

How to

Prepare AltioLive Portlets

Tools: Application Manager,
Designer

Duration: 25 minutes

Skill Level: Medium

Summary: This document covers how to prepare an AltioLive application for deploying on a JSR-168 compliant portal server.

Integra SP

88 Wood Street London
EC2V 7RS
United Kingdom

www.altio.com
tel: +44 (0) 20 8528 1045

Contents

Introduction	2
Configuring AltioLive Portlets	3
Access Control	5
Installation	6
Server Specific Notes	7
WebLogic Portal Server 8.1.....	7
Troubleshooting.....	8

Introduction

This document covers how to prepare an AltioLive application for deploying on a JSR-168 compliant portal server. For help on developing AltioLive Applications, please see the other documents and online help for AltioLive. Please consult your portal server's documentation before deploying AltioLive. Although JSR-168 specifies a standard war file structure for portlet applications, portal servers may require a specific deployment method to be followed, which may differ from the method you use to deploy standard web applications on the same server. Although this document does contain some notes on specific portal servers, they are intended to supplement, not replace the portal server's own documentation.

When developing applications for a portal environment, you should take into consideration the size of the portlet window into which you will be deploying, and the look and feel. Often the portal server will add its own window decoration around your portlet, so you may want to limit your Altio applications to a single window with skins disabled to fit in with the portal. Sample templates are provided in the default application for thick and thin portlet windows (**htmltemplate_thick.txt** and **htmltemplate_thin.xml**). Modify the width and height of the applet space to suit your application and portal.

If you encounter any problems with the steps outlined, please report your experience to **support@altio.com** so that we can update the documentation.

Configuring AltioLive Portlets

Portlets are defined in **portlet.xml** in the **WEB-INF** subdirectory of the Altio web application. A sample **portlet.xml** with some demo portlets is included with your AltioLive installation. Full documentation of the **portlet.xml** file can be found in the JSR-168 spec, at <http://jcp.org/aboutJava/communityprocess/final/jsr168/index.html>

AltioLive Portlets are provided by the class **com.altio.server.portlet.AltioLivePortlet**. This class implements the Portlet API, and obtains the username from the Portlet login, and the application, view and template from the portlet configuration. Different views and templates can optionally be specified for maximized and normal portlets, and different templates can be specified for different device specific content types.

The following configuration parameters are required to be specified as init-param elements of the portlet definition:

- **ApplicationId:** the application to associate with the portlet.
- **Template:** the default HTML template to use for the portlet. This should be an html snippet containing the applet tag, or the javascript required to generate it. The filename is specified relative to the application directory.
- **View:** the default view to use for the application. The view filename is specified relative to the application directory.

The following configuration parameters are optional, and can be specified as init-param elements of the portlet definition to further customize the portlet:

- **MaximizedTemplate:** the HTML template to use for the portlet when it is maximized within the portal. As with Template, this should be a filename specified relative to the application directory.
- **MaximizedView:** the view to use when the portlet is maximized within the portal. The view filename is specified relative to the application directory.
- **AllowAnonymous:** if true, then users do not need to be logged in to view this portlet.
- **XXXTemplate:** the template to use for content-type XXX. XXX is the rightmost part of a MIME type, converted to upper-case. For example, for content-type text/html, XXX would be HTML.
- **XXXMaximizedTemplate:** the template to use for content-type XXX when the portlet is maximized.

The view and template init-param elements can be specified in the portlet-preferences section of the configuration rather than as init-params, in which case it is possible for the user to edit them. This may be useful if your portlet is confined to a small set of trusted users and you have many views per application.

Access Control

Portal Servers contain all the functionality required to implement role based access control to AltioLive applications and views. If access control is required, then all configurations of AltioLive portlets should be done with **init-param** elements, not in the portlet-preferences section. You may also want to configure J2EE security for the Altio servlets to prevent attempts at unauthorized direct connections.

Installation

Once you have defined the portal applications that you want in **portal.xml**, you will need to generate a JSR-168 portlet by zipping the altio web application directory back up into a war file.

In the altio application directory (**tomcat/webapps/altio54**), issue the following command:

```
jar cf ../altio54.war *
```

You may want to clean out your logs directories before you prepare the war file.

Once you have a war file, you should follow your portal server's installation instructions to deploy it.

Due to the fact that Altio needs to be able to write to its configuration and log files, the application must be installed unpacked on servers that give you the option of running directly from war files.

Server Specific Notes

WebLogic Portal Server 8.1

WebLogic Portal Server 8.1 requires JSR-168 portlet war files to be converted before being imported into WebLogic Workshop and deployed. The conversion tool can be downloaded from <http://dev.bea.com/codelibrary/code/portletpreparer.jsp>.

If running in a Unix or Linux environment, you may need to run **dos2unix** on **env.sh** before running it, and you may need to run it in the shell's context with **./env.sh**.

The documentation for deploying on WebLogic Portal 8.1 can be found at <http://dev2dev.bea.com/pub/a/2004/09/deployJava81sp3.html>

Following the instructions for deploying a WSRP portlet allows you to keep Altio in its own context, keeping it separate from the Portal context itself. This may be useful if you want to redeploy Altio at a later date to take advantage of a newer release, or to add new portal applications.

Troubleshooting

- **Error 6: no user id in session messages from AltioLive portlets.**

Check that the hostname and port used to access the portal server are the same as the ones configured under the `http.server.*` properties in `altioserver.xml`. Alternatively, you can modify the html templates of your applications to use javascript to derive the server and port from the browser's address bar instead of using those properties.

- **Could not determine SyncEngine root directory in the Altio server log file.**

Check that the application was unpacked when you deployed it. If not, you need to either redeploy it, or on some servers you can shut down the server, unpack your war file into a subdirectory and move the war file out of the way. The directory into which you unpack the war file will need to be named either exactly the same as the war file, or without the `.war` extension depending on the server.

If you get this message even though the war file is unpacked, try setting the **INITROOT** parameter in `altioserver.xml` to the absolute path on disk to the **WEB-INF/classes** directory of the Altio application. If this does not work after a server restart, then look for other copies of the Altio application in `tmp` or work directories of your app server.

Document Information

Integra SP – Altio

Telephone: +44 (0) 20 8528 1045 Internet: www.altio.com

Copyright © 2009 Integra SP

Copyright in this document is vested in Integra SP. The contents of the document (wholly or in part) must not be reproduced, distributed, used or disclosed without the prior written permission of Integra SP.

Integra recognizes the trademarks or registered trademarks of any third party product or company name referenced in this document at the time of its publication.