Liferay Portal 4 - Development in the Extension Environment

Joseph Shum

Alexander Chow

Redmond Mar

Jorge Ferrer

Liferay Portal 4 - Development in the Extension Environment

Joseph Shum
Alexander Chow
Redmond Mar
Jorge Ferrer

1.0

Table of Contents

Preface

Intended audience. This document is intended as a guide for those who want to do extensive customization on top of Liferay Portal. The extension environment provides a set of tools based on ant to develop your own portlets and extension, to modify the configuration and language messages, etc. It also eases the migration to new versions of Liferay Portal. It is meant for those who wish to customize the portal with their own portlets, configuration, etc that will not be given back to the general public Follow the instructions found in Understanding and Improving Liferay [http://wiki.liferay.com/index.php/Category:Understanding_and_Improving_Liferay] in the Liferay wiki if you want to modify the actual source and contribute your changes back to the project.

Liferay version. This guide has been written for Liferay Portal 4. Some details might be different for previous 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 - Installation Guide

- Liferay Portal 4 - Customization Guide

- Liferay Portal 4 - Portal Administration Guide

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. Our only source of revenue is from professional support and consulting.

Chapter 1. Introduction

The extension environment, also known as the ext environment, is a set of tools that allow developers to build portals and portlets on top of Liferay Portal. It can be seen as a Software Development Kit that is independent of any IDE but integrates well with all of them thanks to its usage of *ant*, the most popular and supported build tool. Another way to think about the extension environment is as a wrapper for Liferay's core source because, in most cases, it mirrors Liferay's core source directories (i.e. ext-ejb/ for portal-ejb/, ext-web/ for portal-web/). It allows you to develop on top of Liferay portal, like a platform, providing help when an upgrade to a new version of Liferay Portal is needed.

The following instructions explain how to use the environment and are meant to keep custom code separated from the Liferay Portal code so that upgrades can be made easily. The Ant scripts referred to by these instructions are only available for JBoss+Jetty, JBoss+Tomcat, OracleAS, Orion, RexIP, and Weblogic and have not yet been made available for the other supported application servers.

Chapter 2. Set up

1. Requirements

The following instructions will help you get your development environment ready for working with the source code. These instructions are specific to setting up for deployment to Orion server and Tomcat 5.5 developing with Java JDK 1.5. Liferay Portal is also compatible with Java 1.4 and a wide array of application servers and containers. You will need to adjust your development environment according to your platform.

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

1.1. JDK 1.5.0 or JDK 1.4.2

1. Download and install JDK 1.5.0 [http://java.sun.com/j2se/1.5.0/download.jsp]. JDK 1.4 is also supported.

2. Set an environment variable called JAVA_HOME to point to your JDK directory.

1.2. Jikes 1.22 or Jikes 1.21

Jikes is a Java compiler by IBM that is much faster than the original one provided by the JDK.

1. Download and unzip Jikes 1.22 [http://www-124.ibm.com/developerworks/oss/jikes]. If using JDK 1.4 you have to use Jikes 1.21 instead.

2. Set an environment variable called JIKES_HOME to point to your Jikes directory.

3. Add *JIKES_HOME\bin* to your PATH environment variable.

1.3. Ant 1.6.5

1. Download and unzip the latest version of Ant [http://ant.apache.org/].

2. Set an environment variable called ANT_HOME to point to your Ant directory.

3. Add ANT_HOME\bin to your PATH environment variable.

4. Learn how tasks work in Ant. Tasks can be run at the root of the project directory and inside each subproject directory.

1.4. Subversion (optional)

If you expect to base your extensions on top of the latest sources of Liferay Portal (instead of using an stable release) or you plan to use subversion to maintain your own code you will need a subversion client. Follow these instructions to set it up:

1. Download and install Subversion and/or a subversion client such as SmartSVN [http://www.smartsvn.com/], TortoiseSVN [http://tortoisesvn.net/] or those provided by IDE environments.

2. If you want to browse the source tree, configure the client to use the **https** protocol to connect to **svn.sourceforge.net**. Authenticate as user **anonymous** with a blank password and specify **/svnroot/lportal/portal** as the UNIX path.

   If you are a developer with privileges to commit to the source tree, configure the svn client to use the https protocol to connect to **svn.sourceforge.net.** Authenticate with your private user and password and specify **/svnroot/lportal/portal** as the UNIX path.

3. Check out the portal.

2. Creating the Extension Environment

The extension environment can be built from the sources of Liferay Portal. You can opt to download the sources of the latest version from Sourceforge and unzip them to a clean directory or to obtain the latest version using subversion as explained in the previous section.

Go to the root directory of the Liferay Portal sources and create a file called `release.${user.name}.properties` where `${user.name}` is the name of the user in your computer. Be sure to write it in the proper letter cases (even in Windows). Set the property lp.ext.dir to the directory where you would like to create the directory (make sure that it does not exist yet). If you expect to be using eclipse add also a property called lp.eclipse.project.name with the desired name of the project: For example:

```
lp.ext.dir=D:/Projects/liferay/portal/ext
lp.eclipse.project.name=liferay-ext
```

After creating the file execute the following ant targets to build the sources and create the extension directory structure:

```
$ ant start
$ ant build-ext
```

You can now go to the directory specified and see the directory structure that has been created. If you use some type of version control repository such as CVS or Subversion you can upload the directory to it. In the next steps you will be creating build products that should not be uploaded to the version control repository.

The rest of the document will be referring to the root directory where the extension environment has been download as `/ext`.

3. Configuring the Environment

The extension environment is build around a set of ant build scripts that are highly configurable through properties files and build scripts. You should create customized versions of the properties files by creating a version with the same name than the original one but inserting your user name before the extension. The two most important files that should be customized are:

- `app.server.${user.name}.properties`: set the application server that you are going to

use. For example:

```
app.server.type=jboss-jetty
```

Read `app.server.properties` to find other properties whose values you may want to override.

- `build.${user.name}.properties`: in this file you can set the compiler you want to use (the default is jikes), the amount of memory that it will have, some extra class paths, specific WARs that you may want to deploy, etc. Here is an example that sets these properties:

```
javac.compiler=modern
javac.debug=off
javac.memoryMaximumSize=128m
classpath.ext=/sharedlibs/mycompanylib.jar
deploy.specific.wars=/sharedportlets/mycompanyportlet.war
```

You can also set the property jsp.precompile to on in `build.properties` to precompile JSPs if you are using Jetty, JBoss+Jetty, JBoss+Tomcat, JOnAS+Jetty (other AP. servers do not support this functionality) . This takes a few minutes and should only be used when deploying to a production server.

Read `build.properties` to look for other properties that you might be interested in overriding.

3.1. Advanced customization of the environment

If the customization options provided by the properties files is not enough you can extend the environment by writing your own build scripts. To make this possible all default targets are written in files called `build-parent.xml` that are inherited from the file that ant will actually look for (`build.xml`). You can modify all script files named `build.xml` and add your own targets or override the default ones. When the extension environment is upgraded your modified files will be preserved.

4. Installing the Application Server

The ext environment supports most of the application servers supported by Liferay Portal. The open source application servers are supported by using the bundle that is automatically downloaded from sourceforge. For the proprietary application servers, the basic configurations needed to make them work using a standard distribution are provided.

The default configuration installs all the application servers in the `/ext/servers` subdirectory. This can be changed for any application server by setting a custom location in the file `app.server.${user.name}.properties`. You can also set the path to the Sourceforge mirror that you want to use for downloads or set it to point to a local mirror if you want to avoid external connections for each of the developers. For example, to choose an European mirror and install tomcat outside of the ext environment you can use the following configuration:

```
app.server.type=tomcat
app.server.tomcat.dir=D:/Java/tomcat-5.5.17
sourceforge.mirror=http://belnet.dl.sourceforge.net/sourceforge/lportal
```

Once you are ready follow the instructions specific for each application server:

JBoss+Jetty	Run **ant install-jboss-jetty** under `/ext/servers`
JBoss+Tomcat	Run **ant install-jboss-tomcat** under `/ext/servers`
Jetty	Run **ant install-jetty** under `/ext/servers`
JOnAS+Jetty	Run **ant install-jonas-jetty** under `/ext/servers`
JOnAS+Tomcat	Run **ant install-jonas-tomcat** under `/ext/servers`

OC4J

1. Download `oc4j_extended.zip` [http://www.oracle.com/] from Oracle and copy it to `/ext/downloads`

2. Run **ant install-oc4j** under `/ext/servers`

Orion

1. Download `orion2.0.7 .zip` [http://www.orionserver.com/mirrordownload.jsp?file=orion2.0.6.zip] to `/ext/downloads`.

2. Run **ant install-orion** under `/ext/servers` to install Orion to `/ext/servers/orion`.

Tomcat

1. If you are using Java 5.0 set the following property in `app.server.${user.name}.properties`:

   `app.server.tomcat.zip.name=liferay-portal-tomcat-jdk5-${downloads.`

2. Run **ant install-tomcat** under `/ext/servers`

It is possible install several application servers in the ext environment. Just make sure to set the value of the app.server.type property in `app.server.${user.name}.properties` to the application server that you are using in any specific moment of time.

5. Upgrading the Environment

The process of upgrading to a new version of Liferay Portal is very similar to the one performed for creating the environment the first time:

1. Obtain the sources of the new version of Liferay Portal

2. Create and edit a file named `release.${user.name}.properties` and set the location of the ext environment:

   `lp.ext.dir=D:/Projects/liferay/portal/ext`

3. Build the sources and execute the `build-ext` target to upgrade the existing environment:

   ```
   $ ant start
   $ ant build-ext
   ```

The `build-ext` target detects that the specified location for the ext environment already exists and will only upgrade it, maintaining all changes made in the appropriate locations

Note

Be aware that when updating to certain new Liferay Portal version you may need to also upgrade the version of the application server or do other manual changes. Check the upgrade documentation specific to that version for more information.

Chapter 3. Usage

This chapter explains how to use the extension environment for the most common development activities. After creating the environment you should see the following directories: `/ext/downloads`, `/ext/ext-ear`, `/ext/ext-ejb`, `/ext/ext-lib`, `/ext/ext-web`, `/ext/lib`, `/ext/portlets`, `/ext/servers`, `/ext/sql` and `/ext/web-sites`. The file `/ext/readme.txt` contains basic instructions on how to deploy the extension EAR on the various application servers.

The following sections explain the purpose of each of the subdirectory and the tasks that can be performed from them.

1. /ext

Is the root directory. From this directory you can build and deploy the whole application to the application server by running **ant deploy**. You should not need to place any extra files here.

2. /ext/downloads

Is the directory where you will place downloads that will be used by the ant scripts.

3. /ext/ext-ear

Contains the Liferay Portal build artifacts upon which the environment is based. The `modules` subdirectory contains the file of a Liferay Portal ear unzipped.

Run **ant deploy** from `/ext/ext-ear` to deploy your custom EAR to your specified application server (see `/ext/app.server.properties`). Your custom EAR utilizes the Liferay Portal JARs and WARs contained in `/ext/ext-ear/modules`. This allows you to easily upgrade Liferay Portal because all the functionality is packed into archives.

4. /ext/ext-ejb

This is the folder that will contain all your sources and configuration files (except those related to the web application). Upon creating of the ext environment several important files will be placed in its subdirectories. The most significant are:

- `/ext/ext-ejb/classes/portal-ext.properties`: this file can be used to override the values of the properties in the `portal.properties` configuration file that ships with Liferay Portal.

- `/ext/ext-ejb/classes/system-ext.properties`: this file can be used to override the values of the properties in the `system.properties` configuration file that ships with Liferay Portal.

- `/ext/ext-ejb/classes/content/Language-ext.properties`: this file can be used to add you're own internationalized text messages or to override the messages that ship with Liferay Portal. You can add variations for other languages using the Java convention for message bundles. For example the translation to Spanish should be named `Language-ext_es.properties`.

Run **ant deploy** from /ext/ext-ejb to compile your source and to deploy your classes to the expanded

EAR in your specified application server's deployment directory.

5. /ext/ext-lib

Place any extra dependent libraries in here. They will be copied over to the deployment directory when you run **ant deploy** from `/ext.`

6. /ext/ext-web

This directory will contain your JSPs, HTMLs, images, and all the web application related files inside the `docroot` subdirectory. Here are some of the most common activities you will be performing from this directory:

- Run **ant deploy** to deploy changes in the directory

  If `/ext/ext-web/tmp` does not exist, the script will unjar `/ext/ext-ear/modules/portal-web.war` into `/ext/ext-web/tmp`. The script will then copy everything from `/ext/ext-web/docroot` over to `/ext/ext-web/tmp` and then copy everything from `/ext/ext-web/tmp` to the deployment directory of your specified application server. Do not manipulate the contents of `/ext/ext-web/tmp` manually. This provides an easy way to extend the portal without changing the source and makes upgrading very easy.

  Note that if you have modified the files inside docroot/WEB-INF it is required to run **ant build-website** before **ant deploy**.

- Run **ant fast-deploy** to deploy only changed JSPs.

  Use this version for quick deployment in the process of developing a JSP or set of JSPs

- To add entries to the web application configuration file edit `/ext/ext-web/docroot/WEB-INF/web.xml`. Be aware that this file will be merged with `/ext/ext-web/tmp/WEB-INF/web.xml` and `/ext/web-sites/liferay.com-web/docroot/WEB-INF/web.xml`. The files `web-ejb-ref.xml` and `web-ejb-local-ref.xml` are also merged in. They will only be used if you use Liferay Service Builder to create your portlets.

- To add a portlet, edit `/ext/ext-web/docroot/WEB-INF/portlet-ext.xml`, `/ext/ext-web/docroot/WEB-INF/liferay-portlet-ext.xml`, `/ext/ext-web/docroot/WEB-INF/liferay-display.xml` and `/ext/ext-ejb/classes/content/Language-ext.properties`. These `*-ext.xml` files are read after their parent files are read and override their parent values.

  See the portlet examples for information on general portlet design.

- If you are using StrutsPortlet to develop your portlets the following files will also be of interest to you:

  - `/ext/ext-web/docroot/WEB-INF/struts-config.xml`: add your custom mappings to it. Be aware that when deployed its name changes to `struts-config-ext.xml`.

  - `/ext/ext-web/docroot/WEB-INF/tiles-defs.xml`: add your custom tiles definitions to it. Be aware that when deployed its name changes to `tiles-defs-ext.xml`.

Note

It is strongly suggested that you place all you JSPs and accompanying files inside the subdirectory

`/ext/ext-web/docroot/html`. This directory will be assumed to exist in certain operations, for example when forwarding to a JSP referenced from `struts-config.xml` or `tiles-defs.xml`.

Tip

It is possible to override the JSPs that ship with Liferay Portal by placing them in `/ext/ext-web/docroot/html` and keeping the exact same path than the original version. When running **ant deploy** these files will be copied to `/ext/ext-web/tmp` and will override the original ones. Be aware that you'll have to review any changed JSPs when you upgrade to a new version of Liferay Portal.

7. /ext/lib

This directory contains all of the portal's deployment time dependent libraries and compile time dependent libraries. It contains the following subdirectories:

- development: contains the libraries that will be used during development but will not be deployed. It also contains some JDBC drivers that you can manually deploy to the application server for the specific database in use.

- global: contains the libraries that will be deployed to the global library directory of the application server.

- portal: contains the libraries that will be placed inside the portal EAR.

8. /ext/portlets

Portlets placed in this directory will be automatically deployed to the application server when running the **ant deploy** target from the `/ext` directory.

9. /ext/servers

Contains the targets used to manage the supported application servers and will also contain the installed application servers unless the default paths are changed.

10. /ext/sql

This directory contains all of the portal's database scripts. Read the *Liferay Portal 4 - Installation Guide* for more information about using them to prepare a database for Liferay Portal and to configure the application server being used to run against it.

There are also two files named `test.properties` and `test.script` that are copied over to your specified application server's bin directory when you run **ant deploy** from `/ext`. The two files are used by Hypersonic and contain a snapshot of the portal's default data.

11. /ext/web-sites

All the files in `/ext/web-sites/liferay.com-web/docroot` are copied to the deployment directory when you run **ant deploy** from `/ext` or `/ext/ext-web`.

By default, `/ext/web-sites/liferay.com-web/docroot/index.html` forwards the user to http://localhost/c. You can replace this file to pretty up the default page or change the default behavior.

Chapter 4. Conclusion

The extension environment is a complete development environment that eases customizing Liferay Portal to your own needs. It integrates several Liferay tools that can optionally be used to develop your portlets and portals, such as Liferay StrutsPortlet and Liferay Service Builder.

This guide covers the fundamental information about the environment and the most common tasks that will be performed with it. You might be able to find more specific information in the section about the Extension Environment in the Liferay wiki [http://wiki.liferay.com] where you will also be able to contribute with more information of your own.