernate.jar` from `JBOSS_HOME/server/default/lib`

12. Edit `$JBOSS_HOME/server/default/deploy/jbossjca-service.xml`:

Change Debug attribute in line 63 from true to false:

```
<attribute name="Debug">false</attribute>
```

13. In

```
$JBOSS_HOME/server/default/deploy/jbossjca-service.xml
```

Comment out deployer service for JSE and EJB2.1 endpoints

line 36-40

```
<!--<mbean name="jboss.ws:service=WebServiceDeployerJSE"
code="org.jboss.ws.server.WebServiceDeployerJSE">
  <depends-list optional-attribute-name="Interceptables">
<depends-list-element>jboss.web:service=WebServer</depends-list-element>
  </depends-list>
</mbean>-->
```

lines 45-49

```
<!--<mbean name="jboss.ws:service=WebServiceDeployerEJB21"
code="org.jboss.ws.server.WebServiceDeployerEJB21">
  <depends-list optional-attribute-name="Interceptables">
<depends-list-element>jboss.ejb:service=EJBDeployer</depends-list-element>
  </depends-list>
</mbean>-->
```

lines 72-75

```
<!--<mbean name="jboss.ws:service=WebServiceDeployerNestedJSE"
code="org.jboss.ws.server.WebServiceDeployerNestedJSE">
  <depends optional-attribute-name="MainDeployer"
proxy-type="attribute">jboss.system:service=MainDeployer</depends>
  <depends>jboss.ws:service=WebServiceDeployerJSE</depends>
</mbean>-->
```

14.Edit

`$(JBOSS_HOME)/server/default/deploy/jms/jbossmq-destinations-service.xml`.
Clear out text between server tags:

```
<?xml version="1.0"?>

<server>
</server>
```

15.Start JBoss. Open your browser to <http://localhost:8080>. Click on My Liferay at the upper right hand corner to enter the login screen. Your login is `test@liferay.com` and your password is `test`.

1.3. JBoss 4.03sp1/4.04/4.05 with Tomcat 5.0.28/5.5.17

1. Download and install JBoss AS [<http://labs.jboss.com/portal/>] into your preferred directory. From now on, the directory where you installed Tomcat will be referred to as `$(JBOSS_HOME)`.
2. Download and install JDK 5 [<http://java.sun.com/javaee/downloads/index.jsp>]. Set an environment variable called `%JAVA_HOME%` (in Windows) or `$(JAVA_HOME)` (in Linux/UNIX) to point to your JDK directory.
3. Download MySQL from www.mysql.com [<http://www.mysql.com>] and install.
4. Download liferay-portal-4.3.x.war [<http://www.liferay.com/web/guest/downloads/additional>].

5. Edit `$JBOSS_HOME/server/default/conf/jboss-service.xml`:

```
<classpath codebase="${jboss.server.lib.url:lib}/ext" archives="*" />
```

6. Delete

`$JBOSS_HOME/server/default/deploy/jbossweb-tomcat55.sar/ROOT.war`.

7. Edit

`$JBOSS_HOME/server/default/deploy/jbossweb-tomcat55.sar/conf/web.xml`:

replace default servlet (lines 79-91) :

```
<servlet>
  <servlet-name>default</servlet-name>
<servlet-class>org.apache.catalina.servlets.DefaultServlet</servlet-class>
  <init-param>
    <param-name>debug</param-name>
    <param-value>0</param-value>
  </init-param>
  <init-param>
    <param-name>listings</param-name>
    <param-value>>true</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>
```

with:

```
<servlet>
  <servlet-name>default</servlet-name>
<servlet-class>org.apache.catalina.servlets.DefaultServlet</servlet-class>
  <init-param>
    <param-name>debug</param-name>
    <param-value>0</param-value>
  </init-param>
  <init-param>
    <param-name>listings</param-name>
    <param-value>>false</param-value>
  </init-param>
  <init-param>
    <param-name>input</param-name>
    <param-value>4096</param-value>
  </init-param>
  <init-param>
    <param-name>output</param-name>
    <param-value>4096</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>
```

8. Create `$JBOSS_HOME/server/default/deploy/liferay-ds.xml` with following content:

```
<datasources>
  <local-tx-datasource>
    <jndi-name>jdbc/LiferayPool</jndi-name>
    <connection-url>
jdbc:mysql://localhost/lportal?useUnicode=true&characterEncoding=UTF-8
    </connection-url>
    <driver-class>com.mysql.jdbc.Driver</driver-class>
```



```

    <user-name></user-name>
    <password></password>
    <min-pool-size>0</min-pool-size>
  </local-tx-datasource>
</datasources>

```

9. Go to `$JBOSS_HOME/server/default/lib/` and create new directory `ext`. Download `mysql-connector-java-{$version}-bin.jar` [<http://www.mysql.com/products/connector/j/>] and copy to this directory. (This is the JDBC connector for MySQL, for other databases, go to appropriate website to download.)

10. Create a database for Liferay. For example:

```
create database lportal character set utf8;
```

Liferay will automatically create the tables and populate it the first time it starts.

11. Download Liferay 's Portal 4.3 Dependencies [<http://www.liferay.com/web/guest/downloads/additional>].

- a. Unzip into `$JBOSS_HOME/server/default/lib/ext`.

12. Set mail properties by replacing the contents of `$JBOSS_HOME/server/default/deploy/mail-service.xml` with:

```

<?xml version="1.0"?>

<server>
  <mbean code="org.jboss.mail.MailService"
name="jboss:service=MailSession">
    <attribute
name="JNDIName">mail/MailSession</attribute>
    <attribute name="User">nobody</attribute>
    <attribute name="Password">password</attribute>
    <attribute name="Configuration">
      <configuration>
        <property name="mail.store.protocol"
value="imap" />
        <property
name="mail.transport.protocol" value="smtp" />
        <property name="mail.imap.host"
value="localhost" />
        <property name="mail.pop3.host"
value="localhost" />
        <property name="mail.smtp.host"
value="localhost" />
      </configuration>
    </attribute>
  </mbean>
</server>

```

13. Configure JAAS. Edit `$JBOSS_HOME/server/default/conf/login-config.xml` and comment out the entire XML for policy `other` in lines 140-156.

```

<!--<application-policy name = "other">-->
  ...
  <!--<authentication>
    <login-module code =

```

```
"org.jboss.security.auth.spi.UsersRolesLoginModule"  
    flag = "required" />  
  </authentication>  
</application-policy>-->
```

14. Deploy liferay-portal-4.3.war.

- a. create new directory
\$JBOSS_HOME/server/default/deploy/liferay-portal.war
- b. unzip liferay-portal-4.3.war to directory
- c. go into \$JBOSS_HOME/server/default/deploy/liferay-portal.war/lib
 - move dom4j.jar,jaxen.jar to JBOSS_HOME/lib
 - move commons-collections.jar goes to JBOSS_HOME/server/default/lib
 - remove hibernate3.jar,jboss-hibernate.jar from JBOSS_HOME/server/default/lib

15. Edit JBOSS_HOME/server/default/deploy/jbossjca-service.xml:

change Debug attribute in line 63 from true to false:

```
<attribute name="Debug">false</attribute>
```

16. Edit JBOSS_HOME/server/default/deploy/jms/jbossmq-destinations-service.xml. Clear out text between server tags:

```
<?xml version="1.0"?>  
  
<server>  
  
</server>
```

17. Start JBoss. Open your browser to <http://localhost:8080>. Click on My Liferay at the upper right hand corner to enter the login screen. Your login is test@liferay.com and your password is test.

1.4. Jetty 5.1.1

1. Download and install JDK 1.4 or 1.5 [<http://java.sun.com/javaee/>]. Set an environment variable called %JAVA_HOME% (in Windows) or \$JAVA_HOME (in Linux/UNIX) to point to your JDK directory.
2. Download MySQL from www.mysql.com [<http://www.mysql.com>] and install.
3. Download and install Jetty 5.1.10-all.zip [http://sourceforge.net/project/showfiles.php?group_id=7322&package_id=132252]. Note: Only this version of Jetty is supported by Liferay. Others may work but will not be covered in this documentation. From now on the home directory where you installed Jetty will be called \$JETTY_HOME.
4. Download liferay-portal-4.3.x.war [<http://www.liferay.com/web/guest/downloads/additional>].
5. Download Liferay's Portal 4.3 Dependencies [<http://www.liferay.com/web/guest/downloads/additional>].

a. Create a `$JETTY_HOME/lib/ext` directory and copy unzip the dependencies in it.

6. Create a database for Liferay. For example:

```
create database lportal character set utf8;
```

Liferay will automatically create the tables and populate it the first time it starts.

7. Edit `$JETTY_HOME/extra/etc/start-plus.config`.

```
$(jetty.home)/lib/ext/  
$(jetty.home)/lib/ext/*
```

8. Create a data source bound to `jdbc/LiferayPool` by editing `$JETTY_HOME/etc/jetty.xml`.

```
<Call name="addService">  
  <Arg>  
    <New class="org.mortbay.jetty.plus.JotmService">  
      <Set name="Name">TransactionMgr</Set>  
      <Call name="addDataSource">  
        <Arg>jdbc/LiferayPool</Arg>  
        <Arg>  
          <New  
class="org.enhydra.jdbc.standard.StandardXADataSource">  
          <Set  
name="DriverName">com.mysql.jdbc.Driver</Set>  
          <Set  
name="Url">jdbc:mysql://localhost/lportal?useUnicode=true&characterEncoding=  
          <Set name="User"></Set>  
          <Set name="Password"></Set>  
          </New>  
        </Arg>  
      </Arg>  
      <Arg>  
        <New  
class="org.enhydra.jdbc.pool.StandardXAPoolDataSource">  
          <Arg type="Integer">4</Arg>  
          <Set name="MinSize">4</Set>  
          <Set name="MaxSize">15</Set>  
          </New>  
        </Arg>  
      </Call>  
    </New>  
  </Arg>  
</Call>
```

9. Download `mysql-connector-java-{$version}-bin.jar` [<http://www.mysql.com/products/connector/j/>] and copy to to `$JETTY_HOME/lib/ext..` (This is the JDBC connector for MySQL, for other databases, go to appropriate website to download.)

10. Create a mail session bound to `mail/MailSession`.

a. Edit `$JETTY_HOME/etc/jetty.xml` and configure a mail session.

```
<Call name="addService">  
  <Arg>  
    <New  
class="org.mortbay.jetty.plus.MailService">  
      <Set name="Name">MailService</Set>  
      <Set
```

```
name="JNDI">mail/MailSession</Set>
                                <Put
name="mail.smtp.host">localhost</Put>
                                </New>
                                </Arg>
</Call>
```

11. Create `$JETTY_HOME/etc/jaas.config`.

```
PortalRealm {
    com.liferay.portal.kernel.security.jaas.PortalLoginModule
required;
};
```

12. Create directory `$JETTY_HOME/webapps/root` and unpack `liferay-portal-4.3.x.war` into it

13. To add support for accessing Liferay's services remotely and to access the document library using WebDAV follow these steps:

- a. Download `tunnel-web.war`
- b. Create a directory called `$JETTY_HOME/webapps/tunnel` and unzip the WAR contents inside

14. Repeat the previous step for `lazslo-web.war` to be able to use the flash based Lazslo tool from portlets and for `cms-web.war` to use the legacy Liferay CMS.

15. Go to `$JETTY_HOME/webapps/root/WEB-INF/lib` and delete `xercesImpl.jar` and `xml-apis.jar`.

16. Copy `$JETTY_HOME/webapps/root/WEB-INF/lib/commons-logging.jar` to `JETTY_HOME/ext` (overwriting existing one).

17. Create batch file.

- a. Create a directory `$JETTY_HOME/bin`
- b. Create `run.bat` (Note, this is for Windows platform. For other platforms, configure accordingly)

```
@echo off

if "" == "%JAVA_HOME%" goto errorJavaHome

%JAVA_HOME%/bin/java -Xmx512m -Dfile.encoding=UTF8
-Duser.timezone=GMT
-Djava.security.auth.login.config=../etc/jaas.config
-DSTART=../extra/etc/start-plus.config -jar ../start.jar
../etc/jetty.xml

goto end

:errorJavaHome
    echo JAVA_HOME not defined.

    goto end

:end
```

Note: If you get a `java.lang.OutOfMemoryError` exception while starting up Jetty, give your JVM more memory by setting `-Xmx512m`.

18. Start Liferay by running `run.bat`. Open your browser to `http://localhost:8080`. Click on My Liferay at the upper right hand corner to enter the login screen. Your login is `test@liferay.com` and your password is `test`.

1.5. Tomcat 5.0.28/5.5.17

1. Download and install Tomcat 5.5.17 [<http://tomcat.apache.org/>] into your preferred directory. From now on, the directory where you installed Tomcat will be referred to as `$TOMCAT_HOME`.
2. Download and install JDK 5 [<http://java.sun.com/javaee/downloads/index.jsp>]. Set an environment variable called `%JAVA_HOME%` (in Windows) or `$JAVA_HOME` (in Linux/UNIX) to point to your JDK directory.

Note: If you are using JDK 1.4, you must download and install the JDK 1.4 Compatibility Package at tomcat.apache.org. **For JDK 1.4 users:** delete `$TOMCAT_HOME/webapps/ROOT/WEB-INF/lib/xercesImpl.jar`. **For JDK 5 users:** move `$TOMCAT_HOME/webapps/ROOT/WEB-INF/lib/xercesImpl.jar` to `TOMCAT_HOME/common/endorsed`.

3. Create and edit `$TOMCAT_HOME/conf/Catalina/localhost/ROOT.xml` to set up the portal web application.

```
<Context path="">
</Context>
```

4. Download `liferay-portal-4.3.x.war` [<http://www.liferay.com/web/guest/downloads/additional>].
5. Download Liferay's Portal 4.3 Dependencies [<http://www.liferay.com/web/guest/downloads/additional>].
 - a. Create a `$TOMCAT_HOME/common/lib/ext` directory and unzip the dependencies ZIP in there. If the files do not extract to this directory, make sure they are in the correct directory by moving them there.
6. Edit `$TOMCAT_HOME/conf/catalina.properties`.

```
common.loader=
  ${catalina.home}/common/classes,\
  .\
  ${catalina.home}/common/lib/ext/*.jar
```

7. Make sure your database server is installed and is working. If it's installed in a different machine make sure that it's accessible from the one where Liferay is being installed.
8. Configure data sources for your database. Make sure the JDBC driver for your database is accessible by Tomcat.
 - a. Obtain the JDBC driver for your version of the database server. In the case of MySQL use `mysql-connector-java-{$version}-bin.jar` [<http://www.mysql.com/products/connector/j/>].
 - b. Copy the JAR file to `$TOMCAT_HOME/common/lib/ext`.

9. Edit `$TOMCAT_HOME/conf/Catalina/localhost/ROOT.xml`.

```
<Context...>
  <Resource
    name="jdbc/LiferayPool"
    auth="Container"
    type="javax.sql.DataSource"
    driverClassName="com.mysql.jdbc.Driver"
url="jdbc:mysql://localhost/lportal?useUnicode=true&characterEncoding=UTF-8"
    username=""
    password=""
    maxActive="100"
    maxIdle="30"
    maxWait="10000"
  />
</Context>
```

10. Create a database for Liferay. For example:

```
create database lportal character set utf8;
```

Liferay will automatically create the tables and populate it the first time it starts.

11. Create a mail session bound to mail/MailSession.

Edit `$TOMCAT_HOME/conf/Catalina/localhost/ROOT.xml` and configure a mail session.

```
<Context...>
  <Resource
    name="mail/MailSession"
    auth="Container"
    type="javax.mail.Session"
    mail.transport.protocol="smtp"
    mail.smtp.host="localhost"
    mail.store.protocol="imap"
    mail.imap.host="localhost"
  />
</Context>
```

12. Configure JAAS.

Edit `$TOMCAT_HOME/conf/Catalina/localhost/ROOT.xml` and configure a security realm.

```
<Context...>
  <Realm
    className="org.apache.catalina.realm.JAASRealm"
    appName="PortalRealm"
userClassNames="com.liferay.portal.security.jaas.PortalPrincipal"
roleClassNames="com.liferay.portal.security.jaas.PortalRole"
    debug="99"
    useContextClassLoader="false"
  />
</Context>
```

13. To add support for accessing Liferay's services remotely and to access the document library using WebDAV follow these steps:

- a. Download `tunnel-web.war`
 - b. Create a directory called `$JETTY_HOME/webapps/tunnel` and unzip the WAR contents inside
 - c. Some versions of Tomcat require an extra step: make a copy of `ROOT.xml` in the same directory and name it `tunnel.xml`
- 14.Repeat the previous step for `lazslo-web.war` to be able to use the flash based Lazslo tool from your own portlets, themes, etc and for `cms-web.war` to use the legacy Liferay CMS.

15.
16.Create `$TOMCAT_HOME/conf/jaas.config`.

```
PortalRealm {
    com.liferay.portal.kernel.security.jaas.PortalLoginModule
required;
};
```

- 17.Edit `$TOMCAT_HOME/bin/catalina.bat` so that Tomcat can reference the login module.

```
rem ----- Execute...

set JAVA_OPTS=-Xms128m -Xmx512m -Dfile.encoding=UTF8
-Duser.timezone=GMT
-Djava.security.auth.login.config=%CATALINA_HOME%/conf/jaas.config
```

- 18.Delete contents `$TOMCAT_HOME/webapps/ROOT` directory.

- 19.Unpack `liferay-portal-4.3.x.war` to `$TOMCAT_HOME/webapps/ROOT`.

- 20.For supporting UTF-8 URIEncoding, edit `$TOMCAT_HOME/conf/server.xml`.

```
<!-- Define a non-SSL HTTP/1.1 Connector on port 8080 -->
  <Connector port="8080" maxHttpHeaderSize="8192"
    maxThreads="150" minSpareThreads="25"
maxSpareThreads="75"
    enableLookups="false" redirectPort="8443"
acceptCount="100"
    connectionTimeout="20000" disableUploadTimeout="true"
    URIEncoding="UTF-8"
  />
```

- 21.Run Tomcat, point browser to `http://localhost:8080`. Sign in as `test@liferay.com` and password `test`

1.6. Resin 3.0.19

1. Download and install Resin [<http://www.caucho.com/>] into your preferred directory. From now on, the directory where you installed Resin will be referred to as `$JBOSS_HOME`.
2. Download and install JDK 5 [<http://java.sun.com/javaee/downloads/index.jsp>]. Set an environment variable called `%JAVA_HOME%` (in Windows) or `$JAVA_HOME` (in Linux/UNIX) to point to your JDK directory.

- Download MySQL from www.mysql.com [<http://www.mysql.com>] and install.
- Edit `$RESIN_HOME/conf/resin.conf`.

replace lines 60-64

```
<class-loader>
  <tree-loader path="{resin.home}/lib"/>
  <tree-loader path="{server.root}/lib"/>
</class-loader>
```

with

```
<class-loader>
  <tree-loader path="{resin.home}/lib"/>
  <tree-loader path="{server.root}/lib"/>
  <compiling-loader path="{server.rootDir}/common/classes"/>
  <library-loader path="{server.rootDir}/common/lib"/>
</class-loader>
```

add:

```
<database>
  <jndi-name>jdbc/LiferayPool</jndi-name>
  <driver type="com.mysql.jdbc.Driver">
<url>jdbc:mysql://localhost/lportal?useUnicode=true&characterEncoding=UTF-8<
  <user></user>
  <password></password>
  </driver>
<prepared-statement-cache-size>8</prepared-statement-cache-size>
  <max-connections>20</max-connections>
  <max-idle-time>30s</max-idle-time>
  </database>

  <resource jndi-name="mail/MailSession"
type="javax.mail.Session">
  <init>
<mail.store.protocol>imap</mail.store.protocol>
<mail.transport.protocol>smtp</mail.transport.protocol>
  <mail.imap.host>localhost</mail.imap.host>
  <mail.pop3.host>localhost</mail.pop3.host>
  <mail.smtp.host>localhost</mail.smtp.host>
  </init>
  </resource>

  <system-property
javax.xml.parsers.DocumentBuilderFactory="org.apache.xerces.jaxp.DocumentBuilder
/>
  <system-property
javax.xml.parsers.SAXParserFactory="org.apache.xerces.jaxp.SAXParserFactoryImpl
/>
  <system-property
javax.xml.transform.TransformerFactory="org.apache.xalan.processor.TransformerFa
/>
  <system-property
org.xml.sax.driver="org.apache.xerces.parsers.SAXParser" />
```

- Go to `$RESIN_HOME` and create new directory `common\lib`. Download `mysql-connector-java-{$version}-bin` [<http://www.mysql.com/products/connector/j/>]. jar and copy to this directory. (This is the JDBC

connector for MySQL, for other databases, go to appropriate website to download.)

6. Create a database for Liferay. For example:

```
create database lportal character set utf8;
```

Liferay will automatically create the tables and populate it the first time it starts.

7. Download the Liferay Portal 4.3 Dependencies [<http://www.liferay.com/web/guest/downloads/additional>] and unzip into `$RESIN_HOME\common\lib`.
8. Delete contents of `$RESIN_HOME\webapps\ROOT`.
9. Unzip `liferay-portal-4.3.x.war` to `$RESIN_HOME\webapps\ROOT`.
10. Download the sources of Liferay Portal and unzip them to a temporal directory:
 - a. Go to `$(lib)\development\` and copy `activation.jar` and `mail.jar` to `$RESIN_HOME\common\lib`, `saxpath.jar` and `xalan.jar` to `$RESIN_HOME\lib`
 - b. Go to `$(lib)\portal` and copy `xercesImpl.jar` and `xml-apis.jar` to `$RESIN_HOME\lib`
11. Go to `$RESIN_HOME\bin` and create `run.bat`.

```
..\httpd.exe -Xmx512m -Dfile.encoding=UTF-8 -Duser.timezone=GMT
```
12. Start Resin. Open your browser to `http://localhost:8080`. Click on **Access** at the upper right hand corner to enter the login screen. Your login is `test@liferay.com` and your password is `test`.

1.7. Websphere 6.0.2.5

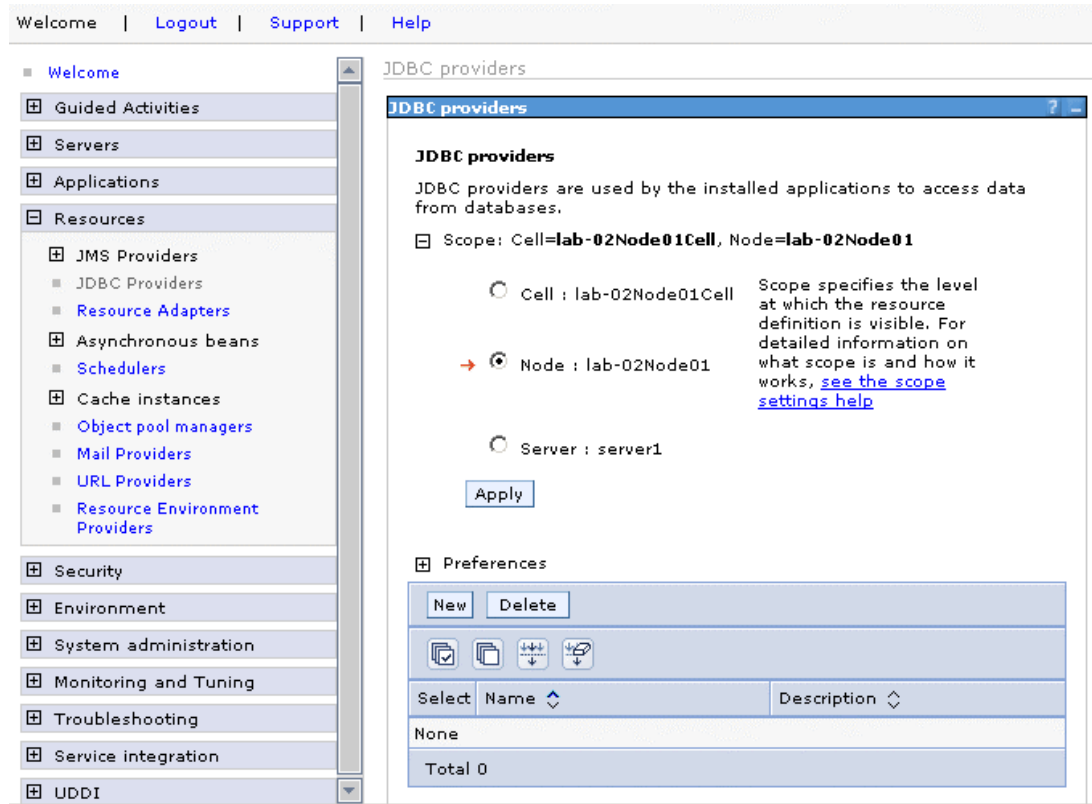
Note: Throughout this installation and configuration process, Websphere will prompt you to Click Save to apply changes to Master Configuration. Do so intermittently to save your changes.

Installation

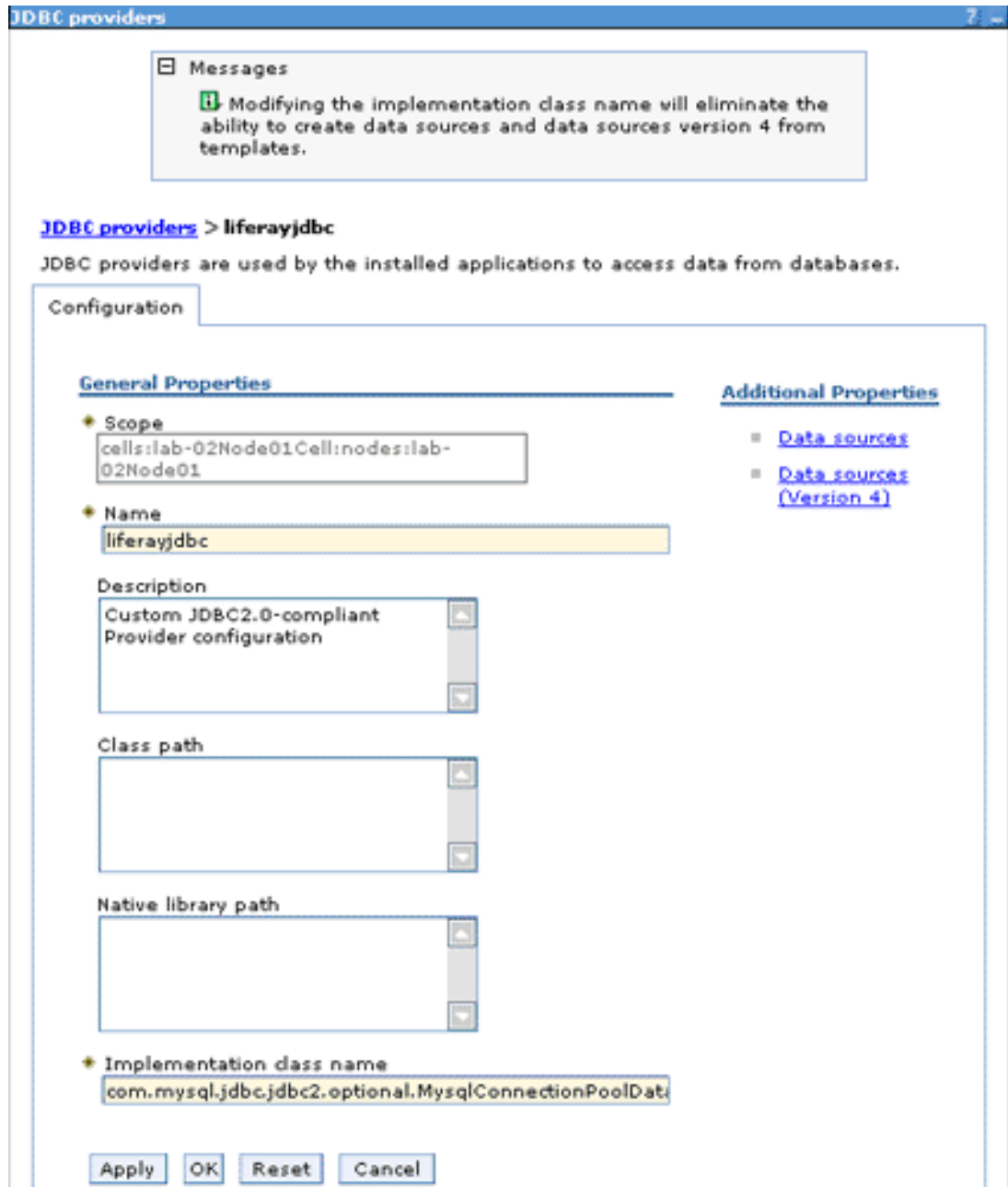
1. Download Liferay Portal 4.3.x WAR , unzip and compile.
2. Install IBM Websphere.
3. Install MySql.
4. Download and extract these Liferay jars to `websphere\appserver\lib\ext`.
 - Dependency libraries (Liferay Portal 4.3 Dependencies [<http://www.liferay.com/web/guest/downloads/additional>])
 - `liferay-portal-jaas.jar` (Liferay Portal Enterprise 4.2 JAAS Libraries from Sourceforge)
 - `mysql-connector-java-3.x.x-bin.jar` (MySQL [<http://dev.mysql.com/downloads/connector/j/3.1.html>])

Set Up Database Service

1. Start Websphere.
2. Open Administrative Console and login.
3. Click **Resources**, click **JDBC Providers**.



4. Click **Next**.
5. For name, enter name of jdbc provider, e.g. liferayjdbc.
6. Clear any text in classpath.
7. For Implementation class name enter
`com.mysql.jdbc.jdbc2.optional.MysqlConnectionPoolDataSource`



8. Click **OK**.
9. Click **Data sources** under Additional Properties.
10. Click **New**.
11. Enter a name: liferaydatabasesource.
12. Enter JNDI, jdbc/LiferayPool.
13. Everything else should stay to default.
14. Click **OK**.

General Properties

Scope
cells:lab-02Node01:Cell/nodes:lab-02Node01

Name
liferaydatabasesource

JNDI name
jdbc/LiferayPool

Use this Data Source in container managed persistence (CMP)

Description

Category

Data store helper class name

Select a data store helper class
Data store helper classes provided by WebSphere Application Server
Generic data store helper
(com.ibm.websphere.rsadapter.GenericDataStoreHelper)

Specify a user-defined data store helper
Enter a package-qualified data store helper class name

Component-managed authentication alias
Component-managed authentication alias
(none)

Authentication alias for XA recovery

Use component-managed authentication alias

Specify:
(none)

Container-managed authentication

Container-managed authentication alias (deprecated in V6.0, use resource reference authentication settings instead)
(none)

Mapping-configuration alias (deprecated in V6.0, use resource reference authentication settings instead)
(none)

The additional properties will not be available until the general properties for this item are saved.

Additional Properties

- Connection pool properties
- WebSphere Application Server data source properties
- Custom properties

Related Items

- J2EE Connector Architecture (J2C) authentication data entries

15. Under Additional Properties, click **Custom properties**.

16. Click **New**.

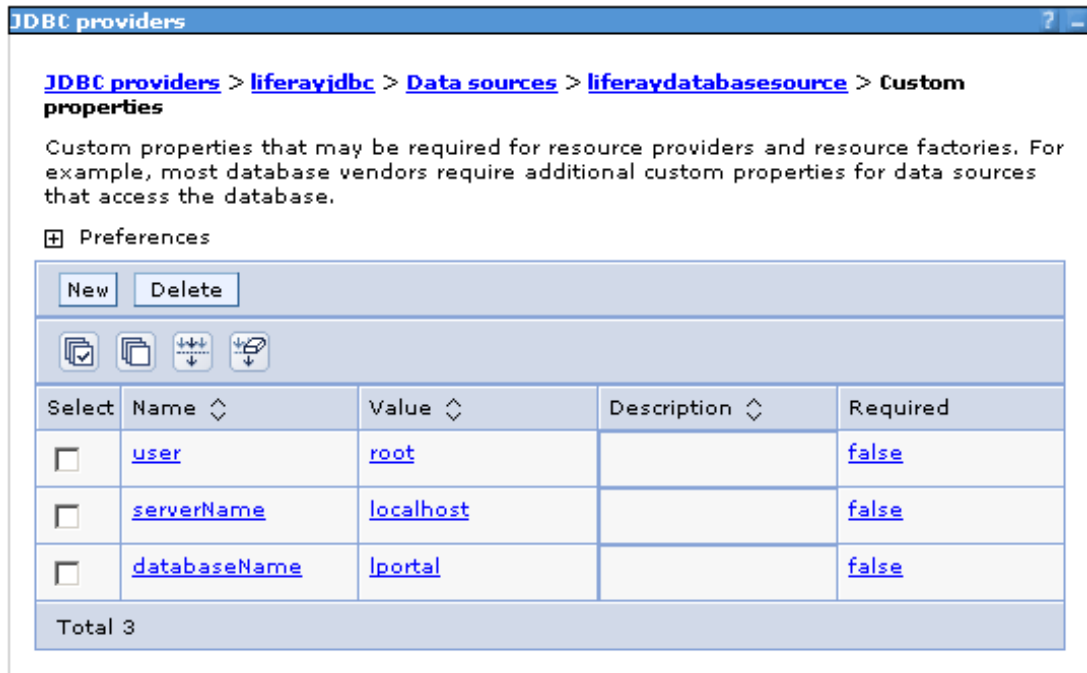
17. Create 3 custom properties by entering **Name**, **Value** and clicking **OK** for each row on this table.

Table 3.1.

name	value
1. user	root
2. serverName	localhost
3. databaseName	lportal

The screenshot shows the 'JDBC providers' configuration window. The breadcrumb path is: **JDBC providers** > **liferayjdbc** > **Data sources** > **liferaydatabasesource** > **Custom properties** > **New**. Below the breadcrumb is a descriptive paragraph: 'Custom properties that may be required for resource providers and resource factories. For example, most database vendors require additional custom properties for data sources that access the database.' The 'Configuration' tab is active, showing the 'General Properties' section. The 'Scope' field contains 'cells:lab-02Node01Cell:nodes:lab-02Node01'. The 'Required' checkbox is unchecked. The 'Name' field contains 'user', and the 'Value' field contains 'root'. The 'Description' field is empty. The 'Type' dropdown menu is set to 'java.lang.String'. At the bottom, there are buttons for 'Apply', 'OK', 'Reset', and 'Cancel'.

18. When done correctly, custom properties should look like this:



19. Create a database for Liferay. For example:

```
create database lportal character set utf8;
```

Liferay will automatically create the tables and populate it the first time it starts.

20. Click data sources test connection to test.

Mail Configuration

1. Click **Resources, Mail providers**.
2. Click **Built-in Mail Provider**.
3. Click **Mail Sessions**.
4. Click **New**.
 - a. **Name:** liferaymail
 - b. **JNDI name:** mail/MailSession



5. Click **OK**.
6. Click **Security**.
7. Click **Global security**.
8. Select **Enable global security**.
9. Deselect **enforce java 2 security**.

10. In **Active user registry**, select custom user registry.

11. Click **Apply** to go to Custom user registry page.

The screenshot shows the 'Global security' configuration window. The title bar reads 'Global security'. Below the title bar, there is a section titled 'Global security' with a descriptive paragraph: 'Specifies the global security configuration for a managed domain. The following steps are required to turn on security: 1. Configure the desired user registry listed under User registries and set its properties. 2. Select the Enable global security option on this panel. 3. Select the configured user registry type from the Active user registry option on this panel.'

The main configuration area is divided into two columns. The left column is titled 'Configuration' and contains a sub-section 'General Properties' with the following options:

- Enable global security
- Enforce Java 2 security
- Enforce fine-grained JCA security
- Use domain-qualified user IDs
- * Cache timeout: 600 seconds (input field)
- Issue permission warning
- Active protocol: CSI and SAS (dropdown)
- Active authentication mechanism: Simple WebSphere Authentication Mechanism (SWAM) (dropdown)
- Active user registry: Custom user registry (dropdown)
- Use the Federal Information Processing Standard (FIPS)

At the bottom of the left column are buttons for 'Apply', 'OK', 'Reset', and 'Cancel'.

The right column contains several expandable sections:

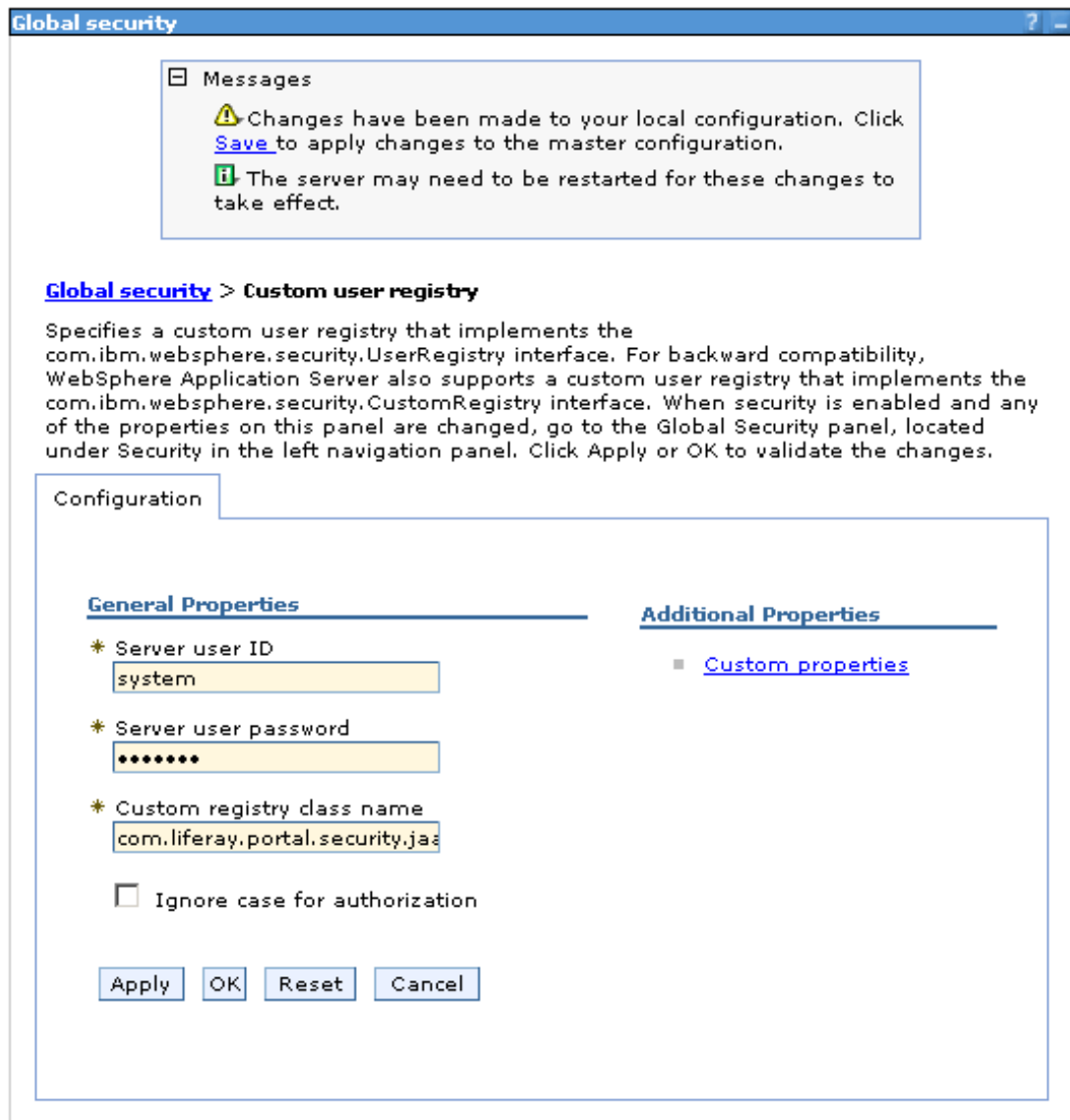
- User registries**:
 - Custom
 - LDAP
 - Local OS
- Authentication**:
 - Authentication mechanisms
 - Authentication protocol
 - JAAS Configuration
- Authorization**:
 - Authorization providers
- Additional Properties**:
 - Custom properties

12. Enter 'system' for server user ID.

13. Enter 'password' for server user password.

14. Enter Custom registry class name
`com.liferay.portal.security.jaas.ext.websphere.PortalUserRegistry`.

15. Click **Apply**.



16. Insert username/password into database.

17. Open a mysql console.

18. Enter Use `lportal`

19. Enter `Insert into User_ (companyId, userId, password_) values ('system', 'system', 'password');`

```

C:\WINDOWS\system32\cmd.exe - mysql
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 15 to server version: 4.1.20-community-nt

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

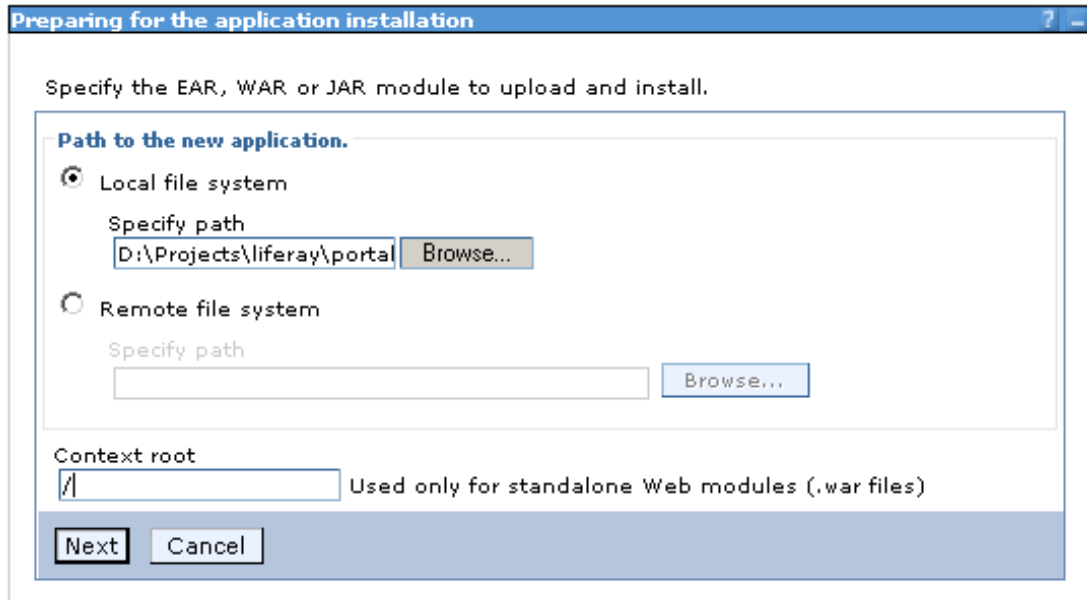
mysql> use lportal
Database changed
mysql> insert into User_ (companyId, userId, password_) values ('system', 'system', 'password');
Query OK, 1 row affected (0.03 sec)

mysql>

```

Install Liferay

1. Click **Applications**, click **Install new applications**
2. Browse for liferay-portal-4.3.x.war.
3. Enter context root '/'.



4. Click **Next**
5. Select **Generate Default Bindings>Override default bindings>Use default virtual host name for web modules:**

Preparing for the application installation

Choose to generate default bindings and mappings.

Generate Default Bindings

Override:

Do not override existing bindings

Override existing bindings

Virtual Host

Do not use default virtual host name for Web modules

Use default virtual host name for Web modules:

Host name
default_host

Specific bindings file
Browse...

Previous Next Cancel

6. Click **Next**. Click **Continue**. For **Steps 1 to 4**, click **Next** to apply defaults.

Install New Application ? -

Specify options for installing enterprise applications and modules.

→ **Step 1: Select installation options**

[Step 2: Map modules to servers](#)

[Step 3: Map resource references to resources](#)

[Step 4: Map virtual hosts for Web modules](#)

[Step 5: Map security roles to users/groups](#)

[Step 6: Summary](#)

Select installation options

Specify the various options that are available to prepare and install your application.

Pre-compile JSP

Directory to install application

Distribute application

Use Binary Configuration

Deploy enterprise beans

Application name

Create MBeans for resources

Enable class reloading

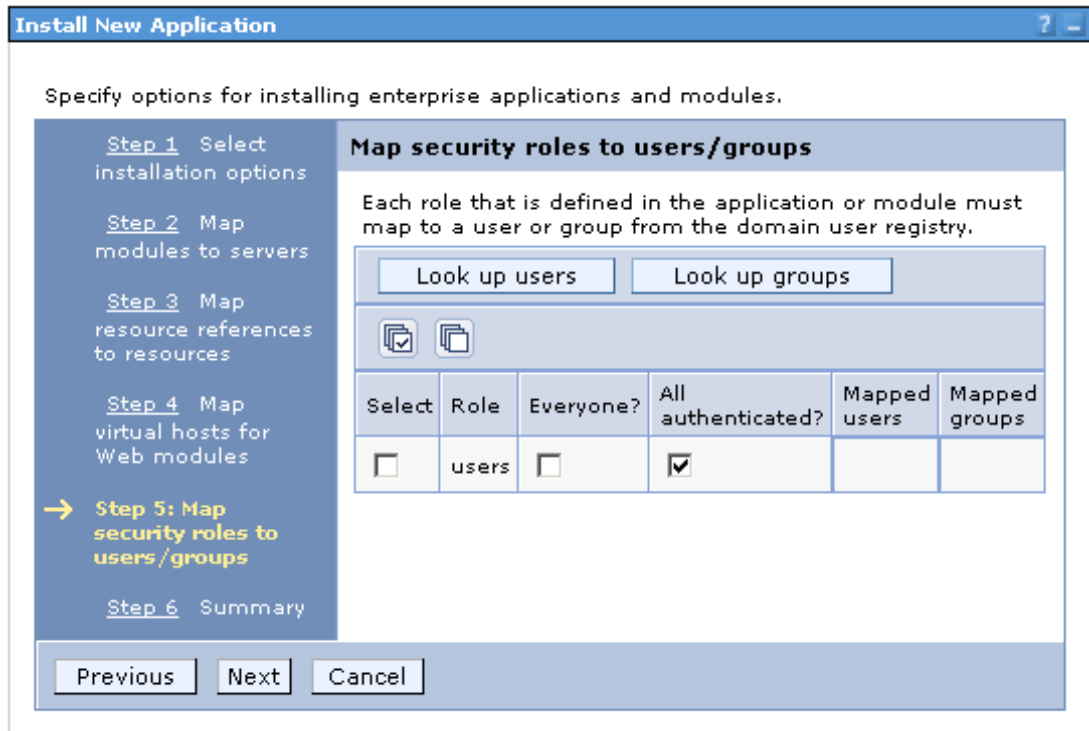
Reload interval in seconds

Deploy Web services

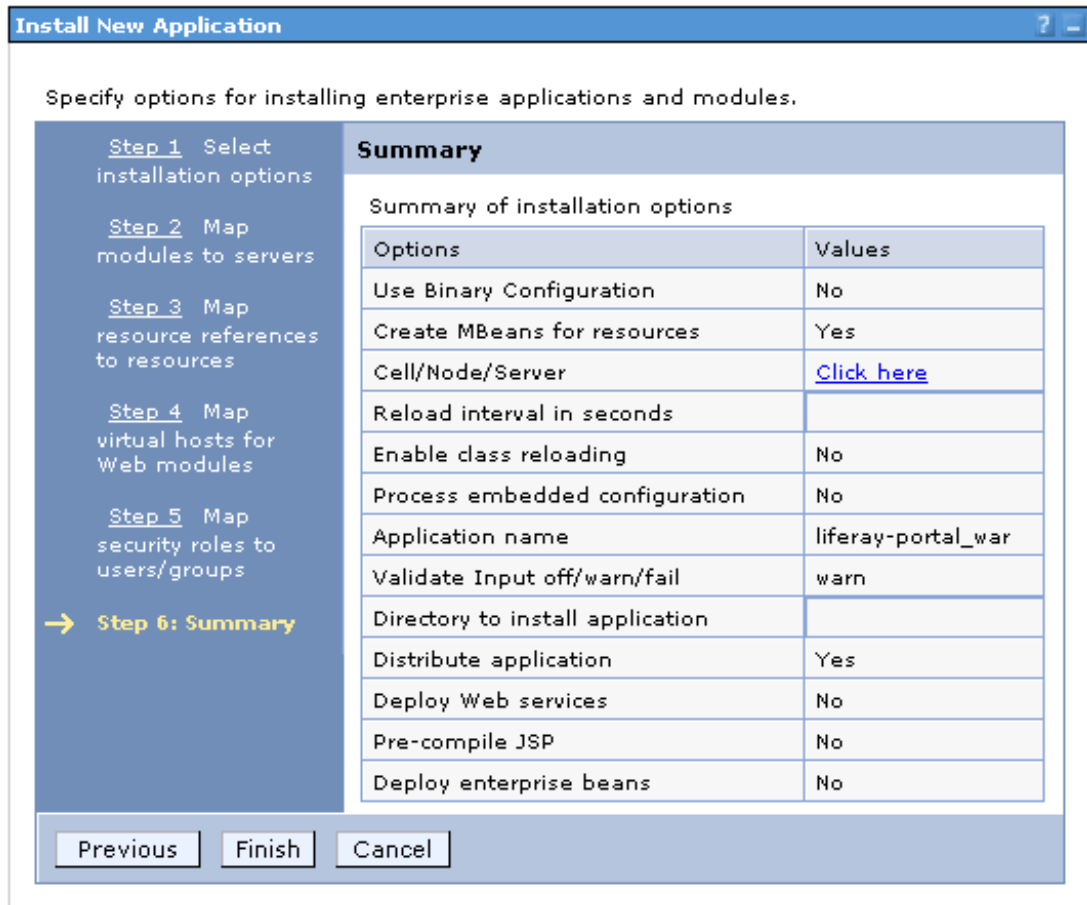
Validate Input off/warn/fail

Process embedded configuration

7. In **Step 5**, check **all authenticated**



8. Click **Next**.



9. Click **Finish**.

10. Wait for installation process.

Installing...

If there are enterprise beans in the application, the EJB deployment process can take several minutes. Please do not save the configuration until the process completes.

Check the SystemOut.log on the Deployment Manager or server where the application is deployed for specific information about the EJB deployment process as it occurs.

ADMA5016I: Installation of liferay-portal_war started.

ADMA0014E: Validation failed. Duplicate root context(/) was found on the samenode lab-02Node01 and same default host default_host

ADMA5068I: The resource validation for application liferay-portal_war completed successfully, but warnings occurred during validation.

ADMA5058I: Application and module versions validated with versions of deployment targets.

ADMA5005I: The application liferay-portal_war is configured in the WebSphere Application Server repository.

ADMA5053I: The library references for the installed optional package are created.

ADMA5005I: The application liferay-portal_war is configured in the WebSphere Application Server repository.

ADMA5001I: The application binaries are saved in C:\Program Files\IBM\WebSphere\AppServer\profiles\default1wstemp\0\workspace\cells\lab-02Node01 Cell\applications\liferay-portal_war.ear\liferay-portal_war.ear

ADMA5005I: The application liferay-portal_war is configured in the WebSphere Application Server repository.

SECJ0400I: Successfully updated the application liferay-portal_war with the appContextIDForSecurity information.

ADMA5011I: The cleanup of the temp directory for application liferay-portal_war is complete.

ADMA5013I: Application liferay-portal_war installed successfully.

Application liferay-portal_war installed successfully.

To start the application, first save changes to the master configuration.

[Save to Master Configuration](#)

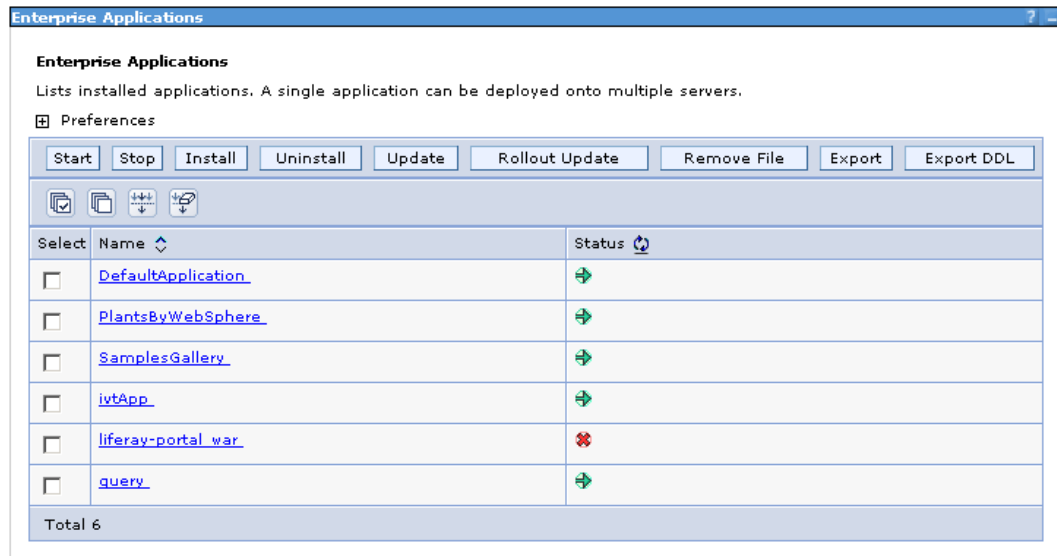
To work with installed applications, click the "Manage Applications" button.

[Manage Applications](#)

11. Save this configuration to master configuration by clicking on **System administration** and **Save Changes to Master Repository**.

Start Liferay Portal

1. Applications.
 - a. Click **Enterprise Applications**.



- b. Uninstall **DefaultApplication**, **PlantsByWebSphere** and **SamplesGallery**.
 - c. Select **liferay-portal.war**, click **start**
2. Open up browser and point to <http://localhost:9080>. Liferay portal home page should be seen.
 3. Edit your Stop the Server shortcut to set the user id and password. If you don't do this, you will not be able to stop the server after you restart WebSphere:


```
"C:\Program Files\WebSphere\AppServer\bin\stopServer.bat" server1 -user system -password password
```
 4. Stop Websphere and retart it. Login on with '**system**' for username and '**password**' for password.

2. Databases

Please note that this documentation uses MySQL as the database. To use other databases, please subsitute that database JDBC driver and configure accordingly. Consult your Database documentation for more details. Below are notes for known special case instructions for other databases.

2.1. Oracle

For Oracle 9 and 10 use jdbc driver ojdbc14.jar
[\[http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/htdocs/jdbc_10201.html\]](http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/htdocs/jdbc_10201.html)

For Oracle 10, also follow this step:

1. Edit portal.properties and add:

```
hibernate.jdbc.batch_size=0
```

Do not use the default setting of 20.

Chapter 4. Configuring Liferay Portal Paths

Liferay Portal's configuration is held in two files: `portal.properties` and `system.properties`. After installing the portal both files can be found in `WAR-FILE/WEB-INF/classes`

The location of the WAR file will depend on the application server. Check its documentation if you cannot find them. It is not recommended to directly modify `portal.properties` and `system.properties`. Instead create two files named `portal-ext.properties` and `system-ext.properties` and write in only the properties whose values you want to override. These two files can be placed with the original one or in the global classpath of the application server.

It is recommended to review and adjust the values of the following properties (shown with their default values at the time of writing):

- `auto.deploy.deploy.dir=${user.home}/liferay/deploy`: necessary to enable autodeploy
- `lucene.dir=${user.home}/liferay/lucene/`: necessary for search to work
- `jcr.jackrabbit.repository.root=${user.home}/liferay/jackrabbit`: necessary for the document library

For more information about this files and the configuration possibilities provided by Liferay, read the *Liferay Portal 4 - Customization Guide*.

Chapter 5. Integration with External Systems

This chapter covers the steps necessary to integrate Liferay Portal with external systems. This configuration may be necessary for the correct operation provided by some the bundled portlets.

1. Mail Servers

The section about the detailed installation procedure explains the integration with mail systems for sending emails through SMTP and browsing them through IMAP. This section will cover how to extend the integration to achieve synchronization of user accounts and setting of options related to the usage of the mail system.

Liferay Portal can integrate with Washington IMAP+Sendmail, Cyrus IMAP+Postfix, and Dovecot+Postfix. Support for integration with Microsoft Exchange and other IMAP servers are planned and will be implemented in the near future.

The portal synchronizes with the mail server's user authentication by adding a mail server account when a portal account is added, deleting a mail server account when a portal account is deleted, and updating a mail server account when a portal account is updated. To do this, the portal must have privileges to modify and to update the mail server's user database.

The portal must also keep track of how email addresses map to certain accounts. For example, in the default installation, the portal maps the user id `liferay.com.1` to the email address `test@liferay.com`.

Users access their email through an IMAP server. Access is limited to IMAP so that the portal does not have to be programmed to know where to persist the mail.

1.1. Washington IMAP+Sendmail

1. Install Sendmail and Expect on your mail server. Expect allows you to add, delete, or update users in one command. An example script for Red Hat is included in `/mail-impl/scripts/redhat`.
2. Configure `/portal-impl/classes/portal.properties` for your mail server.
3. The following instructions assume:
 - The server environment is linux
 - The server name is called `PORTAL_HOST`
 - You are logged in as root
 - The distribution is Liferay Portal bundled with Tomcat
 - Tomcat is installed at `/usr/local/tomcat`
 - Tomcat is running under the user named `tomcat`, group name `tomcat`
 - You are using sendmail for email

- Portal sendmail users are created under the path /home/liferay/users
- sendmail is running on PORTAL_HOST

```
# Install expect command
apt-get install expect
```

```
# Give tomcat user a password
passwd tomcat
```

```
#give tomcat user a login shell
vi /etc/passwd
tomcat:x:500:500::/usr/local/tomcat:/bin/bash
```

```
# Use sudo to allow tomcat to add users
visudo
```

```
Defaults logfile=/var/log/sudolog
Defaults:tomcat timestamp_timeout=-1, passwd_tries=1
tomcat ALL=/usr/sbin/adduser, /usr/sbin/userdel, /usr/bin/passwd
```

```
# Enable UW-imap
vi /etc/xinet.d/imap
```

```
# default: off
# description: The IMAP service allows remote users to access their
mail using \
#           an IMAP client such as Mutt, Pine, fetchmail, or
Netscape \
#           Communicator.
service imap
{
    socket_type           = stream
    wait                  = no
    user                   = root
    server                 = /usr/sbin/imapd
    log_on_success        += HOST DURATION
    log_on_failure        += HOST
    disable                = no
}
```

```
# Restart the xinetd daemon
/etc/rc.d/init.d/xinetd restart
```

```
# Add Tomcat mail/MailSession settings
```

```
vi /usr/local/tomcat/conf/Catalina/localhost/liferay.xml
    <parameter>
        <name>mail.smtp.host</name>
        <value>localhost</value>
    </parameter>
    <parameter>
        <name>mail.imap.host</name>
        <value>localhost</value>
    </parameter>
    <parameter>
        <name>mail.store.protocol</name>
        <value>imap</value>
    </parameter>
    <parameter>
        <name>mail.transport.protocol</name>
        <value>smtp</value>
    </parameter>
    <parameter>
        <name>mail.pop3.host</name>
        <value>localhost</value>
    </parameter>

# Make the email mapping table writable by tomcat
chmod 664 /etc/mail/virtusertable
chmod 664 /etc/mail/virtusertable.db
chgrp tomcat /etc/mail/virtusertable

# Create lucent paths
mkdir /usr/local/tomcat/liferay/lucene

# Create sendmail users path
mkdir /home/liferay
mkdir /home/liferay/users

chown -R tomcat /home/liferay
chgrp -R tomcat /home/liferay
chmod -R 660 /home/liferay

# Create custom portal properties
# see http://www.liferay.com/static/content/portal.properties.html

vi /usr/local/tomcat/common/classes/portal-ext.properties

mail.hook.impl=com.liferay.mail.util.SendmailHook
mail.mx.update=true
mail.hook.sendmail.add.user=/usr/local/tomcat/bin/autouseradd %1%
mail.hook.sendmail.change.password=/usr/local/tomcat/bin/autopasswd
%1% %2%
mail.hook.sendmail.delete.user=/usr/local/tomcat/bin/autouserdel %1%
mail.hook.sendmail.home=/home/liferay/users
mail.hook.sendmail.virtusertable=/etc/mail/virtusertable
mail.box.style=mail/
mail.username.replace=true
passwords.allow.dictionary.word=false
```

```
mail.junk-mail.warning.size=512000
mail.trash.warning.size=512000
mail.attachments.max.size=3072000
mail.audit.trail=root@PORTAL_HOST
lucene.dir /usr/local/tomcat/liferay/lucene/
```

```
# Create change password command
vi /usr/local/tomcat/bin/autopasswd
```

```
#!/usr/bin/expect -f
set password [lindex $argv 1]
```

```
spawn sudo /usr/bin/passwd [lindex $argv 0]
expect -i $spawn_id "password:"
sleep .5
send "$password\r"
expect "password:"W
sleep .5
send "$password\r"
expect eof
```

```
# Create user add command
vi /usr/local/tomcat/bin/autouseradd
```

```
#!/usr/bin/expect -f
# 1st argument is the user id to add.
# Note: setting mail.username.replace=true in
# /common/classes/portal-ext.properties
# will replace the .'s with _'s in userid, which is required for
# linux
```

```
set userid [lindex $argv 0]
spawn sudo /usr/sbin/adduser $userid -s /bin/false
expect eof
```

```
# Create user remove command
vi /usr/local/tomcat/bin/autouserdel
```

```
#!/usr/bin/expect -f
# 1st argument is the user id to remove
# Note: setting mail.username.replace=true in
# /common/classes/portal-ext.properties
# will replace the .'s with _'s in userid, which is required for
# linux
```

```
set userid [lindex $argv 0]
spawn sudo /usr/sbin/userdel -r $userid
```

```
expect eof
```

```
# Set command file permissions
chmod 700 /usr/local/tomcat/bin/autopasswd
chown tomcat /usr/local/tomcat/bin/autopasswd
chgrp tomcat /usr/local/tomcat/bin/autopasswd
chmod 700 /usr/local/tomcat/bin/autouseradd
chown tomcat /usr/local/tomcat/bin/autouseradd
chgrp tomcat /usr/local/tomcat/bin/autouseradd
chmod 700 /usr/local/tomcat/bin/autouserdel
chown tomcat /usr/local/tomcat/bin/autouserdel
chgrp tomcat /usr/local/tomcat/bin/autouserdel

# Activate tomcat sudo, so it never prompts again
su tomcat
/usr/local/tomcat/bin/autouseradd badusername
/usr/local/tomcat/bin/autopasswd badusername asst1453
/usr/local/tomcat/bin/autouserdel badusername
exit
```

1.2. Cyrus IMAP+Postfix

1. Install Fedora Core 4 [<http://fedora.redhat.com/>].

For a minimal installation, choose to install a custom server. Deselect all packages groups. Select the package groups: Text-based Internet, Mail Server, DNS Name Server, FTP Server, MySQL Database, Network Servers, Development Tools, Legacy Software Development, Administration Tools, and System Tools.

Make sure the following RPMs are also selected. The packages `cyrus-imapd` and `cyrus-imapd-utils` are only available in Fedora Core 2 and Fedora Core 4. They were not part of Fedora Core 1 and needed to be compiled manually. In Fedora Core 4, they were moved to Extras and you will need to use `yum` to install these packages.

Mail Server: `+cyrus-imapd, +cyrus-imapd-utils`

MySQL Database: `+mysql-server`

Development Tools: `+expect`

2. Update Fedora. This may take a while even if you have a fast connection.

```
rpm --import /usr/share/rhn/RPM-GPG-KEY-fedora yum list yum upgrade
```

3. Turn off Sendmail.

```
chkconfig --level 3 sendmail off
/etc/rc.d/init.d/sendmail stop
```

4. Edit `/etc/sysconfig/saslauthd`.

Replace **MECH=shadow** with **MECH=pam**.

Turn on Cyrus SASL.

```
chkconfig --level 3 saslauthd on
/etc/rc.d/init.d/saslauthd start
```

5. Download Cyrus IMAP. If you are using Fedora Core 2 or later, you can use the RPMs from Fedora: `cyrus-imapd` and `cyrus-imapd-utils`. If you are using Fedora Core 1 or an earlier version of Red Hat, download `cyrus-imapd-2.1.16-6.src.rpm` [<http://www.invocha.ch/pub/packages/cyrus-imapd/2.1/cyrus-imapd-2.1.16-6.src.rpm>] and build the RPM for your environment from the source distribution.

Build Cyrus IMAP.

```
rpmbuild --rebuild cyrus-imapd-2.1.16-6.src.rpm
```

Install Cyrus IMAP.

```
rpm -i cyrus-imapd-2.1.16-6.i386.rpm
```

```
rpm -i cyrus-imapd-utils-2.1.16-6.i386.rpm
```

Turn on Cyrus IMAP.

```
chkconfig --level 3 cyrus-imapd on
/etc/rc.d/init.d/cyrus-imapd start
```

6. Download the source distribution of Postfix [<http://ftp.wl0.org/official/2.1/SRPMS/postfix-2.1.6-1.src.rpm>].

Install Postfix with support for MySQL and Cyrus SASL.

```
rpm -ivh postfix-2.1.6-1.src.rpm
```

```
cd /usr/src/redhat/SOURCES
```

```
bash
```

```
export POSTFIX_MYSQL_REDHAT=1
```

```
export POSTFIX_SASL=2
```

```
export POSTFIX_TLS=1
```

```
sh make-postfix.spec
```

```
exit
```

```
cd /usr/src/redhat/SPECS
```

```
rpmbuild -ba postfix.spec
```

```
cd /usr/src/redhat/RPMS/i386
```

```
rpm -i --force postfix-2.1.6-1.mysql.sasl2.tls.fc4.i386.rpm
```

7. Download the source distribution of PAM MySQL
[http://www.invocha.ch/pub/packages/cyrus-imapd/contrib/pam_mysql-0.5-0.src.rpm].

Install PAM MySQL.

```
rpm -ivh pam_mysql-0.5-0.src.rpm
```

```
cd /usr/src/redhat/SPECS
```

```
rpmbuild -ba pam_mysql.spec
```

```
cd /usr/src/redhat/RPMS/i386
```

```
rpm -i pam_mysql-0.5-0.i386.rpm
```

8. Copy /mail-impl/scripts/fedora/cyrus/mysql_virtual.cf to /etc/postfix/mysql_virtual.cf. Modify mysql_virtual.cf to point to your MySQL database.

Edit /etc/postfix/virtual. Add the line **yourdomain.com anything** for each virtual domain that Postfix will manage. A corresponding entry is needed in the MySQL database so that email to postmaster@yourdomain.com can be delivered to a Cyrus IMAP account.

Transform /etc/postfix/virtual to a format Postfix can read.

```
postmap /etc/postfix/virtual
```

Edit /etc/postfix/master.cf. Replace the two instances of /cyrus/bin/deliver with /usr/lib/cyrus-imapd/deliver. Add these two lines:

```
procmail unix - n n - - pipe
  flags=R user=cyrus argv=/usr/bin/procmail -t -m USER=${user}
EXTENSION=${extension} /home/cyrus/procmailrc
```

Edit /etc/postfix/main.cf. Add these lines:

```
#
# Custom Settings
#

mynetworks = 127.0.0.0/8, 192.168.0.0/16, 128.135.12.7/32

mailbox_command = /usr/bin/procmail -t -a "$EXTENSION"
mailbox_transport = procmail

virtual_maps = hash:/etc/postfix/virtual,
mysql:/etc/postfix/mysql_virtual.cf

smtpd_recipient_restrictions = permit_mynetworks,
check_client_access hash:/etc/postfix/pop-before-smtp,
check_relay_domains
```

Set mynetworks to the IPs that are allowed to connect to Postfix. Turn on Postfix.

chkconfig --level 3 postfix on

/etc/rc.d/init.d/postfix start

9. Copy /mail-impl/scripts/fedora/cyrus/procmailrc to /home/cyrus/procmailrc. Make sure the cyrus user can access the script.

chown cyrus:mail /home/cyrus

chown cyrus:mail /home/cyrus/procmailrc.

Copy /mail-impl/scripts/fedora/cyrus/cyrus_adduser to /usr/bin/cyrus_adduser. Edit cyrus_adduser and replace localhost with the mail server's host name. Make sure the script can be executed.

chmod u+x /usr/bin/cyrus_adduser.

Copy /mail-impl/scripts/fedora/cyrus/cyrus_userdel to /usr/bin/cyrus_userdel. Edit cyrus_userdel and replace localhost with the mail server's host name. Make sure the script can be executed.

chmod u+x /usr/bin/cyrus_userdel.

Note

If you copy cyrus_adduser and cyrus_userdel from a Windows environment to a Linux environment, you need to run dos2unix cyrus_adduser to convert the file so that Linux can read the file correctly.

10. Edit /etc/pam.d/pop so that POP authentication is checked via MySQL. Remove the current lines and add these lines:

```
##PAM-1.0
auth sufficient pam_mysql.so user=dbuser passwd=dbpassword
host=127.0.0.1 db=cyrus
table=CyrusUser usercolumn=userId passwdcolumn=password_ crypt=0

account required pam_mysql.so user=dbuser passwd=dbpassword
host=127.0.0.1 db=cyrus
table=CyrusUser usercolumn=userId passwdcolumn=password_ crypt=0
```

Edit /etc/pam.d/imap so that IMAP authentication is checked via MySQL. Remove the current lines and add these lines:

```
##PAM-1.0
auth sufficient pam_mysql.so user=dbuser passwd=dbpassword
host=127.0.0.1 db=cyrus
table=CyrusUser usercolumn=userId passwdcolumn=password_ crypt=0

account required pam_mysql.so user=dbuser passwd=dbpassword
host=127.0.0.1 db=cyrus
table=CyrusUser usercolumn=userId passwdcolumn=password_ crypt=0
```

11. Turn on MySQL.

chkconfig --level 3 mysqld on

/etc/rc.d/init.d/mysqld start

Configure MySQL so that it can be accessed by the username dbuser and password dbpassword.

```
use mysql;
```

```
insert into user values ('127.0.0.1', "dbuser", password("dbpassword"), "Y", "Y", "Y", "Y",  
"Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y");
```

Create the database and tables that will be used to authenticate IMAP users.

```
create database cyrus;
```

```
use cyrus;
```

```
create table CyrusUser ( userId varchar(75) not null primary key, password_ varchar(75) not  
null );
```

```
create table CyrusVirtual ( emailAddress varchar(75) not null primary key, userId varchar(75)  
not null );
```

The Expect scripts `cyrus_adduser` and `cyrus_userdel` that are used to add and delete Cyrus IMAP users require a default cyrus user to authenticate with.

```
insert into CyrusUser (userId, password_) values ('cyrus', 'cyrus_password');
```

Every virtual domain requires a `postmaster@yourdomain.com` entry so that email to `postmaster@yourdomain.com` can be delivered to a Cyrus IMAP account.

```
insert into CyrusVirtual (emailAddress, userId) values ('postmaster@yourdomain.com',  
'your_domain_1');
```

Create a default account for `your_domain_1`.

```
insert into CyrusUser (userId, password_) values ('your_domain_1', 'your_password');
```

```
insert into CyrusVirtual (emailAddress, userId) values ('joe.blogs@yourdomain.com',  
'your_domain_1');
```

```
quit;
```

```
cyrus_adduser cyrus_password your_domain_1
```

12. Turn on SpamAssassin.

```
chkconfig --level 3 spamassassin on
```

```
/etc/rc.d/init.d/spamassassin start
```

13. Download ClamAV [<http://crash.fce.vutbr.cz/crash-hat/3/clamav/clamav-0.86.1-1.i386.rpm>].

Install ClamAV.

```
rpm -i clamav-0.86.1-1.i386.rpm
```

Turn on ClamAV.

```
chkconfig --level 3 clamd on
```

```
/etc/rc.d/init.d/clamd start
```

Download ClamAssassin [<http://drivel.com/clamassassin/clamassassin-1.2.2.tar.gz>].

Install ClamAssassin.

```
gunzip clamassassin-1.2.2.tar.gz
```

```
tar xvf clamassassin-1.2.2.tar
```

```
cd clamassassin-1.2.2
```

```
./configure
```

```
cp clamassassin /usr/local/bin
```

Edit /usr/local/bin/clamassassin.

Set SUBJECTHEAD to "[VIRUS]".

14. Copy /mail-impl/scripts/fedora/cyrus/procmail_vacation to /usr/local/bin/procmail_vacation. Make sure the script can be executed.

```
chmod u+x /usr/local/bin/procmail_vacation.
```

Download SendEmail [<http://caspien.dotconf.net/menu/Software/SendEmail/sendEmail-v1.52.tar.gz>].

Install SendEmail.

```
gunzip sendEmail-v1.52.tar.gz
```

```
tar xvf sendEmail-v1.52.tar
```

```
cd sendEmail-v1.52
```

```
chmod u+x sendEmail
```

```
chown cyrus:mail sendEmail
```

```
cp sendEmail /usr/local/bin
```

15. Download the source distribution of Pop-before-smtp [<http://sourceforge.net/projects/popbsmtp>].

Pop-before-smtp requires perl-TimeDate and perl-Net-Netmask.

Install perl-TimeDate from the distributed RPM.

Install perl-Net-Netmask.

```
perl -MCPAN -e 'install Net::Netmask'
```

Install Pop-before-smtp.

```
gunzip pop-before-smtp-1.38.tar.gz
```

```
tar xvf pop-before-smtp-1.38.tar
```

```
cd pop-before-smtp-1.38
```

```
chown root:root *
```

```
cp pop-before-smtp.init /etc/rc.d/init.d/pop-before-smtp
```

```
cp pop-before-smtp /usr/sbin/
```

```
cp pop-before-smtp-conf.pl /etc
```

Edit /etc/pop-before-smtp-conf.pl by uncommenting and modifying certain sections so it matches the following information.

```
$dbfile = '/etc/postfix/pop-before-smtp';
$grace = 120*60;

# Set the log file we will watch for pop3d/imapd records.
$file_tail{'name'} = '/var/log/maillog';

# For Cyrus (including a tweak for IP adrrs that don't resolve):
$pat = '^(... .. ..:..:..) \S+ (?:pop3d|imapd)\[\d+\]: ' .
```

Turn on Pop-before-smtp.

```
chkconfig --level 3 pop-before-smtp on
```

```
/etc/rc.d/init.d/pop-before-smtp start
```

16.Restart your mail server.

```
shutdown -r now
```

1.3. Dovecot+Postfix

First build a generic Liferay email hook, ShellHook.java, that shells out all of the email methods. You install it by adding these lines to portal-ext.properties:

```
mail.hook.impl=com.liferay.mail.util.ShellHook
```

```
mail.hook.shell.script=/usr/sbin/mailadmin.ksh
```

```
mail.box.style=INBOX
```

We next built a generic Korn Shell Script, mailadmin.ksh, that implements each method for Dovecot, or any other email system you want. It supports an interactive command line interface for testing:

```
mailadmin.ksh --help
```

```
mailadmin.ksh
```

```
mailadmin.ksh addForward [userId] [emailAddresses]
```

```
mailadmin.ksh addUser [userId] [password] [firstName] [middleName]  
[lastName] [emailAddress]
```

```
mailadmin.ksh addVacationMessage [userId] [emailAddress]  
[vacationMessage]
```

```
mailadmin.ksh deleteEmailAddress [userId]
```

```
mailadmin.ksh deleteUser [userId]
mailadmin.ksh updateBlocked [userId] [blockedEmailAddress]
mailadmin.ksh updateEmailAddress [userId] [emailAddress]
mailadmin.ksh updatePassword [userId] [password]
```

All of the code is in SVN. mailadmin is at: mail-impl/scripts/fedora/ksh/mailadmin.ksh Here are the step-by-step installation instructions:

```
# Edit SASL-auth authentication to use MySQL with the Postfix setup

vi /etc/pam.d/smtpl
#%PAM-1.0
auth sufficient pam_mysql.so user=DBUSR passwd=DBPASSWD host=127.0.0.1
db=mail table=postfix_users usercolumn=email passwdcolumn=clear
crypt=0
account required pam_mysql.so user=DBUSR passwd=DBPASSWD
host=127.0.0.1 db=mail table=postfix_users usercolumn=email
passwdcolumn=clear crypt=0

# CONFIGURE VMAIL USER AND EMAIL PATHS

groupadd -g 510 vmail
useradd -u 510 -g vmail vmail
mkdir -p /var/vmail/EMAILDOMAIN
chown -R vmail:vmail /var/vmail
chmod -R 770 /var/vmail

# Add vmail user to tomcat group and tomcat user to vmail group
# Note the vmail uid, 510, is inserted into the postfix_users table
below
vi /etc/group
tomcat:x:500:vmail
vmail:x:510:tomcat

# CONFIGURE MYSQL

# Add DBUSR to MySql database for managing email tables

mysql -u root -p
use mysql;
insert into user values ('127.0.0.1', "DBUSR",
old_password("DBPASSWD"), "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y",
"Y", "Y", "Y", "Y", "Y");
commit;
quit

# Login as email user and build email database, "mail", and postfix
tables
mysql -u DBUSR -p
```

```

create database mail;
use mail;

CREATE TABLE postfix_alias (
  id int(11) unsigned NOT NULL auto_increment,
  alias varchar(128) NOT NULL default '',
  destination varchar(128) NOT NULL default '',
  PRIMARY KEY (id)
) TYPE=MyISAM;

CREATE TABLE postfix_relocated (
  id int(11) unsigned NOT NULL auto_increment,
  email varchar(128) NOT NULL default '',
  destination varchar(128) NOT NULL default '',
  PRIMARY KEY (id)
) TYPE=MyISAM;

CREATE TABLE postfix_transport (
  id int(11) unsigned NOT NULL auto_increment,
  domain varchar(128) NOT NULL default '',
  destination varchar(128) NOT NULL default '',
  PRIMARY KEY (id),
  UNIQUE KEY domain (domain)
) TYPE=MyISAM;

CREATE TABLE postfix_users (
  id int(11) unsigned NOT NULL auto_increment,
  email varchar(128) NOT NULL default '',
  clear varchar(128) NOT NULL default '',
  crypt varchar(128) NOT NULL default '',
  name tinytext NOT NULL,
  uid int(11) unsigned NOT NULL default '1004',
  gid int(11) unsigned NOT NULL default '1004',
  homedir tinytext NOT NULL,
  maildir tinytext NOT NULL,
  quota tinytext NOT NULL,
  access enum('Y','N') NOT NULL default 'Y',
  postfix enum('Y','N') NOT NULL default 'Y',
  PRIMARY KEY (id),
  UNIQUE KEY email (email)
) TYPE=MyISAM;

CREATE TABLE postfix_virtual (
  id int(11) unsigned NOT NULL auto_increment,
  email varchar(128) NOT NULL default '',
  destination varchar(128) NOT NULL default '',
  PRIMARY KEY (id)
) TYPE=MyISAM;

CREATE TABLE postfix_access (
  id int(10) unsigned NOT NULL auto_increment,
  source varchar(128) NOT NULL default '',
  access varchar(128) NOT NULL default '',
  type enum('recipient','sender','client') NOT NULL default
'recipient',
  PRIMARY KEY (id)
) TYPE=MyISAM

commit;

# Add an email domain
INSERT INTO `postfix_transport` VALUES
(3,'EMAILDOMAIN','virtual:');

```

```
# Add an email user (automated by Liferay using ShellHook, and
mailadmin.ksh)
# vmail uid is 510
INSERT INTO `postfix_users` VALUES
(17,'LIFERAYUSR@EMAILDOMAIN','LIFERAYPWD','','',510,510,'/var/vmail','EMAILDOMAIN/
# Add an email forward
INSERT INTO `postfix_virtual` VALUES
(27,'LIFERAYLOGIN','LIFERAYUSR@EMAILDOMAIN');

commit;
quit
```

```
# CONFIGURE POSTFIX
```

```
cd /etc/postfix/
rm -rf ssl/
rm -rf sasl/
```

```
vi /etc/postfix/mysql-aliases.cf
user = DBUSR
password = DBPASSWD
dbname = mail
table = postfix_alias
select_field = destination
where_field = alias
hosts = 127.0.0.1
```

```
vi /etc/postfix/mysql-client.cf
user = DBUSR
password = DBPASSWD
dbname = mail
table = postfix_access
select_field = access
where_field = source
additional_conditions = and type = 'client'
hosts = 127.0.0.1
```

```
vi /etc/postfix/mysql-recipient.cf
user = DBUSR
password = DBPASSWD
dbname = mail
table = postfix_access
select_field = access
where_field = source
additional_conditions = and type = 'recipient'
hosts = 127.0.0.1
```

```
vi /etc/postfix/mysql-relocated.cf
user = DBUSR
password = DBPASSWD
dbname = mail
table = postfix_relocated
select_field = destination
where_field = email
hosts = 127.0.0.1
```

```
vi /etc/postfix/mysql-sender.cf
user = DBUSR
password = DBPASSWD
dbname = mail
table = postfix_access
select_field = access
where_field = source
additional_conditions = and type = 'sender'
hosts = 127.0.0.1

vi /etc/postfix/mysql-transport.cf
user = DBUSR
password = DBPASSWD
dbname = mail
table = postfix_transport
select_field = destination
where_field = domain
hosts = 127.0.0.1

vi /etc/postfix/mysql-virtual-gid.cf
user = DBUSR
password = DBPASSWD
dbname = mail
table = postfix_users
select_field = gid
where_field = email
additional_conditions = and postfix = 'y'
hosts = 127.0.0.1

vi /etc/postfix/mysql-virtual-maps.cf
user = DBUSR
password = DBPASSWD
dbname = mail
table = postfix_users
select_field = maildir
where_field = email
additional_conditions = and postfix = 'y'
hosts = 127.0.0.1

vi /etc/postfix/mysql-virtual-uid.cf
user = DBUSR
password = DBPASSWD
dbname = mail
table = postfix_users
select_field = uid
where_field = email
additional_conditions = and postfix = 'y'
hosts = 127.0.0.1

vi /etc/postfix/mysql-virtual.cf
user = DBUSR
password = DBPASSWD
dbname = mail
table = postfix_virtual
select_field = destination
where_field = email
hosts = 127.0.0.1

chmod 640          /etc/postfix/mysql-*
chown root:postfix /etc/postfix/mysql-*
```



```

vi /etc/postfix/main.cf
# see /usr/share/postfix/main.cf.dist for a commented, fuller version
of this file.
# Do not change these directory settings - they are critical to
Postfix operation.
command_directory = /usr/sbin
daemon_directory = /usr/libexec/postfix
program_directory = /usr/libexec/postfix
smtpd_banner = $myhostname ESMTP $mail_name
setgid_group = postdrop
biff = no
append_dot_mydomain = no
myhostname = EMAILDOMAIN
myorigin = $myhostname
mydestination = EMAILDOMAIN, $transport_maps
relayhost =
mynetworks = 127.0.0.0/8
mailbox_command =
mailbox_size_limit = 0
recipient_delimiter = +
smtpd_sasl_auth_enable = yes
smtpd_recipient_restrictions = permit_sasl_authenticated,
permit_mynetworks, reject_unauth_destination
smtpd_sasl_security_options = noanonymous
smtpd_sasl_local_domain = $myhostname
broken_sasl_auth_clients = yes
smtpd_recipient_restrictions = permit_mynetworks,
permit_sasl_authenticated, check_recipient_access
mysql:/etc/postfix/mysql-recipient.cf, reject_unauth_destination,
permit
smtpd_sender_restrictions = check_sender_access
mysql:/etc/postfix/mysql-sender.cf
smtpd_client_restrictions = check_client_access
mysql:/etc/postfix/mysql-client.cf
alias_maps = mysql:/etc/postfix/mysql-aliases.cf
relocated_maps = mysql:/etc/postfix/mysql-relocated.cf
transport_maps = mysql:/etc/postfix/mysql-transport.cf
virtual_maps = mysql:/etc/postfix/mysql-virtual.cf
virtual_mailbox_base = /var/vmail
virtual_mailbox_maps = mysql:/etc/postfix/mysql-virtual-maps.cf
virtual_uid_maps = mysql:/etc/postfix/mysql-virtual-uid.cf
virtual_gid_maps = mysql:/etc/postfix/mysql-virtual-gid.cf
local_recipient_maps = $alias_maps $virtual_mailbox_maps

```

```

chmod 644 /etc/postfix/main.cf
chown root:root /etc/postfix/main.cf

```

```

vi /etc/postfix/master.cf
smtp inet n - n - - smtpd
pickup fifo n - n 60 1 pickup
cleanup unix n - n - 0 cleanup
qmgr fifo n - n 300 1 qmgr
rewrite unix - - n - -
trivial-rewrite
bounce unix - - n - 0 bounce
defer unix - - n - 0 bounce
trace unix - - n - 0 bounce
verify unix - - n - 1 verify
flush unix n - n 1000? 0 flush
proxymap unix - - n - - proxymap

```

```
smtp      unix  -   -   n   -   -   smtp
relay    unix  -   -   n   -   -   smtp
showq    unix  n   -   n   -   -   showq
error    unix  -   -   n   -   -   error
local    unix  -   n   n   -   -   local
virtual  unix  -   n   n   -   -   virtual
lmtp     unix  -   -   n   -   -   lmtp
anvil    unix  -   -   n   -   1   anvil
maildrop unix  -   n   n   -   -   pipe
 flags=DRhu user=vmail argv=/usr/local/bin/maildrop -d ${recipient}
old-cyrus unix  -   n   n   -   -   pipe
 flags=R user=cyrus argv=/usr/lib/cyrus-imapd/deliver -e -m
${extension} ${user}
cyrus    unix  -   n   n   -   -   pipe
 user=cyrus argv=/usr/lib/cyrus-imapd/deliver -e -r ${sender} -m
${extension} ${user}
uucp     unix  -   n   n   -   -   pipe
 flags=Fqhu user=uucp argv=uux -r -n -z -a$sender -
$nexthop!rmail postfix ($recipient)
ifmail   unix  -   n   n   -   -   pipe
 flags=F user=ftn argv=/usr/lib/ifmail/ifmail -r $nexthop ($recipient)
bsmtp    unix  -   n   n   -   -   pipe
 flags=Fq. user=foo argv=/usr/local/sbin/bsmtp -f $sender $nexthop
$recipient
procmail unix  -   n   n   -   -   pipe
 flags=R user=cyrus argv=/usr/bin/procmail -t -m USER=${user}
EXTENSION=${extension} /home/cyrus/procmailrc
```

```
chmod 644      /etc/postfix/master.cf
chown root:root /etc/postfix/master.cf
```

CONFIGURE DOVECOT

```
cd
wget
http://dag.wieers.com/packages/dovecot/dovecot-0.99.13-1.1.el3.rf.i386.rpm
apt-get install rh-postgresql-libs
rpm -Uvh dovecot-0.99.13-1.1.el3.rf.i386.rpm
```

```
vi /etc/dovecot.conf
protocols = imaps pop3s imap pop3
ssl_disable = yes
ssl_cert_file = /etc/ssl/certs/dovecot.pem
ssl_key_file = /etc/ssl/private/dovecot.pem
login = imap
login_executable = /usr/libexec/dovecot/imap-login
login = pop3
login_executable = /usr/libexec/dovecot/pop3-login
mail_extra_groups = mail
default_mail_env = maildir:/var/vmail/%d/%n/Maildir
imap_executable = /usr/libexec/dovecot/imap
pop3_executable = /usr/libexec/dovecot/pop3
auth = default
auth_mechanisms = plain
auth_default_realm = EMAILDOMAIN
auth_userdb = mysql /etc/dovecot-mysql.conf
```

```

auth_passdb = mysql /etc/dovecot-mysql.conf
auth_user = root
auth_verbose = yes

vi /etc/dovecot-mysql.conf
db_host = 127.0.0.1
db_port = 3306
db = mail
db_user = DBUSR
db_passwd = DBPASSWD
db_client_flags = 0
default_pass_scheme = PLAIN
password_query = SELECT clear FROM postfix_users WHERE email = '%n%d'
or email = '%n@EMAILDOMAIN'
user_query = SELECT maildir, uid, gid FROM postfix_users WHERE email =
'%n%d' or email = '%n@EMAILDOMAIN'

# CONFIGURE LIFERAY

# configure mailadmin.ksh
cp mailadmin.ksh /usr/sbin
vi /usr/sbin/mailadmin.ksh
DOMAIN=EMAILDOMAIN # Domain being managed
MYSQL_USERNAME=DBUSR # MySQL user
MYSQL_PASSWORD=DBPASSWD # MySQL password
TOMCAT_UID=500 # Mail File Creation user id -
tomcat
VMAIL_GID=510 # Mail File Creation group id -
vmail

chmod 750 /usr/sbin/mailadmin.ksh
chown tomcat:tomcat /usr/sbin/mailadmin.ksh

# create mailadmin log file
touch /var/log/mailadmin.log
chmod 660 /var/log/mailadmin.log
chown tomcat:tomcat /var/log/mailadmin.log

# configure Liferay to use mailadmin.ksh
vi /usr/local/tomcat/common/classes/portal-ext.properties
mail.hook.impl=com.liferay.mail.util.ShellHook
mail.hook.shell.script=/usr/sbin/mailadmin.ksh
mail.box.style=INBOX

# update these JARs with latest from SVN HEAD
/usr/local/tomcat/common/lib/ext/mail-impl.jar -> add
com.liferay.mail.util.ShellHook.class
/usr/local/tomcat/common/lib/ext/portal-impl.jar -> update
com.liferay.portal.util.PropsUtil.class
/usr/local/tomcat/common/lib/ext/portal-impl.jar -> update
com.liferay.util.StringUtil.class

# Configure Tomcat

#add mail/MailSession settings
vi /usr/local/tomcat/conf/Catalina/localhost/liferay.xml
<parameter>

```

```
        <name>mail.smtp.host</name>
        <value>localhost</value>
</parameter>
<parameter>
        <name>mail.imap.host</name>
        <value>localhost</value>
</parameter>
<parameter>
        <name>mail.store.protocol</name>
        <value>imap</value>
</parameter>
<parameter>
        <name>mail.transport.protocol</name>
        <value>smtp</value>
</parameter>
<parameter>
        <name>mail.pop3.host</name>
        <value>localhost</value>
</parameter>
```

```
# Enable autostart on reboots
```

```
chkconfig postfix on
chkconfig dovecot on
```

```
# Verify Install Commands
```

```
tail -f 50 /var/log/maillog
tail -f 50 /var/log/messages
```

```
# Restart saslauthd before Postfix, so that Postfix doesn't start with
# a bad SASL setup, otherwise it doesn't answer smtp requests
/etc/init.d/saslauthd restart
```

```
# make sure saslauthd restarts
ps -ef | grep saslauthd | grep -v grep
```

```
# make sure postfix restarts
/etc/init.d/postfix restart
ps -ef | grep postfix | grep -v grep
```

```
# make sure dovecot restarts
/etc/init.d/dovecot restart
ps -ef | grep dovecot | grep -v grep
```

```
reboot
```

```
# make sure everything starts
ps -ef | grep postfix | grep -v grep
```

```
ps -ef | grep dovecot | grep -v grep
ps -ef | grep saslauthd | grep -v grep
```

```
# Test SMTP by sending an email to LIFERAYUSR@EMAILDOMAIN
```

```
telnet localhost 25
EHLO EMAILDOMAIN
MAIL FROM:test@test.com
RCPT TO:LIFERAYUSR@EMAILDOMAIN
DATA
Test msg
.

quit
```

```
# Test SMTP by sending an email to alias LIFERAYLOGIN
```

```
telnet localhost 25
EHLO EMAILDOMAIN
MAIL FROM:test@test.com
RCPT TO:LIFERAYLOGIN
DATA
Test msg
.

quit
```

```
#TEST IMAP by logging in as LIFERAYUSR@EMAILDOMAIN
```

```
telnet localhost imap
x LOGIN LIFERAYUSR@EMAILDOMAIN LIFERAYPWD
x STATUS "INBOX" (MESSAGES)
x SELECT "INBOX"
x FETCH 1 BODY[HEADER]
x LOGOUT
```

```
# Test using usedId without a Domain name
```

```
telnet localhost imap
x LOGIN LIFERAYUSR LIFERAYPWD
x STATUS "INBOX" (MESSAGES)
x SELECT "INBOX"
x FETCH 1 BODY[HEADER]
x LOGOUT
```

1.4. Microsoft Exchange

Coming soon...

2. LDAP Integration

In this section you will learn about LDAP integration. The initial set of instructions will guide you through the installation of the Apache directory server and an LDAP browser. The instructions will then

guide you to input a user into the LDAP browser. After the user has been entered into the browser the user will be integrated with Liferay Portal.

2.1. Installing Apache Directory Server

1. Go to www.apache.org [<http://www.apache.org>].
2. Click **Directory**.
3. Click **Download**.
4. Click on the suggested mirror site for download.
5. Click **apacheds>stable>1.0>1.0 RC3>apacheds-1.0-RC3-win32-setup.exe**
6. **Save** file.
7. Click on the **Apache** icon and follow the installation instructions.
8. Click **Start**.

2.2. Installing LDAP Browser

1. Go to www.jxplorer.org [<http://www.jxplorer.org>].
2. Click **Downloads>precompiled java package>Windows** platform.
3. **Save** file.
4. Click on the LDAP browser icon and follow the installation instructions.

2.3. Inputting User in LDAP Browser

1. Open the LDAP browser.
2. Click **File>Connect**.

Open LDAP/DSML Connection

Host: localhost Port: 389

Protocol: LDAP v3

DSML Service:

Optional Values

Base DN:

Security

Level: Anonymous

User DN:

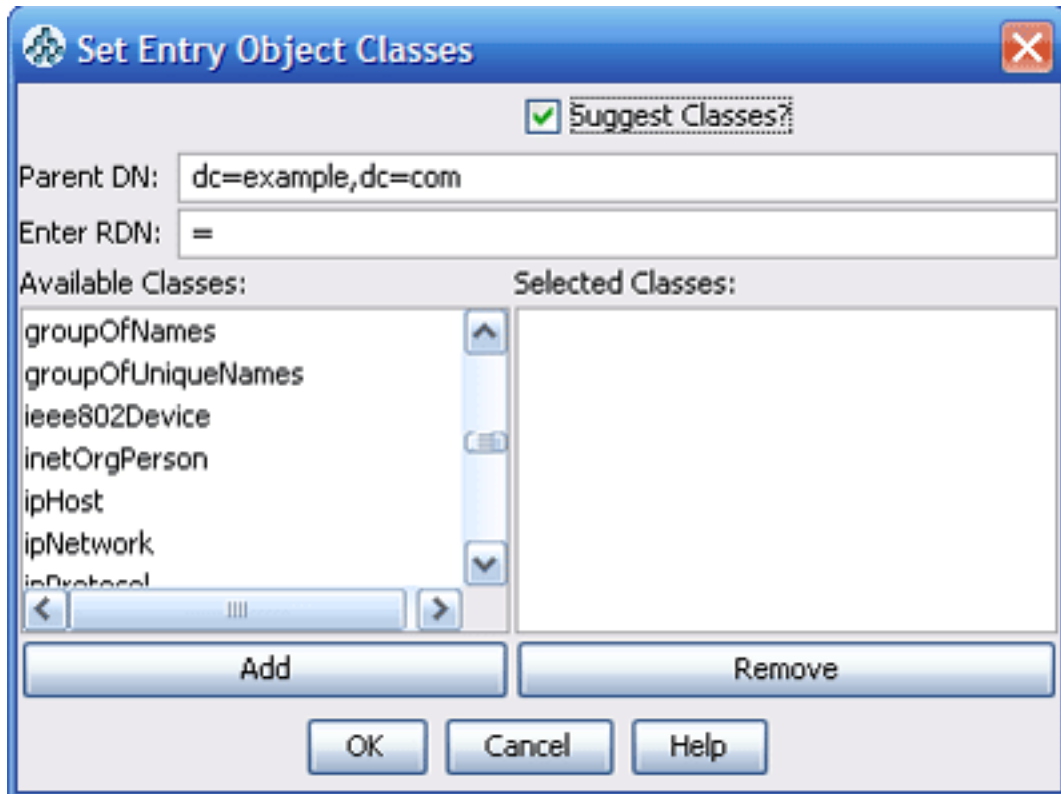
Password:

Use a Template

Save default Delete Default

OK Cancel Help

3. Change the port to 10389.
4. In the **Level** drop-down menu, choose **User+Password**.
5. Insert `uid=admin,ou=system` in the User DN input field.
6. The password is **secret**.
7. Click **Save** and enter a name for the template.
8. Right click on **Example** and click **New**.



9. Add **inetorgperson** to the Selected Class.

10. User Jane Smith will be added. Enter **cn=janesmith** in the Enter RDN field and click **OK**.

attribute type	value
cn	janesmith
objectClass	inetOrgPerson
objectClass	organizationalPerson
objectClass	person
objectClass	top
sn	
audio	
businessCategory	
carLicense	
departmentNumber	
description	
destinationIndicator	
displayName	
employeeNumber	
employeeType	
facsimileTelephoneNumber	
givenName	
homePhone	

11. In the Table Editor enter **Smith** in the **SN** line.

12. Enter **Jane** in the givenName line.

13. For the mail enter janesmith@liferay.com.

14. For the userpassword enter **test**.

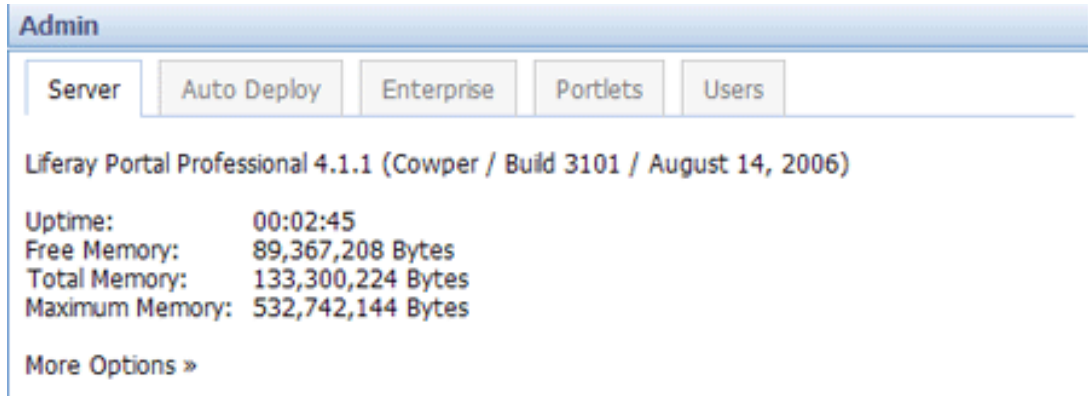
15. Click **Submit**.

2.4. Integration

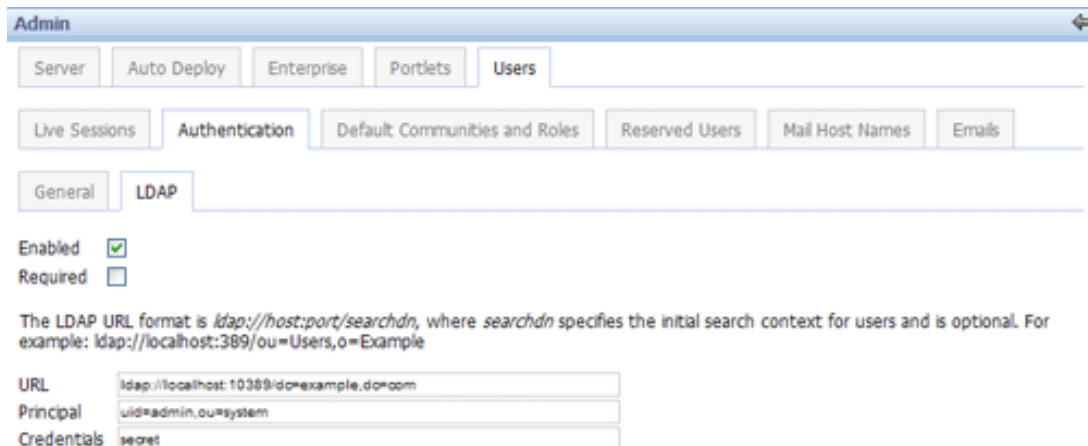
1. With user Jane Smith entered into the LDAP browser, the user will now be integrated with Liferay

Portal. Begin by logging into Liferay Portal as the Administrator.

2. The login is **test@liferay.com** and the password is **test**.
3. Currently, Jane Smith's profile exists only on the LDAP browser. To integrate her information into Liferay Portal, click on the **Users tab** in the Admin portlet.



4. Click **Authentication**.
5. Click **LDAP**.
6. Check the **Enabled** box.



7. If the **Required** box is checked only users in the LDAP server will be able to log into Liferay Portal. For this demonstration leave the box unchecked.
8. Liferay Portal supports other directory servers in addition to the ones provided. The Apache Directory Server, Microsoft Active Directory Server, and Novell eDirectory comes preconfigured.

(mail=@email_address@)

If the user is valid and the user exists in the LDAP server but not in Liferay, the user will be synchronized from the LDAP server to Liferay. Below is a mapping of Liferay attributes and the pair name used to populate the Liferay field from LDAP.

userId=cn
 password=userPassword
 emailAddress=mail
 firstName=givenName
 lastName=surname
 jobTitle=title

9. Select the **Apache Directory Server** and click **Save**.

10. Once Jane Smith logs in to her account on Liferay Portal and agrees to the terms of use, her user information will be added to Liferay Portal. To demonstrate this, assume that Jane Smith has logged into her account. While logged in as the Administrator, a search for Jane Smith will show that the user has been integrated into Liferay Portal.

Enterprise Admin

Users Organizations Locations User Groups Roles

First Name Middle Name Last Name Email Address Active
 jane Yes

<input type="checkbox"/>	Name	Job Title	Location	City	Region	Country
<input type="checkbox"/>	Jane Smith	title				

3. Chat Portlet

Note

Windows Firewall must be turned off for the Chat Portlet to run properly

1. Create a folder and name it Wildfire.
2. Download the Wildfire Server at www.jivesoftware.com. It is recommended that you download the zip file. If you do not have Java JRE on your machine you must use the .exe file.
3. Extract the file to the Wildfire folder.
4. Open the Wildfire folder and click on **bin**.
5. Click on **wildfire.exe**.

6. Click **Launch Admin**.
7. Proceed with configuring Wildfire based on your requirements.
8. To configure Wildfire with Liferay, open **portal-ext.properties** (\webapps\Root\WEB-INF\classes).
9. Enter the following:

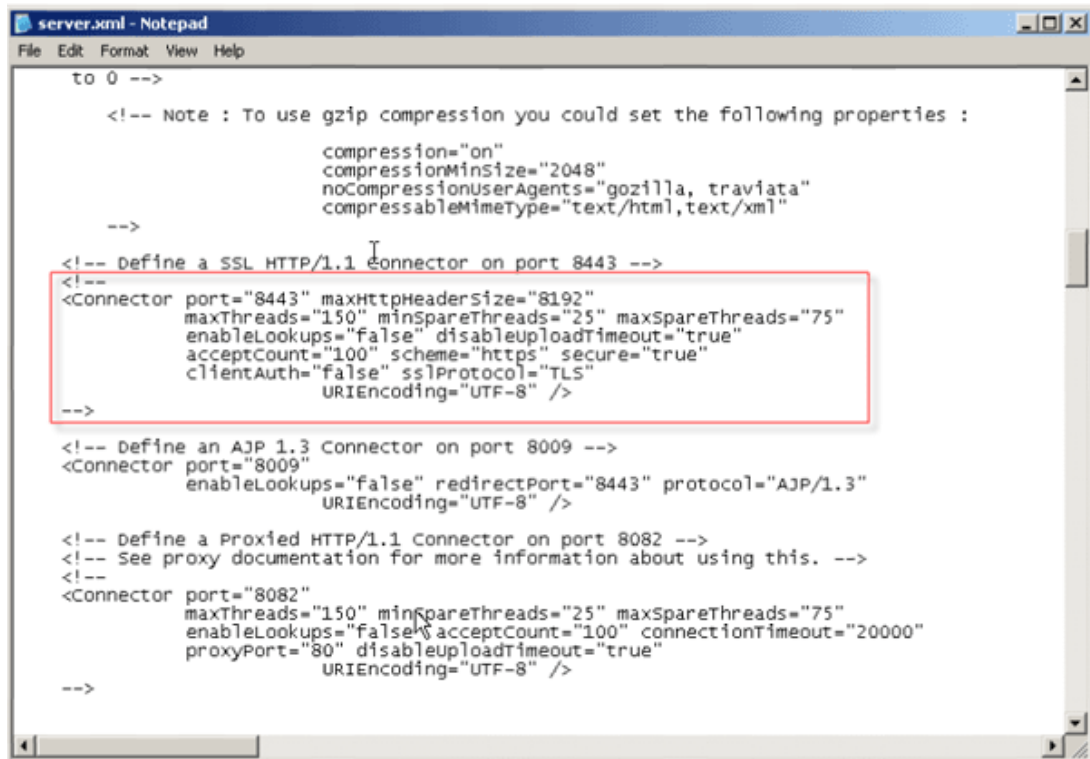

```
jabber.xmpp.server.enabled=true
reverse.ajax.enabled=true
```
10. **Save** the changes.
11. To view the Chat Portlet go to Liferay Portal and sign in as a user.
12. Add the Chat Portlet to you desktop.

4. CAS Server

Note

In a production environment the CAS server should run on its own Tomcat instance but for purposes of this demonstration we will drop it in the same instance as Liferay Portal.

1. Go to the server.xml file and uncomment the SSL section to open port 8443.



```
server.xml - Notepad
File Edit Format View Help

to 0 -->

<!-- Note : To use gzip compression you could set the following properties :

        compression="on"
        compressionMinSize="2048"
        noCompressionUserAgents="gozilla, traviata"
        compressableMimeType="text/html,text/xml"

-->

<!-- Define a SSL HTTP/1.1 Connector on port 8443 -->
<!--
<Connector port="8443" maxHttpHeaderSize="8192"
maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
enableLookups="false" disableUploadTimeout="true"
acceptCount="100" scheme="https" secure="true"
clientAuth="false" sslProtocol="TLS"
URIEncoding="UTF-8" />
-->

<!-- Define an AJP 1.3 Connector on port 8009 -->
<Connector port="8009"
enableLookups="false" redirectPort="8443" protocol="AJP/1.3"
URIEncoding="UTF-8" />

<!-- Define a Proxied HTTP/1.1 Connector on port 8082 -->
<!-- See proxy documentation for more information about using this. -->
<!--
<Connector port="8082"
maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
enableLookups="false" acceptCount="100" connectionTimeout="20000"
proxyPort="80" disableUploadTimeout="true"
URIEncoding="UTF-8" />
-->
```

2. **Save**.

3. Download the latest available version of the Liferay-portal-cas-4.3.x.war file from www.liferay.com/web/guest/downloads.
4. Rename the file to cas-web.war.
5. Copy the war file and paste it to the Tomcat webapps directory.
6. To download the Yale CAS Client 2.0.11 go to <http://www.ja-sig.org/products/cas/downloads/index.html> [???].
7. Create a folder and extract the cas.client zip file into this folder.
8. Open the folder and navigate to the casclient jar file (cas-client-2.0.11\java\lib).
9. Copy the file.
10. Go to the lib file in Tomcat (webapps\ROOT\WEB_INF\lib) and paste the casclient jar file to replace the existing one.
11. In the command prompt go to the ROOT directory.
12. Enter the following:

```
keytool -genkey -alias tomcat -keypass changeit -keyalg RSA
```

13. Enter **changeit** for the password.
14. Answer the list of questions. Note that the first and last name must be the host name of your server and cannot be an IP address. This is very important because an IP address will fail client hostname verification even if it is correct.

```

C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\liferay>cd\
C:\>cd tomcat\webapps\ROOT
C:\tomcat\webapps\ROOT>keytool -genkey -alias tomcat -keypass changeit -keyalg RSA
Enter keystore password: changeit
What is your first and last name?
  [Unknown]: localhost
What is the name of your organizational unit?
  [Unknown]:
What is the name of your organization?
  [Unknown]:
What is the name of your City or Locality?
  [Unknown]:
What is the name of your State or Province?
  [Unknown]:
What is the two-letter country code for this unit?
  [Unknown]:
Is CN=localhost, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown correct?
  [no]: yes

C:\tomcat\webapps\ROOT>_
    
```

15. Enter the following command to export the cert you generated from your personal keystore:

```
keytool -export -alias tomcat -keypass changeit -file server.cert
```

16. Enter **changeit** for the password.
17. Import the cert into Java's keystore with this command:

```
keytool -import -alias tomcat -file %FILE_NAME% -keypass changeit
```

-keystore %JAVA_HOME%/jre/lib/security/cacerts

18. Enter **changeit** for the password.

19. Enter **yes**.

```

C:\WINDOWS\system32\cmd.exe
What is the name of your organization?
[Unknown]:
What is the name of your City or Locality?
[Unknown]:
What is the name of your State or Province?
[Unknown]:
What is the two-letter country code for this unit?
[Unknown]:
Is CN=localhost, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown correct?
[no]: yes

C:\tomcat\webapps\ROOT>keytool -export -alias tomcat -keystore changeit -file server.cert
Enter keystore password: changeit
Certificate stored in file <server.cert>

C:\tomcat\webapps\ROOT>keytool -import -alias tomcat -file server.cert -keystore %
JAVA_HOME%/jre/lib/security/cacerts
Enter keystore password: changeit
Owner: CN=localhost, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown
Issuer: CN=localhost, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown
Serial number: 45897b1c
Valid from: Wed Dec 20 10:04:12 PST 2006 until: Tue Mar 20 11:04:12 PDT 2007
Certificate fingerprints:
MD5: 6A:30:7C:9F:AC:47:1F:2D:61:81:DF:D2:51:F8:AD:17
SHA1: 60:44:07:6A:93:8E:FC:CF:CC:3D:FE:50:D0:11:4E:40:E9:D8:68:1E
Trust this certificate? [no]: yes
Certificate was added to keystore

C:\tomcat\webapps\ROOT>
    
```

20. To set up Liferay Portal, navigate to the classes file in Tomcat (webapps\ROOT\WEB_INF\classes).

21. Create portal-ext.properties if it does not exist yet, somewhere in the server classpath:

22. Open portal-ext.properties and enter:

```
cas.auth.enabled=true
```

23. If necessary assign appropriate values to the following properties (default values shown below):

```

#
# A user may be authenticated from CAS and not yet exist in the
# portal. Set
# this to true to automatically import users from LDAP if they do
# not exist
# in the portal.
#
cas.import.from.ldap=false

#
# Set the default values for the required CAS URLs.
#
cas.login.url=https://localhost:8443/cas-web/login
cas.logout.url=https://localhost:8443/cas-web/logout
cas.service.url=http://localhost:8080/c/portal/login
cas.validate.url=https://localhost:8443/cas-web/proxyValidate
    
```

24. **Save**.

25. Start Tomcat and go to Liferay Portal.

26. Click **Sign In**. If everything is set up correctly you will be redirected to the CAS server's login screen.

5. Installation of Workflow services

The workflow services allows a user to define through the workflow portlet any number of simple to complex business processes/workflows, deploy them, and manage them through a portal interface. The power of this portlet is that it allows users to create forms-based data entry applications that have knowledge of users, groups, and roles without writing a single line of code – it only requires creation of a single XML document. The portlet relies on Apache ServiceMix to function as an Enterprise Service Bus that acts as a broker between the portal and a workflow engine. Essentially, the portal provides a generic interface through which workflow services are requested via normal HTTP calls. The requests are routed through ServiceMix which in turn calls a workflow engine implementation that the user has defined in the ServiceMix configuration. By default, Liferay provides an implementation of JBoss' jBPM workflow engine (version 3.1.2).

The next section provides detailed installation instructions for setting up the environment for using the Workflow portlet. The instructions are for Tomcat, but can be easily adapted for any application server. After installing, refer to the user guides for more information on how to use the workflow portlet to create custom business processes.

5.1. Installation

Because the default implementation of the workflow portlet depends on both ServiceMix and jBPM, the installation requires more than just the normal portal bundle. The following provides detailed instructions for deploying the workflow portlet and testing it with a simple process.

1. It is assumed that Liferay Portal has already been installed following one of the procedures explained earlier. If it's currently running stop it.
2. Download the latest version of liferay-portal-servicemix-4.3.x.war and liferay-portal-jbpm-4.3.x.war (substituting the 'x' with the highest number available) from Liferay's download page [<http://www.liferay.com/web/guest/downloads/additional>]. Rename them to "servicemix-web.war" and "jbpm-web.war" respectively.
3. Move the wars from step 2 to the `$TOMCAT_HOME/webapps` directory.
4. Delete the `C:\home\liferay` directory if it exists.
5. Go to the `$TOMCAT_HOME/bin` directory and run `startup.bat`.

5.2. Testing the installation

To test the installation we are going to use the Workflow portlet to create a simple definition and add and instance of it:

1. Login to the portal. The default user is `test@liferay.com` with password `test`.
2. Add "Workflow" portlet to a page.
3. Click on the **Definitions** tab.
4. Click on the **Add** button.
5. Copy and paste the contents of `jbpm-web.war/WEB-INF/definitions/datatypes_definition.xml` into the text area and click the **Save New Version** button.

6. Click on the **Add Instance** icon.
7. From the **Instances** tab, click on the **Manage** icon next to **Enter data**.
8. Fill out the form and click the **Save** button; alternatively, you can test the various error checking capabilities by inputting incorrect values and clicking the **Save** button.
9. Eventually, enter correct values and click the **Save** button.
10. From the **Instances** tab, click on the **Manage** icon next to **View Data**.
11. Confirm that all the data was entered correctly and click the **Finished** button.
12. Confirm that the instance is now in the **End** state.

5.3. Configuration of the jBPM database

The default implementation of jBPM uses an HSQL database found in `jbpm-web.war/WEB-INF/sql/jbpm.*`. To change the location of the HSQL database, change the value of the `hibernate.connection.url` property in `jbpm-web.war/WEB-INF/classes/hibernate.cfg.xml`. The location is addressed to where the start up script for your server is located.

To use a database other than HSQL, first create the database schema using one of the SQL create scripts supplied in the `jbpm-web.war/WEB-INF/sql` directory. Then uncomment the corresponding hibernate connection properties block in `jbpm-web.war/WEB-INF/classes/hibernate.cfg.xml`.

6. Alfresco

Liferay Portal 4.3 provides a portlet that allows the publication of Alfresco contents through the portal. Liferay's download page provides an Alfresco package prepared for deployment.

1. Download the Alfresco war file:
`http://downloads.sourceforge.net/lportal/liferay-portal-alfresco-4.3.x.war?download [??]`.
2. Rename the file to `alfresco.war`.
3. Increase the maximum memory. If you do not do this you will get a PermGen memory error.
4. Go to `tomcat/bin/catalina.bat`. Look for a line that looks similar to the following:

```
SET JAVA_OPTS="-Xms128m -Xmx512m -Dfile.encoding=UTF8  
-Duser.timezone=GMT  
-Djava.security.auth.login.config=$CATALINA_HOME/conf/jaas.config"
```

Note

In Linux/UNIX, edit `catalina.sh` instead. The line should be the same but without the SET command at the beginning.

5. Edit to increase the memory size:

```
SET JAVA_OPTS="-Xms1024m -Xmx1024m -XX:MaxPermSize=128m  
-Dfile.encoding=UTF8 -Duser.timezone=GMT  
-Djava.security.auth.login.config=$CATALINA_HOME/conf/jaas.config"
```

6. Startup Tomcat.
7. Log in as the administrator.
8. In the Plugin Installer Portlet, click on the Upload tab.
9. Click **Browse** and locate the `alfresco.war` file.
10. Click **Deploy**.
11. You can now add the Alfresco Client and Alfresco Content portlet to your page.

Chapter 6. Multiple Portal Instances

Liferay was built from the ground up to be used by application service providers. Because of this it supports having multiple portal instances in a single installation to obtain a complete isolation of the users, organizations, locations, communities and any other data created through Liferay's portlets is achieved.

The following is a sample list of portals running off of one portal instance hitting one database and shows the capabilities of Liferay: <http://demo.liferay.net>, <http://my.ccuc.net>, <http://my.3sixteen.com>, <http://www.gatewayfriends.org>, <http://www.jasonandiris.com>. Users in each of these portals have no information about the other portals. They are separated by domain and each portal exists in its own space based on the company's id.

Starting with Liferay 4.3 it is possible to create new portal instances directly from the web UI with no need to restart the application server. Also this method works with any application server. The creation and administration of portal instances is done through the *admin* portlet. Here are the steps necessary to create a new instance:

1. Preparation: make the necessary DNS configuration to assign a new domain to the server where Liferay is installed. Make sure all the necessary changes to external software (web servers, load balancers, firewalls, ...) have also been done. For these instructions we'll assume that the domain *alpha.com* has been set up.
2. Access the portal as an administrator and go to the *Admin* portlet (for example, add it to the user's personal community if it's not there already)
3. Click the *Instances* tab and then click the *Add* button.
4. Fill the form fields:
 - Web ID: The identifier that will be assigned to this instance. It is recommended that the domain name is used. For example: *alpha.com*
 - Virtual Host: The domain that will be used to access this portal instance. For example: *www.alpha.com*
 - Mail Domain: The domain that will be used to assign email addresses to the users of this instance. For example: *alpha.com*
5. Click save
6. Access the newly created instance in a different browser window. In our example you can do it through the URL <http://www.alpha.com> (or <http://www.alpha.com:8080> if you are accessing tomcat directly and is running in its default port)

Chapter 7. Conclusion

This guide has explained how to install and configure Liferay Portal for the main supported applications servers and databases.

Note that for a production environment you will need to do some extra fine tuning operations. If you need help you can use the community support or the professional support services referred to in the preface of the document.