

Liferay Portal 4 - Theme Development Guide

Joseph Shum

Alexander Chow

Jorge Ferrer

Liferay Portal 4 - Theme Development Guide

Joseph Shum
Alexander Chow
Jorge Ferrer

Work in progress - Early stages
Copyright © 2000, 2007 Liferay Inc.

Table of Contents

Preface	v
1. Introduction	1
2. Creating themes	2
3. Upgrading Themes	3
1. Upgrading Themes to Liferay Portal 4	3
1.1. Why Hot-Deployable Themes Must Use Velocity Templates:	3
1.2. Should I Use Velocity or JSP?	3
1.3. Upgrading Themes from 3.6.1 to 4 - Velocity	4
1.4. Upgrading Themes from 3.6.1 to 4 - JSP	4

Preface

Intended audience. This document is intended for those who want to create their own themes to change the look and feel provided by Liferay. It is expected that readers have knowledge of Velocity or JSPs and XML. Knowledge of Java packaging is also useful when time comes to deploy the theme. Themes can be developed in the extension environment or as an independent artifact deployed as a WAR file.

Liferay version. This guide has been written for Liferay 4. It includes information for upgrading from 4.1 and some previous versions.

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

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

Note

This guide is a work in progress in its early stages

Chapter 1. Introduction

Chapter 2. Creating themes

Chapter 3. Upgrading Themes

This chapter is targeted to theme developers that want to upgrade themes made for previous versions of Liferay to one of the new versions.

1. Upgrading Themes to Liferay Portal 4

One of the biggest changes for themes in Liferay Portal 4 is that hot-deployable themes must now use Velocity templates. The following outlines why this change was necessary:

1.1. Why Hot-Deployable Themes Must Use Velocity Templates:

Up until Liferay Portal 4.0.0RC2, Liferay used to place its JARs in the shared class loader. The benefit of this was that all web applications could access Liferay's classes since they were in the shared class loader. The drawback was that JARs in the shared class loader would automatically override JARs in other web applications. So if a web application required Hibernate 2.0 and Liferay required Hibernate 3.0, this would cause conflicts because the JARs in the shared class loader would override the JARs in the web application.

Starting from Liferay Portal 4.0.0 final, Liferay no longer places its JARs in the shared class loader. This allows Liferay to work with web applications that require different versions of Struts, Hibernate, etc. without any changes to the code. This also means that hot-deployable themes cannot access Liferay's classes because they are not in the shared class loader.

Here's a more concrete example with Tomcat as the servlet container:

Tomcat's shared class loader is here: `tomcat/common/lib/ext`. Instead of placing its JARs in the shared class loader, Liferay now puts them here: `tomcat/liferay/WEB-INF/lib`. Web applications in `tomcat/webapps` cannot access Liferay's classes because they are no longer in `tomcat/common/lib/ext`.

If your theme is named `*.war`, it is a web application. For example, `ant-themes-4.0.0.war` is a web application. These themes are also called hot-deployable themes since you can deploy them at runtime. Themes that have a `.war` extension must use Velocity templates because they cannot access Liferay's classes.

1.2. Should I Use Velocity or JSP?

Use Velocity for cleaner code and better maintainability. We converted all of our community themes [??] to use Velocity. These themes are a great way to learn how to build your own themes using Velocity. Also see `VelocityTaglib.java` for a list of tags that are available to you.

Use JSP for flexibility or if you do not have time to learn Velocity. If you want to create themes using JSP, you will need to create them in `ext/ext-web/docroot/html/themes`. See `portal/portal-web/docroot/html/themes` for examples on how to create JSP themes.

Note: All themes developed in `ext/themes` or themes that have a `.war` extension cannot access Liferay's classes because of the new class loader. These themes must be developed using Velocity templates. On the other hand, JSP themes must be developed in `ext/ext-web/docroot/html/themes`. These themes must be developed here because they need to access Liferay's classes.

1.3. Upgrading Themes from 3.6.1 to 4 - Velocity

Convert your theme to Velocity. See our community themes [???] for examples on how to create themes using Velocity templates.

1.4. Upgrading Themes from 3.6.1 to 4 - JSP

Port your theme to the ext/ext-web/docroot/html/themes directory. If you do not know how to create an extension environment, read this section first:

<http://content.liferay.com/4.0.0/docs/quickstart/ch04s01.html> [???]

The following guidelines will help you to port your theme to Liferay Portal 4.

1. If you developed your theme as a *.war file or if you developed your theme in ext/themes, deploy your theme to your server. Then copy the resulting theme directory to ext/ext-web/docroot/html/themes. Skip this step if your theme is already located in ext/ext-web/docroot/html/themes.

Example: If your theme is ant-themes-4.0.0.war, deploy this theme to your server. If you are using Tomcat, the theme will be deployed to tomcat/webapps. Copy the resulting "ant-themes-4.0.0" directory and all of its contents to ext/ext-web/docroot/html/themes

When you are finished the directory structure should look like this: ext/ext-web/docroot/html/themes/ant-themes-4.0.0. If your "ant-themes-4.0.0" directory is empty you have not deployed the theme correctly.

2. Convert the JSPs:

Before	After
<liferay:runtime-portlet	<liferay-portlet:runtime
<liferay:include flush="true"	<liferay-util:include Remove "flush" attribute
layoutId	plid
getLayoutId	getPlid
getUserId	getOwnerId
getResTotal	getResolution Only use this method in css.jsp
LayoutServiceUtil.getLayout	LayoutLocalServiceUtil.getLayout
tilesSubNav	This variable was removed

3. Convert icons in portlet_top.jsp to <liferay-portlet> taglib icons:

```
<liferay-portlet:icon-configuration />
<liferay-portlet:icon-edit />
<liferay-portlet:icon-edit-guest />
<liferay-portlet:icon-help />
<liferay-portlet:icon-print />
<liferay-portlet:icon-minimize />
<liferay-portlet:icon-maximize />
<liferay-portlet:icon-close />
```

4. Convert hrefs in top.jsp to use themeDisplay getter methods

```
getURLMyAccount
```

```
getURLSignOut  
getURLAddContent  
getURLPageSettings  
getURLSignIn
```

5. Convert “Communities” to “My Places”

```
<div id="layout-my-places">  
  <liferay-portlet:runtime portletName="<%= PortletKeys.MY_PLACES  
%>" />  
</div>
```

6. Images

- Liferay convention is to put all custom images into images/custom
- Copy default images from Classic theme.

7. CSS Styles

- Remove any styles with “n1” or “w1” in the name
- Add the following styles:

```
portal-add-content layout-column_column-? layout-column-highlight portlet-dragging-placeholder  
layout-my-places
```

- Change the following styles:

Before	After
#gamma-tab	.gamma-tab
#current	.current