Liferay Portal 4.0 - Quick Start Guide

Joseph Shum

Alexander Chow

Liferay Portal 4.0 - Quick Start Guide

Joseph Shum
Alexander Chow

Table of Contents

Part I. Users

Table of Contents

Chapter 1. Getting Started

This quck start guide is intended for those who are new to Liferay Portal or are interested in reviewing the latest version. The goal is to get your portal up and running on your machine as quickly as possible. A tour of Liferay Portal and an outline of core functionalities and key features will be given. Installing and setting up the portal on Microsoft Windows XP and Redhat Enterprise Linux 4 using the Liferay Portal and Apache Tomcat will be covered. Each of the topics covered here are also documented in much greater detail in the User Guide and Development Guide. This guide is a great starting point for anyone new to JSR-168 portals and evaluating the technology or reviewing the features of Liferay Portal.

Before we can get started, the following components must be installed on your server.

1. Getting and Installing Java JDK1.5
    - Download and install the **Windows Platform - J2SE(TM) Development Kit 5.0 Update** from http://java.sun.com/j2se/1.5.0/download.jsp.
    - From the **Control Panel**, go to **System** --> **Advanced Tab** --> **Environment Variables**
    - Add a new user variable called **JAVA_HOME** and point this to the directory where you installed the Java JDK.
2. Getting and Installing Liferay Portal and Tomcat Bundle
    - Download and install Liferay Portal Professional 4.0.0 (Bundled with Tomcat and JDK5.0) here [http://www.liferay.com/web/guest/downloads].
    - Unzip the bundle to a directory on your hard drive. Make sure the path of the directory does not have any spaces. Herein, we will call that directory **TOMCAT_HOME.**
3. Starting up Tomcat
    - Go into the bin directory where you installed the bundle **TOMCAT_HOME/bin**
    - Run the file `startup.bat` (if you are looking at it from the Windows Explorer, it may be called `startup`).

Your Apache Tomcat server should now be up with the precompiled Liferay Portal running on it. By default, the portal uses an embedded database called Hypersonic (HSQL). Point your browser to http://localhost:8080 and you should see the portal homepage.

Chapter 2. Using The Portal

From the Liferay Portal home page we can login to the portal by using the account *test@liferay.com* and the password *test*. Once you are logged in, you will see a set of default portlets on your main page. The default account you logged in with has Admin rights to the portal so you are now in complete control to use and customize the portal anyway you like. The rest of this chapter will outline some of the basic features and functionality of the portal.

Adding Portlets

You can add additional portlets to your page by clicking on the **Add Content** link. This will bring up a Content Panel on your screen. Choose a portlet from the menu and add it to your page. If the **Enterprise Admin** portlet is not on your page, add it now. You will see that the portlet has been added to the bottom of your page. To change the portlet placement, click on the title bar of the portlet and drag it to where you like. You can also change the template for your page in the Content Panel. This will allow you to arrange your portlets in one, two or three columns as well as designate the width of the columns. Try to experiment with adding and arranging all the portlets that you would like on this page. There are quite a few portlets that come bundled with Liferay Portal, so be sure to review them.

User Administration

Next, add a user for yourself in the portal. Portal administrative tasks are performed on the **Enterprises Admin** portlet. Only users who have admin rights can access this portlet.

Adding a New User

- Click on the **Users** tab in the Enterprise Admin portlet. (This will list the users in the system)
- Click **Add**.
- Fill in fields and click **Save**.
- Create a password for your new user to log in with. Click **Save**. You now have your own user you can log onto the portal.

Adding Roles to a User

- To give your user Admin rights or other permissions, you need to assign your user specific roles. Click on the **Roles** tab in the **Enterprise Admin** portlet.
- Choose the role that you would like to add your user to. Click on the assign 👥 icon for **Administrator**. You will now see a list of all users who have the adminstrator role.
- Click on the **Available** tab and select your user. Then click on the **Update Associations** button. Your user now has admin rights to the portal.

Now that you've created a new user for yourself, log out of the portal and log in as your new user. You will notice that the portlets you added for the default user are no longer on the screen. You'll need to add the portlets you want back on, and they will be there the next time you log in as the same user. This is because each new user starts off with the same set of portlets. Like just about everything in the Liferay Portal this can be customized to your preference. If you are interested in adding your own organization and location then continue reading. Otherwise feel free to skip ahead to the next section or you can always read the detailed **Security and Permissions** chapter in the Liferay Portal Users Guide.

Adding an Organization

- Select the **Organizatons** tab in the **Enterprise Admin** portlet and then click **Add**.

- Fill in your organization's information and click **Save**. You now have a new organization.

Adding a Location

- Select the **Locations** tab in the **Enterprise Admin** portlet and then click **Add**.

- Fill in your location's information and click **Save**. You now have a new location.

Now that you have Organizations and Locations you can add Roles to these groups just like you would with a user.

Adding a Location to an Organization

- Select the **Organizations** tab in the **Enterprise Admin** portlet and then click the **Add Location** icon.

- Fill in your location's information and click **Save**. You now have a new location under that organization.

Adding Users to an Organization or Location Group

- Click on the **Add Users** icon for the Organization or Group.

- Fill in your User's information and click **Save**. You now have a new User in the organization or location.

- NOTE: You can also move an existing user into a Location or Organization by editing the user and assigning it to an **Organization** or **Location**. This can be done in the **Users** section of the **Enterprise Admin** portlet. Just select the user you are editing and change its details. Don't forget to **Save**!

That's it. You now know enough about user administration to manage an entire company. Experiment with the your own user group hierarchies to obtain the kind of organization and permissioning that your application requires.

Changing the Theme

At this point you have all the portlets you want on your page and you've created and logged in with your own user. For security, delete the test user that you logged in with in the previous section. To delete the test user, click on the **Users** tab in Enterprise Admin portlet and search for the test user. Click on the **Delete** icon.

You can now change the look and feel of your portal. Liferay Portal comes pre-bundled with different themes that you can apply to your page.

Changing the Theme

- Click on the **Page Settings** link at the top of your page.

- Select the page you would like to theme on the left hand side. Note that by default all children pages will inherit the theme from the parent.

- Now select the **Look and Feel** tab for that page.

- You will see a number of bundled themes that are available with Liferay Portal. Choose your theme and color scheme. You can experiment with it as much as you like until you find a theme that pleases you. Don't spend too much time here now... time to move on...

NOTE: Although Liferay comes with many bundled themes, there are also a vast number of themes contributed to Liferay Portal from the community. You can view these on our website or develop your own custom theme for your company or application. Please read the **Building Themes** chapter in the **Liferay Portal Developers Guide.**

Using the Content Management System

One of the core features built into Liferay Portal is the Content Management System (CMS). The CMS provides functionality in two major areas: document management (repositories for documents and images) and publishing. Document management allows you to store, index, and access all electronic documents that you have uploaded to portal. Publising provides a clean separation between your website content and the layout and presentation of that content. You will begin by adding one of your documents to the portal, and then you will learn to design a new page with web content.

Adding a Document to Document Library

Suppose you have a document on your computer right now that you want to share with others in your company or with specific members of your team.

- Clich the **Add Content** link to bring up the **Content Panel** and add the **Document Library** portlet located under the CMS heading.

- Add a new folder by clicking the **Add Folder** button and fill in your folder information. Keep the default permissions and click **Save**.

- Select the document folder you just added and click on **Add Document**.

- Use the **Browse** button to find the document on your local machine and give it a title and description. Click **Save**.

You have just uploaded a document. You can now customize security access for this document folder by clicking on the **Permissons** icon for your folder.

Publishing Content

Creating and publishing content in the CMS is done through the Journal portlet. It is highly recommended that you read over the Content Management System chapter of the Users Guide to understand the various components that make up the CMS and how they work. This section will show you how to create a simple article that you can publish to a page. We will be creating three components to display our content: a structure, a template, and an article.

- **Structures** define the type and number of content pieces (e.g. text, text area, images, selection lists, etc.)
- **Templates** determine how content pieces will be arranged or laid out
- **Articles** are where each content piece is populated with actual text and images

Adding a Structure

- Click the **Add Content**link to bring up the **Content Panel**, and add the **Journal** portlet located under the CMS heading.
- Begin by adding a structure. Click on the **Structures** Tab of the **Journal** Portlet.
- Assign your article a unique **ID**. Fill in a name and description.
- Now you will add the components that you want to use in your content. Under the **XML Schema Definition** section click the **Add Row** button twice to add two compontents. Name the first one *textarea* and the second one *image*. (You can name your components with any name you wish). Next, use the dropdown menu to select **Text Area(HTML)** for the first row and **Image** for the second row. Click **Save**, and you have your first **Structure**.

Now that you have a structure that specifies what type of components you want to work with in your article you will also need a template that specifies how these components will be arranged and styled.

Adding a Component

- Select the **Templates** tab from the **Journal** portlet and click the **Add** button, Alternately you can also click the **Add Template** icon to perform the same task.
- Give your template a unique ID and also add a name and description.
- You will need to select the **Structure** you created if you did not use the **Add Templates** icon for that Structure.

- Now you will need to input the XSL script which formats your components on the page. Click on **Launch Editor** to start formatting your XSL. If you don't know XSL, below are some XSL codes that will layout the two components in your structure for you. It will place your text first and then your image in the next row.

  Replace what is in the editor with this code.

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

  <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  version="1.0">
   <xsl:output method="html" omit-xml-declaration="yes"/>
    <xsl:template match="/">
        <xsl:value-of disable-output-escaping="yes"
  select="root/dynamic-element[@name='textarea-1']/dynamic-content"/>
        <br/>
        <img>
          <xsl:attribute name="src">
            <xsl:value-of disable-output-escaping="yes"
  select="root/dynamic-element[@name='image-1']/dynamic-content"/>
          </xsl:attribute>
        </img>
    </xsl:template>
  </xsl:stylesheet>
  ```

  Click **Update** in the editor, and then click **Save.** A new **Template** has been created.

If you do not have text content and an image you want to add to the site you will need to find some now. We will be putting your content into an article for publishing.

Adding an Article

- Select the **Articles** tab from the **Journal** portlet and click the **Add** button. Alternately, you can click the **Add Article** icon to perform the same task.

- Give your Article a unique ID and add a title. You can also choose a category but we will use General in this case.

- The Journal portlet allows you to specify when you would like your article to be published and when the article expires. Keep the defaults for now.

- You will need to select the **Template** you created if you did not use the **Add Article** icon for that Template.

- The text area is for your text content **which** you should add now. Note that Text Area's allow HTML as input.

- Upload the image you want to display and click **Save**.

You now have an Article created, but it must be approved before you can use the article. Each article created in the

CMS needs to be reviewed and approved by a CMS admin before it can be published. Select the article you've created in the Journal Portlet. Click the **Edit** button and then click **Approve**. Your article is now ready to be published.

Publishing

- Click the **Add Content** link to bring up the **Content Panel** and add the **Journal Content** portlet located under the heading CMS. You will see a message indicating **"Please contact the administrator to setup this portlet. If you are the administrator, go to the configuration screen to setup this portlet."** Disregard this message because you are the administrator. Click the **Configuration** ⚙ icon to view a list of articles you can add to the **Journal Content** portlet.

- Select the article you wrote and click **Choose**. Now go back to the main page by clicking the back ⇦ icon on the portlet tool bar (icon appears when you scroll over the toolbar).

You have just posted your first Journal Content on the Liferay Portal CMS. This was a simple example, but it provided you with a general understanding of how the CMS works. Each article can also have specific permissions set on it or be placed on the main home page or in a secure page. It's all customizable and it's up to you how you use it.

Chapter 3. Configuring The Portal

On start up, the portal loads many default options that configure the portal's behaviour and enables or disables certain features. These properties are loaded from files on the class path called **portal.properties** and **system.properties**. The files can be found in the directory *TOMCAT_HOME\common\classes*. You can overide these files by adding properties to the file **portal-ext.properties**. Some properties that you can add to **portal-ext.properties** are listed below.

Default Location for Document Library

```
dl.root.dir=/home/liferay/documentlibrary/root/
dl.version.root.dir=/home/liferay/documentlibrary/vroot/
```

Default Location for Jackrabbit

```
jcr.jackrabbit.repository.root=/home/liferay/jackrabbit
```

Default Location for Auto Deployment

```
auto.deploy.deploy.dir=/home/liferay/deploy
```

Default Location for Lucene

```
lucene.dir=/home/liferay/lucene/
```

Default Theme and Color Scheme

```
default.theme.id=classic
default.color.scheme.id=01
```

Log all the Paths a User Clicks

```
log.user.paths=true
```

These are just a few properties that can be customized in the portal-ext.properties file. Go to the file portal.properties to see all the properties.

Part II. Developers

Table of Contents

Chapter 4. Tutorials

Setting up the Development Environment

This section is intended for those who have limited to no experience developing portlets in Liferay Portal.

Introduction

This tutorial will cover the setup of a development environment, from using an IDE (such as Eclipse) to setting up your database connections. Although it will be assumed that you are running on a Windows OS, the explanations will be kept general enough so that they can be used for any operating system.

Recommended Requirements

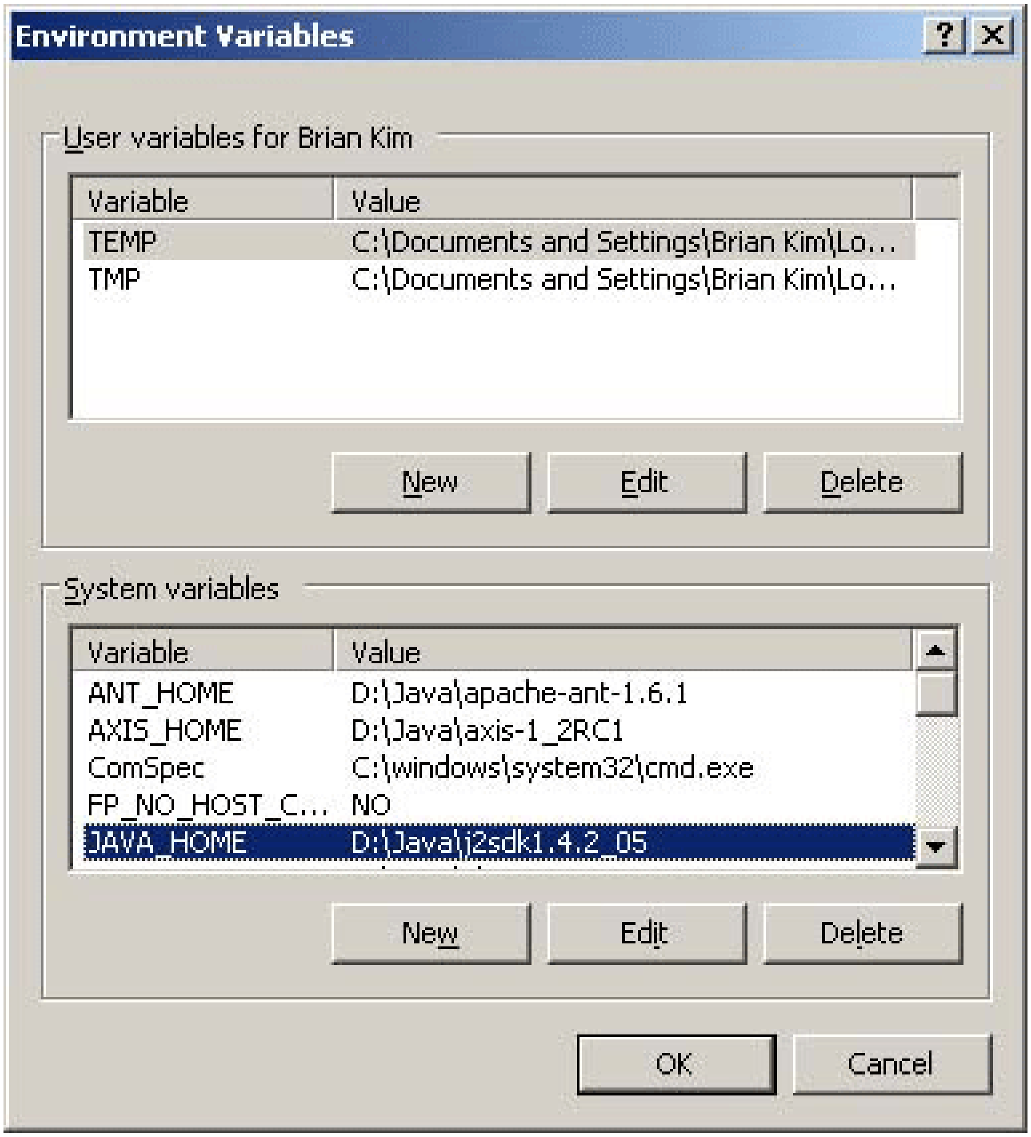
Although lesser hardware requirements may work when developing and customizing Liferay, we recommend the following:

- 1 gig of RAM
- 2.0 ghz Pentium processor

Installation

Setting Up Your Development Environment

As with any enterprise Java application, setting up Liferay can be a daunting task. Liferay leverages several frameworks. Fortunately, most of these frameworks come bundled with Liferay. What you do need to install are some of the basic tools that Liferay uses. First ensure that you are using the right Java JDK. You should be using the latest release of J2SE 1.4.2 or J2SE 5.0, which may already be installed on your machine. If not, download the latest version from http://java.sun.com. Since we will be installing several Java technologies, install/unzip your files under a common folder such as D:\Java. You can choose to install them into any directory of your preference, but from here on out the installation folder will be referred to as {Java}. Although you can use Sun’s default Java compiler, we recommend using IBM’s Jikes compiler – it is faster and generally more descriptive with compile errors. Make sure you download version 1.21 if you are using J2SE 1.42 or version 1.22 if you are using J2SE 1.5. Finally, you will need to download the latest version of Ant, a Java-based build tool, from http://ant.apache.org. Once you have downloaded and installed/unzipped all the files described above, you will need to set your environmental variables to point to them. In Windows, open your **Control Panel**, choose **System** properties, click the **Advanced** tab, and click **Environmental Variables**.



Variable	Value
TEMP	C:\Documents and Settings\Brian Kim\Lo...
TMP	C:\Documents and Settings\Brian Kim\Lo...

Variable	Value
ANT_HOME	D:\Java\apache-ant-1.6.1
AXIS_HOME	D:\Java\axis-1_2RC1
ComSpec	C:\windows\system32\cmd.exe
FP_NO_HOST_C...	NO
JAVA_HOME	D:\Java\j2sdk1.4.2_05

Figure 2.1 Environmental Variables

Choose **New**, and then add the following three variables: **JAVA_HOME, JIKES_HOME**, and **ANT_HOME**. Each respective variable needs to point to the directory in which your tools reside. If you look at Figure 2.1, you can see that the JAVA_HOME variable points to {Java}\j2sdk1.4.2_05. It is recommended that you remove the CLASSPATH variable to prevent conflicts when compiling. It is generally good practice to develop without the use of a classpath variable, since doing so creates a dependency on the developer's machine. Once you have your three System variables setup, you need to edit the **Path** variable by adding the following:

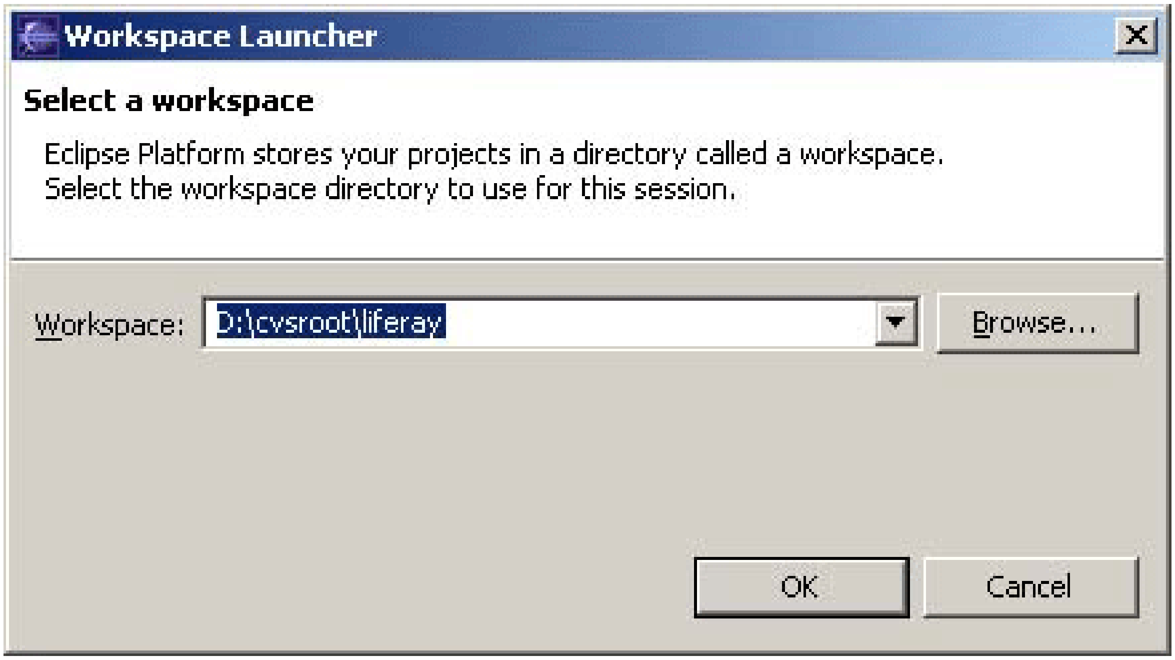
```
%ANT_HOME%\bin;%JAVA_HOME%\bin;%JIKES_HOME%\bin
```

Obtaining Liferay Portal's Source Code

In order to keep current with Liferay's upgrades, you will want to create an extension environment. First, go to http://www.liferay.com/web/guest/downloads and download the latest stable Liferay Portal Enterprise source. Unzip the bundle into a directory like D:\svnroot\liferay (we use SVN as a code repository), which we will refer to as {Liferay}. Then download the latest stable Liferay Portal bundle, in particular Liferay Portal Professional (Bundled with Tomcat) for this tutorial. The Tomcat bundle will be explained later on in this chapter.

Installing Your IDE – Eclipse

Despite the fact that you could modify and deploy Liferay Portal with a simple editor like Textpad, a good IDE can help speed up the process. We will now see how Eclipse plugins can make life easier for editing, deploying, and running Liferay Portal. Download the newest version of the Eclipse from http://www.eclipse.org/downloads/index.php. Install Eclipse into {Java}\eclipse and then run it. Select **File** from the menu bar and then choose **Switch Workspace**. Within the input box, type the path of {Liferay}, as shown in Figure 2.2.


Workspace Launcher

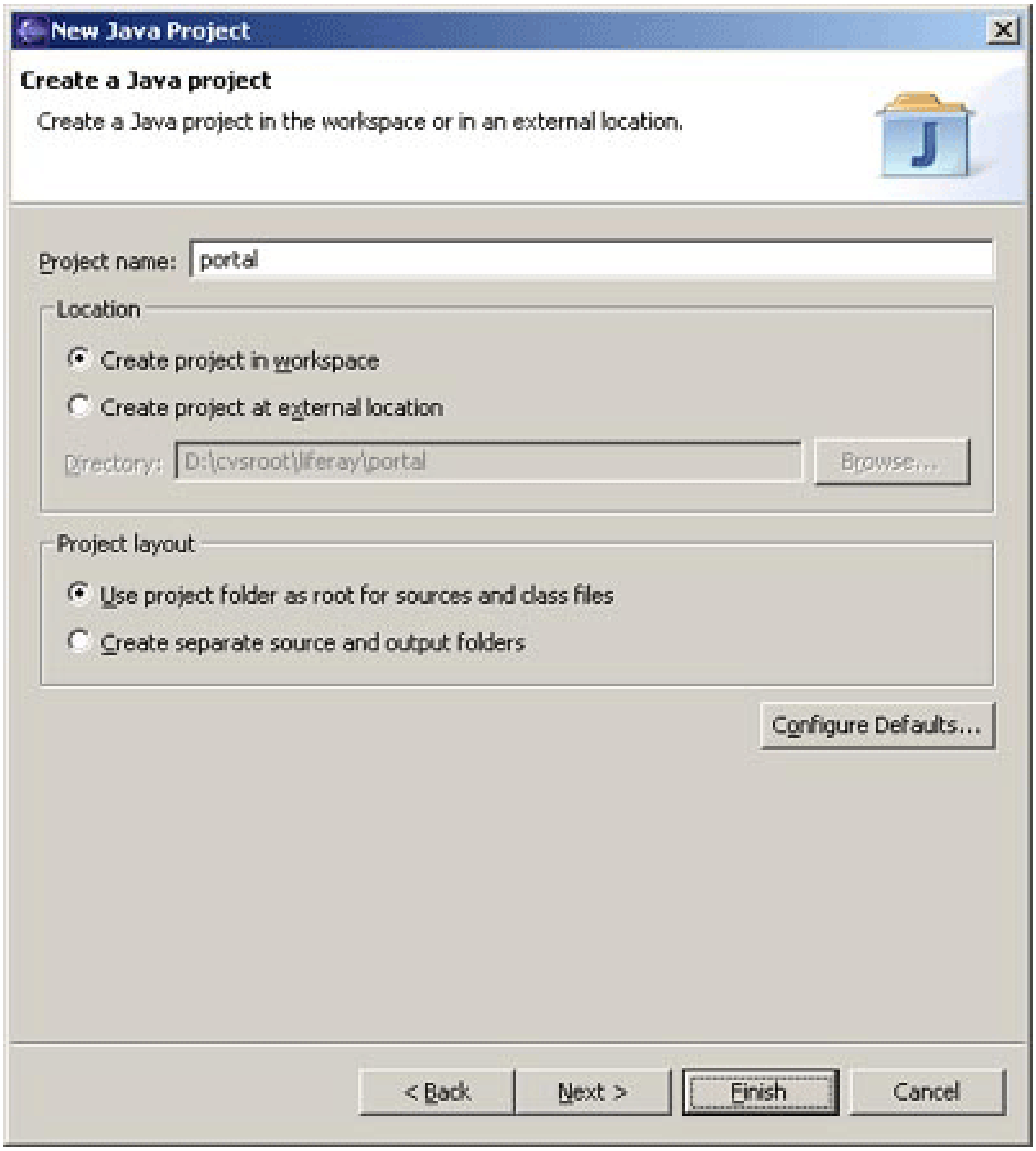
Select a workspace

Eclipse Platform stores your projects in a directory called a workspace.
Select the workspace directory to use for this session.

Workspace: D:\cvsroot\liferay

Figure 2.2 Creating a new Workspace

To setup your project within your workspace, select **File** from the menu bar, **New**, then **Project**. Select **Java Project** from the dialogue box and click **Next**. In the New Java Project window, enter “portal# as the **Project Name**, and then click **Finish**.



New Java Project

Create a Java project

Create a Java project in the workspace or in an external location.

Project name: portal

Location

● Create project in workspace

○ Create project at external location

Directory: D:\cvsroot\liferay\portal

Project layout

● Use project folder as root for sources and class files

○ Create separate source and output folders

Figure 2.3 Adding a new java project

If you have setup the Workspace correctly, you should be able to see the *portal* project within the Navigator pane of the Resource perspective. If you right click on the root directory of the project, select **Properties**, choose **Java Build Path** from the left navigation, and the **Libraries** tab from the right, your portal project should have all the necessary jars loaded already. These jars are loaded via the .classpath file that exists in {Liferay}\portal.

Adding plugin support for Eclipse is even easier. Since Eclipse comes bundled with Ant, setting up Ant is as easy as selecting **Window** from the menu bar, choosing **Show View** and then **Ant**. You should see the Ant tab displayed within your current perspective. To add Liferay's build files, select the Ant tab in your perspective and then click the Ant icon with the + symbol. Note that each directory within Liferay contains its own build.xml file, which Ant reads. When clicking the Ant icon, select the build.xml file within the *portal* directory. Your Ant setup should look like Figure 2.3.

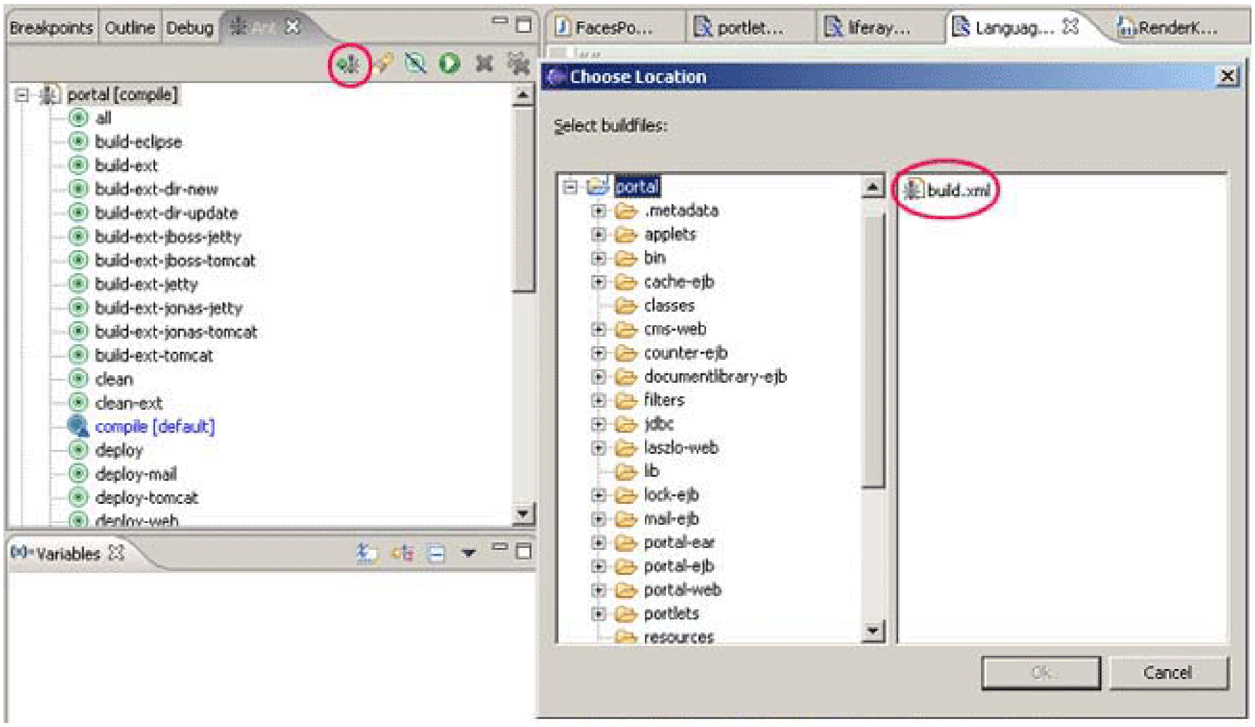


Figure 2.3 Adding the Ant plugin

Creating an Extension Environment

It is recommended that you develop your portlets via an Extension Environment. This will allow you to upgrade your portal easily. Any time a new stable version of Liferay Portal is released, all you would need to do is drop in the new jars and you're set to go. With that said, your development should only be done within the Extension Environment—meaning, all new classes, JSPs, and all changes in general should be within {Liferay}\ext. This will be further explained later.

Setting up your Extension Environment is simple. Before anything, you will need to change a property file (.properties) within {Liferay}\portal. Note that you should never change the values of the default .properties files. Instead, follow the instructions commented at the top of each .properties file, which states that you should instead create an extension file named like *release.${username}.properties*. You can find out what your ${username} value is by opening a command prompt from Start # Run and seeing what name shows up after **Documents and Settings** (i.e. release.Brian Kim.properties). Within your new .properties file, add just one line (with {Liferay} being the actual directory of course):

```
lp.ext.dir={Liferay}/ext
```

Next, in the Ant View described in 2.3, you should be able to expand *portal's* build.xml target list. Among the targets, you should see **clean**, **start**, and **build-ext**. Double click these targets (in that order) while watching the **Console View**. You should see a lot of text scrolling by – indicating that your Extension Environment is being built. To double check, quickly skim the Console output for any red errors. You may find that you might have incorrectly set some of your properties. If all goes well, you should be able to browse your new extension directory by going to {Liferay}\ext!.

Remember the Liferay Portal Professional (Bundled with Tomcat) that you downloaded in section 2.2? Well, here's what you need to do with it.Unzip the file into {Liferay}\ext\servers\tomcat. Like before, you need to create an *app.server.{username}.properties* file within {Liferay}\ext that includes only the two lines:

```
lp.ext.dir={Liferay}/ext app.server.type=tomcat
```

This designates where your Extension Environment is to copy files from and what server directory to deploy to.

In order to develop in Eclipse using the Extension Environment, follow the same instructions for adding a New Java Project (as described earlier for *portal*). However, this time, use the name *ext* as the Project name instead.

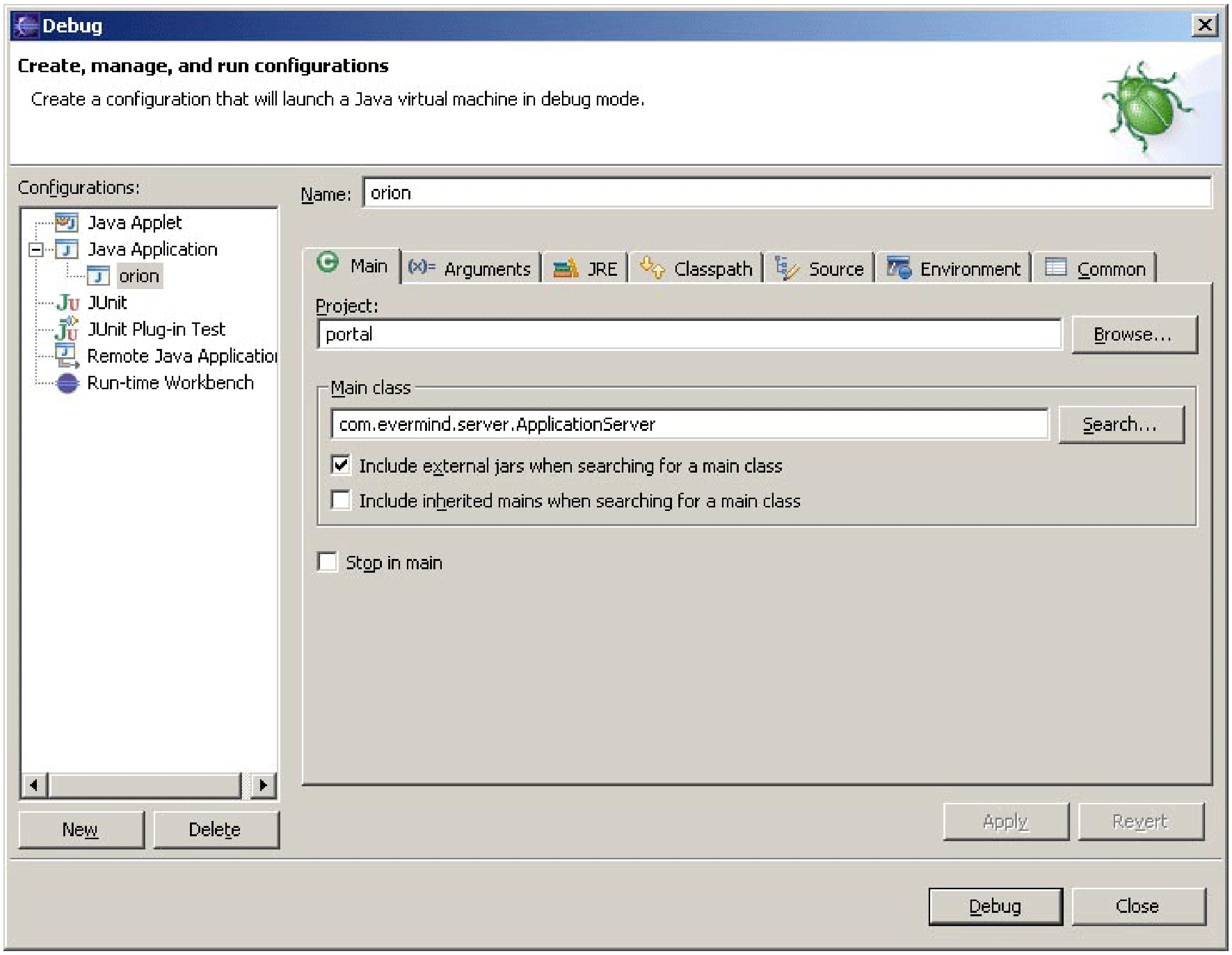
Adding Plugin Support

If you are planning on using Liferay Portal Professional, which implements the Spring framework, it is recommended that you download the Sysdeo Eclipse Tomcat plugin. This plugin will allow you to troubleshoot Liferay Portal using Eclipse's debugger. Download the latest version (I have version 3.0) of the Sysdeo plugin from http://www.sysdeo.com/eclipse/tomcatPlugin.html, and then unzip it into the plugins folder (i.e. {Java}\eclipse\plugins\com.sysdeo.eclipse.tomcat_3.0.0). Close and then restart Eclipse. You should see the Tomcat icons just below the **Search** menu.

Let's configure your Tomcat plugin now. Choose **Windows** from the menu bar, **Preferences**, then select **Tomcat** from the left navigation. You will need to select the appropriate Tomcat version. If you downloaded Liferay Portal Professional (Bundled with Tomcat), then check *Version 5.x*. You will also need to set the following variables:

Tomcat home: {Liferay}\ext\servers\tomcat

Configuration file: {Liferay}\ext\servers\tomcat\conf\server.xml



Debug

Create, manage, and run configurations

Create a configuration that will launch a Java virtual machine in debug mode.

Name: orion

Project: portal

Main class: com.evermind.server.ApplicationServer

☑ Include external jars when searching for a main class

☐ Include inherited mains when searching for a main class

☐ Stop in main

Figure 2.4 Setting up Tomcat in Eclipse

Since Liferay uses JAAS for authentication, you need to add a JVM parameter. Select **JVM Settings** from the left navigation (under the Tomcat node), and click the **Add** button respective to the **Append to JVM Parameters** section. When the small prompt comes up, add this:

```
-
Djava.security.auth.login.config={Liferay}/ext/servers/tomcat/conf/jaas.config
```

To allocate enough memory for Tomcat when Liferay is running, you need to add the following parameter as well (assuming you have at least 512 megs of RAM):

```
-Xmx512m
```

Note: make sure that within your JVM Settings your JRE points to a SDK as opposed to just a JRE (i.e. my JRE has j2sdk1.4.2_05 showing in the dropdown box). This may require you to add another entry within Java (from the left navigation) # Installed JREs. Otherwise, you'll find that your Tomcat runs correctly the first time, but not on successive attempts.

If you are wondering what port your portal will be running on, check out your {Liferay}\ext\servers\tomcat\conf\server.xml and search for your "Connector port# setting. To test to see if you have setup Tomcat correctly, click on the cat-like icon at the upper left of your Eclipse window, and it should start up. Remember that you still have a database to setup.

Hooking Up To a Database

The Liferay bundles come with Hypersonic SQL as the built-in database. But, if you would like to setup Liferay to run on a database, take a look at the **Database** section of the **User Guide**. There it will go in depth into how to setup and connect to your database of choice depending on your server.

Starting Liferay

Now that you have your environment all configured, click the Tomcat icon to start up your web server. As long as you have left the default settings, you should be able to browse to http://localhost and see Liferay Portal load.

Note: because of the nature of Hypersonic, the Sysdeo plugin will not work correctly with our configuration. If you plan on using Hypersonic as your database, you will need to actually browse to {Liferay}\ext\servers\tomcat\bin in your command prompt, and manually type in "startup# to start Tomcat

Developing Portlets

In the next sections, you will learn how to develop a JSR 168 portlet and leverage a lot of the tools that come with Liferay to make deployment easier. You will also learn how to create a simple JSPPortlet before moving on to the more complicated StrutsPortlet.

Writing a Simple JSPPortlet

Although a JSPPortlet does little more than display content, there is still some work that needs to be done. Let's start by creating a new directory called `myjspportlet` within {Liferay}\ext\ext-web\docroot\html\portlet\ext\. Next, open portlet-ext.xml within {Liferay}\ext\ext-web\docroot\WEB-INF\.

Note: You may need to associate .xml files to Eclipse if your .xml files are being opened in a separate editor. You can do this by selecting **Window** from the menu bar and then **Preferences**. Expand the **Workbench** navigation, and click on **File Associations**. From there you can add *.xml as a new File type and associate it to open in Eclipse.

Notice how the portlets are uniquely identified by their portlet-name. As such, you will want to create a new portlet that is an increment of the portlet name, such as EXT_2. Since we are creating a JSPPortlet, you will want the portlet-class to reference the full class name: com.liferay.portlet.JSPPortlet. For this tutorial, add the following to your portlet-ext.xml (you may find it easier to copy and paste EXT_1 and just make the necessary changes):

```
<portlet>
  <portlet-name>EXT_2</portlet-name>
  <display-name>My JSPPortlet</display-name>
  <portlet-class>com.liferay.portlet.JSPPortlet</portlet-class>
  <init-param>
    <name>view-jsp</name>
    <value>/portlet/ext/myjspportlet/view.jsp</value>
  </init-param>
  <expiration-cache>300</expiration-cache>
  <supports>
    <mime-type>text/html</mime-type>
  </supports>
  <portlet-info>
    <title>My JSP Portlet</title>
  </portlet-info>
  <security-role-ref>
    <role-name>Power User</role-name>
  </security-role-ref>
  <security-role-ref>
    <role-name>User</role-name>
  </security-role-ref>
</portlet>
```

Here is a basic summary of what each of the elements represents:

portlet-name	The portlet-name element contains the canonical name of the portlet. Each portlet name is unique within the portlet application.
display-name	The display-name type contains a short name that is intended to be displayed by tools. It is used by display-name elements. The display name need not be unique.
portlet-class	The portlet-class element contains the fully qualified class name of the portlet.
init-param	The init-param element contains a name/value pair as an initialization param of the portlet.
expiration-cache	Expiration-cache defines expiration-based caching for this portlet. The parameter indicates the time in seconds after which the portlet output expires. -1 indicates that the output never expires.
supports	The supports element contains the supported mime-type. Supports also indicates the portlet modes a portlet supports for a specific content type. All portlets must support the view mode.
portlet-info	Portlet-info defines portlet information.
security-role-ref	The security-role-ref element contains the declaration of a security role reference in the code of the web application. Specifically in Liferay, the role-name references which role’s can access the portlet. (A Power User can personalize the portal, whereas a User cannot.)

Now that you have configured your portlet-ext.xml, the next step is to create the jsp pages. Within your `/myjspportlet directory`, add a file called `init.jsp`. Within this file, add the following two lines of code:

```
<%@ include file="/html/common/init.jsp" %>
<portlet:defineObjects />
```

These two lines import all the common class files and also set common variables used by each portlet. If you need to import portlet specific classes or initialize portlet specific variables, be sure to add them to their directory specific init.jsp, as opposed to the common/init.jsp.

These two lines import all the common class files and also set common variables used by each portlet. If you need to import portlet specific classes or initialize portlet specific variables, be sure to add them to their directory specific init.jsp, as opposed to the common/init.jsp.

Now, add a view.jsp. This jsp file will hold the content of your JSPPortlet. Write “Hello [your name here]# within the jsp. So the question is then, how does the portal know how to load these particular files? If you look back at the portlet element that was added within portlet-ext.xml, you will notice that you initialized a `view-jsp` parameter as having the value `/ext/myjspportlet/view.jsp`. By specifying this init-param, you are giving the portlet a default jsp to load.

Finally, in order to be able to add your portlet to the portal, you need to define the name within Language-ext.properties by adding the following line:

```
javax.portlet.title.EXT_2=My JSP Portlet
```

Since you have setup the Extension Environment, you need to deploy the changes you have made to your applicable server by running **deploy** within the build.xml of {Liferay}\ext\. In Eclipse, double click the **deploy [default]** target within your Ant view. Start Tomcat again as soon as the deployment finishes. Browse to the **Home** tab of the portal, and in the **Add Portlet to Wide Column** dropdown add “My JSP Portlet# to your portal.