

JBoss Portal 2.2

User Guide

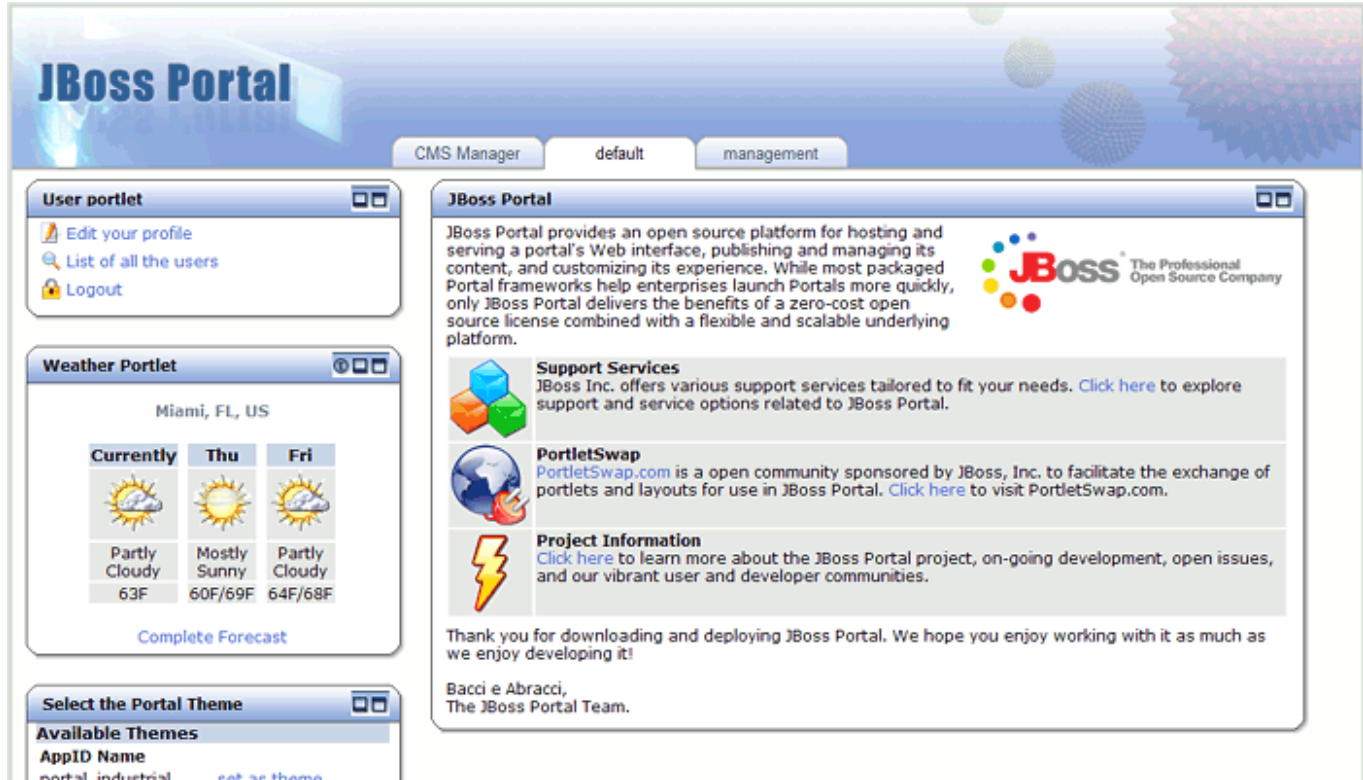
Table of Contents

JBoss Portal - Overview	v
Feature List	vii
Target Audience	x
Acknowledgements	xi
1. System Requirements	1
1.1. Minimum System Requirements	1
1.2. Supported Operating Systems	1
1.3. JBoss Application Server	1
1.4. Database	1
1.5. Source building	2
2. Installation	3
2.1. Install Overview	3
2.1.1. JBoss Application Server	3
2.1.2. Getting the JBoss Portal files	3
2.1.3. Database	3
2.1.4. JDBC connector	4
2.2. Setting up your environment	4
2.2.1. Database	4
2.2.2. JBoss AS Configuration	4
2.3. Deploying JBoss Portal	4
2.3.1. From the binaries	4
2.3.2. From the sources	5
2.4. Running JBoss Portal	5
2.4.1. Logging in	6
3. Customizing your installation	7
3.1. Changing the port	7
3.2. Changing the context path	7
3.3. Forcing the DB dialect	8
3.3.1. DB Dialect settings for the portal core	8
3.3.2. DB Dialect settings for the CMS component	8
3.4. Configuring the Content Store Location	9
3.4.1. Configuring a FileSystem Store	9
3.4.2. Configuring External Blobs with a DB Store	9
4. Upgrading 2.0 - 2.2	10
4.1. Deployment Descriptors	10
4.1.1. Example - Assigning a Portlet on a Page	10
4.2. Content Management System	11
4.2.1. Migrating Content	12
4.3. Forums Migration	13
4.3.1. Forums DB schema issues	13
4.3.2. Portal 2.0.0 to 2.0.1 Forums migration	13
4.3.3. Necessary steps to migrate Forums	14
5. Portal Object Management - Dynamicity	15
5.1. Introduction	15

5.2. Functionalities	16
5.3. Managing portal instances	16
5.3.1. Creating portal pages	17
5.3.2. Securing the portal instance	17
5.3.3. Modifying the portal instance appearance	17
5.4. Managing portlets and portlet instances	17
5.4.1. Creating portlet instances	17
5.4.2. Modifying portlet instance preferences	18
5.4.3. Securing portlets	19
5.4.4. Modifying portlet window appearance	19
5.5. Managing portal pages	20
5.5.1. Creating portlet windows	20
5.5.2. Securing portal pages	21
5.5.3. Modifying the portal page appearance	21
6. CMS Portlet	22
6.1. Introduction	22
6.2. Features	22
6.3. CMS Configuration	23
6.3.1. Portlet Configuration	23
6.3.2. Service Configuration	23
6.4. Localization Support	24
7. CMSAdmin Portlet	25
7.1. Introduction	25
7.2. Features	26
7.2.1. Viewing Directories	26
7.2.2. Viewing File Properties	26
7.2.3. Copying Files/Directories	27
7.2.4. Moving Files/Directories	27
7.2.5. Deleting Files/Directories	28
7.2.6. Creating Directories	29
7.2.7. Creating Text/HTML Files	29
7.2.8. Uploading Files	31
7.2.9. Uploading Archives	31
7.2.10. Editing Text/HTML Files	32
8. User and Role Administration	35
8.1. User Portlet	35
8.2. Role Portlet	37
9. Forums Portlet	39
9.1. Introduction	39
9.2. Functionalities	40
9.3. Installation	41
9.4. Configuration	41
9.5. Security	41
10. Security	43
10.1. Introduction	43
10.2. Securing Portlets	44
10.3. Securing Portlet Instances	44
10.4. Securing Portlet Windows	44
10.5. Securing Pages	44

10.6. Securing Portals	45
11. Clustering Configuration	46
11.1. Introduction	46
11.2. Configuring from Source	46
11.3. Configuring from Binary	47
12. Troubleshooting and FAQ	50
12.1. Troubleshooting and FAQ	50

JBoss Portal - Overview



Many IT organizations look to achieve a competitive advantage for the enterprise by improving business productivity and reducing costs. Today's top enterprises are realizing this goal by deploying enterprise portals within their IT infrastructure. Enterprise portals simplify access to information by providing a single source of interaction with corporate information. Although today's packaged portal frameworks help enterprises launch portals more quickly, only JBoss Portal can deliver the benefits of a zero-cost open source license combined with a flexible and scalable underlying platform.

JBoss Portal provides an open source and standards-based environment for hosting and serving a portal's Web interface, publishing and managing its content, and customizing its experience. It is entirely standards-based and supports the JSR-168 portlet specification, which allows you to easily plug-in standards-compliant portlets to meet your specific portal needs. JBoss Portal is available through the business-friendly LGPL [<http://www.jboss.com/company/aboutopensource>] open source license and is supported by JBoss Inc. Professional Support and Consulting [<http://www.jboss.com/services/index>]. JBoss support services are available to assist you in designing, developing, deploying, and ultimately managing your portal environment. JBoss Portal is currently developed by JBoss, Inc. developers, Novell developers, and community contributors.

The JBoss Portal framework and architecture includes the portal container and supports a wide range of features including standard portlets, single sign-on, clustering and internationalization. Portal themes and layouts are configurable. Fine-grained security administration down to portlet permissions rounds out the security model. JBoss Portal includes a rich content management system and message board support.

JBoss Portal Resources:

1. JBoss Portal Home Page [<http://www.jboss.org/products/jbossportal>]

2. Forums: User [<http://www.jboss.org/index.html?module=bb&op=viewforum&f=215>] | Developer
[<http://www.jboss.org/index.html?module=bb&op=viewforum&f=205>]
3. Wiki [<http://www.jboss.com/wiki/Wiki.jsp?page=JBossPortal>]
4. PortletSwap.com portlet exchange [<http://www.portletswap.com>]
5. Our Roadmap
[<http://jira.jboss.com/jira/browse/JBPORTAL?report=com.atlassian.jira.plugin.system.project:roadmap-panel>]

The JBoss Portal team encourages you to use this guide to install and configure JBoss Portal. If you encounter any configuration issues or simply want to take part in our community, we would love to hear from you in our forums.

Feature List

The following list details features found in this document's related release. For a technical view of our features, view the Project Roadmap and Task List [<http://jira.jboss.com/jira/browse/JBPORTAL>] .

Technology and Architecture

- **JEMS:** Leverages the power of JBoss Enterprise Middleware Services : JBoss Application Server, JBoss Cache, JGroups, and Hibernate.
- **DB Agnostic:** Will work with any RDBMS supported by Hibernate
- **SSO/LDAP:** Leverages Tomcat and JBoss single sign on (SSO) solutions.
- **JAAS Authentication:** Custom authentication via JAAS login modules.
- **Cacheing:** Utilizes render-view caching for improved performance.
- **Clusterable:** Cluster support allows for portal state to be clustered for all portal instances.
- **Hot-Deployment:** Leverages JBoss dynamic auto deployment features.
- **SAR Installer:** Browser-based installer makes installation and initial configuration a breeze.

Supported Standards

- **Portlet Specification and API 1.0 (JSR-168)**
- **Content Repository for Java Technology API (JSR-170)**
- **Java Server Faces 1.2 (JSR-252)**
- **Java Management Extension (JMX) 1.2**
- **Full J2EE 1.4 compliance when used with JBoss AS**

Portal and Portal Container

- **Multiple Portal Instances:** Ability to have multiple Portal instances running inside of one Portal container.
- **IPCTM** Inter-Portlet Communication API enables portlets to create links to other objects such as a page, portal or window .
- **DynamicityTM** The ability for administrators and users to create and destroy objects such as portlets, pages, portals, themes, and layouts at runtime.
- **Internationalization:** Ability to use internationalization resource files for every portlet.
- **Pluggable services:** Authentication performed by the servlet container and JAAS make it possible to swap the authentication scheme.

- **Page-based Architecture:** Allows for the grouping/division of portlets on a per-page basis.
- **Existing Framework support:** Portlets utilizing Struts, Spring MVC, Sun JSF-RI, AJAX, or MyFaces are supported.

Themes and Layouts

- **Easily swappable themes/layouts:** New themes and layouts containing images can be deployed in WAR archives.
- **Flexible API:** Theme and Layout API are designed to separate the business layer from the presentation layer.
- **Per-page layout strategy:** Different layouts can be assigned to different pages.

User and Group Functionality

- **User registration/validation:** Configurable registration parameters allow for user email validation before activation.
- **User login:** Makes use of servlet container authentication.
- **Create/Edit Users:** Ability for administrators to create/edit user profiles.
- **Create/Edit Roles:** Ability for administrators create/edit roles.
- **Role Assignment:** Ability for administrators to assign users to roles.

Permissions Management

- **Extendable permissions API:** Allows custom portlets permissions based on role definition.
- **Administrative interface:** Allows for permissions assignments to roles at any time for any deployed portlet, page, or portal instance.

Content Management System

- **JCR-compliant:** The CMS is powered by Apache Jackrabbit, an open source implementation of the Java Content Repository API.
- **DB or Filesystem store support:** Configurable content store to either a filesystem or RDBMS.
- **External Blob Support:** Configurable content store allowing large blobs to reside on filesystem and content node references/properties to reside in RDBMS.
- **Versioning support:** All content edited/created is autoversioned with a history of edits that can be viewed at any time.
- **Content Serving Search-engine-friendly URLs:** <http://yourdomain/portal/content/index.html> (Does not apply to portlet actions.)

- **No long portal URLs:** Serve binaries with simple urls. (<http://domain/files/products.pdf>)
- **Multiple HTML Portlet instance support:** Allows for extra instances of static content from the CMS to be served under separate windows.
- **Directory Support:** create, move, delete, copy, and upload entire directory trees.
- **File Functions:** create, move, copy, upload, and delete files.
- **Embedded directory-browser:** When copying, moving, deleting, or creating files, administrators can simply navigate the directory tree to find the collection they want to perform the action on.
- **Ease-of-use architecture:** All actions to be performed on files and folder are one mouse-click away.
- **Full-featured HTML editor:** HTML Editor contains WYSIWYG mode, preview functionality, and HTML source editing mode. HTML commands support tables, fonts, zooming, image and url linking, flash movie support, bulleted and numbered list, and dozens more.
- **Editor style-sheet support:** WYSIWYG editor displays current Portal style-sheet, for easy choosing of classes.
- **Internationalization Support:** Content can be attributed to a specific locale and then served to the user based on his/her browser settings.

Message Boards

- **Instant reply:** Instant reply feature, makes for one-click replies to posts.
- **Post quoting:** Quote an existing topic and poster within a reply.
- **Flood control:** Prevents abuse of multiple posts withing a set configurable time-frame.
- **Category creation:** Create a category that contains forums within it.
- **Forum creation:** Create a forum and assign it to a specific category.
- **Forum modification:** Edit, move, delete forums.
- **Forum and category reordering:** Reorder categories and forums in the order you would like them to appear on the page.

Target Audience

This document is intended for those using JBoss Portal. It details the default features found within the standard Portal distribution and addresses configuration issues found in each component. For developers wanting to develop and deploy custom portlets, create/deploy custom themes, or utilize the JBoss Portal API, there is a reference document available.

Acknowledgements

We would like to thank **all** the developers that participate in the JBoss Portal project effort.

Specifically,

1. Thomas Heute, for his help on the first-ever version of JBoss Portal and the corresponding documentation. ;-)
2. Remy for his help with Tomcat configuration.
3. Mark Fernandes and Paul Tamaro from Novell, for their hard work in supplying the portal project with usable and attractive themes and layouts.
4. Kev "kevs3d" Roast for supplying us with two working portlets that integrate existing frameworks in to the portal: Sun JSF-RI and Spring MVC Portlet.
5. Swarn "sdhaliwal" Dhaliwal for supplying us with the Struts-Bridge, that will allow for existing struts applications to work with the Portal.

Contributions of any kind are always welcome, you can contribute by providing ideas, filling bug reports, producing some code, designing a theme, writing some documentation, etc... To report a bug please use our Jira system [<http://jira.jboss.com>] .

System Requirements

Thomas Heute <theute@jboss.org>

Roy Russo <roy@jboss.org>

A list of tested versions or reported as working by users, before reporting a problem please make sure that you are using a compatible version.

If you successfully installed JBoss portal on versions not listed here please let us know so we can add it here.

1.1. Minimum System Requirements

- JDK 1.4 or higher (1.4.2 is recommended)
- 512 MB RAM
- 50 MB hard disk space
- 400 MHz CPU

1.2. Supported Operating Systems

JBoss Portal is 100% pure Java and therefore interoperable with most operating systems capable of running a Java Virtual Machine (JVM); including Windows, UNIX, and Linux.

1.3. JBoss Application Server

JBoss Portal only works with JBoss Application Server.

Note

Currently we recommend using JBoss AS 4.0.3sp1, or greater. Previous versions of JBoss Application Server are not supported.

1.4. Database

JBoss Portal is Database-Agnostic.

Note

JBoss Portal employs Hibernate as an interface to RDBMS. Most RDBMS supported by Hibernate will work with JBoss Portal.

The following list, outlines known-to-be-working database vendor and version combinations:

- MySQL 4.x.x (along with the connector 3.0.16)
- MySQL 5 (known issue [<http://wiki.jboss.org/wiki/Wiki.jsp?page=AvoidMySQL5DataTruncationErrors>])
- PostgreSQL 8.x
- HypersonicSQL
- Derby
- Oracle 9 and 10g
- MSSQL

1.5. Source building

The source building mechanism works on Windows, Linux, MacOS X and any 'Unix like' operating system.

Installation

Thomas Heute <theute@jboss.org>

Roy Russo <roy at jboss dot org>

2.1. Install Overview

There are a couple of archives that you will need to download in order to install JBoss Portal.

2.1.1. JBoss Application Server

Of course you will need to install JBoss Application Server prior to install JBoss portal, if you didn't do so yet, please install JBoss 4.0.3SP1 from Sourceforge [http://sourceforge.net/project/showfiles.php?group_id=22866&package_id=16942&release_id=365509] .

2.1.2. Getting the JBoss Portal files

You can download JBoss portal in different ways, packaged in binaries, sources or from the CVS.

- Packaged: From the JBoss portal project page [<http://www.jboss.com/products/jbossportal/downloads>]
- CVS 2.2 HEAD (The most up to date sources at your own risks):

```
cvs -d :pserver:anonymous@anoncvs.forge.jboss.com:/cvsroot/jboss co jboss-portal-2.2
```

Warning

Do not attempt to get jboss-portal module. The latest stable release *is* jboss-portal-2.2, and the latest in-development release *is* jboss-portal-2.4

2.1.3. Database

You will need a database to store the data of the system, you can use any database supported by Hibernate. We have tested JBoss Portal on the following, but other should work just the same:

- MySQL DB
- Hypersonic DB

- PostgreSQL
- Oracle 10g

2.1.4. JDBC connector

You must make sure that your JDBC connector for your database is under `$JBOSS_HOME/server/default/lib`. The MySQL JDBC connector is available here [<http://dev.mysql.com/downloads/connector/j/3.1.html>] , the postgresSQL JDBC connector is available here [<http://jdbc.postgresql.org/download.html>] .

2.2. Setting up your environment

2.2.1. Database

All databases supported by hibernate are support by JBoss Portal. Below is a generic ordered list of steps that should be followed on any DB:

1. Create a new Database. For MySQL we name it *jbossportal* .
2. Give access rights to whatever user with whatever password to this new database. For MySQL we create a user " *portal* " and give him a password " *portalpassword* ", and grant him rights to the jbossportal DB.

Note

All database tables will be created for you at runtime. The only thing you need to make certain is that there is a database created, a working JDBC connector, and that the user/password combination works.

2.2.2. JBoss AS Configuration

If you need a custom setup of JBoss AS, you should read the documentation about JBoss application server. In our case, we will use the default configuration shipped with JBoss AS 4.0.3SP1.

2.3. Deploying JBoss Portal

2.3.1. From the binaries

The downloaded archive contains the following files:

- jboss-portal.sar
- /setup/portal-*-ds.xml

Note

It is important that you configure the correct datasource file under /setup. There are a few already created for support of popular databases. You can also create your own. Please verify that the username, password, url, and driver-class are correct for your flavor of DB. You can deploy the datasource file by itself to test, in advance.

1. Copy/Move jboss-portal.sar, and your configured portal datasource file to \$JBOSS_HOME/server/default/deploy

2.3.2. From the sources

1. At this stage you should have jboss-4.0.3SP1 installed. First you need to setup JBOSS_HOME environment variable and point it to the root path of the JBoss AS install (ie. C:\jboss-4.0.3SP1) otherwise you won't be able to compile JBoss Portal. To do so go to Start > Settings > Control Panel > System > Advanced > Environment Variables , and add the JBOSS_HOME environment variable. Or do export JBOSS_HOME=/path/to/your/jboss/directory on a Unix-like system.
2. First, build the sources and deploy them, go to jboss-portal-2.0/build and type sh build.sh deploy, you should read BUILD SUCCESSFUL at the end of the operation. This operation should have copied the jboss-portal.sar to your \$JBOSS_HOME/server/default/deploy directory.

Warning

Make sure that JBOSS_HOME is still defined in the environment or it will not work.

3. Now you will need to build the datasource files for DB access. To do so go to jboss-portal-2.0/core and type sh build.sh datasource . It will create all the files under jboss-portal-2.0\core\output\resources\setup.

Note

It is important that you configure the correct datasource file jboss-portal-2.0\core\output\resources\setup. There are a few already created for support of popular databases. You can also create your own. Please verify that the username, password, url, and driver-class are correct for your flavor of DB. You can deploy the datasource file by itself to test, in advance.

4. Before you deploy the application by itself, you will need to have the database deployment descriptor (portal-*-ds.xml) in the \$JBOSS_HOME\server\default\deploy directory. To do so copy the correct portal-*-ds.xml file in to the /deploy directory.
5. You will also need to put the jar file of your database connector in \$JBOSS_HOME\server\default\lib , if you have not already done so.

2.4. Running JBoss Portal

Now you can start JBoss AS by going into \$JBOSS_HOME/bin and typing run . All database tables, cms directories, and initial content for each will be created/inserted during the startup process, if it does not exist.

Using your browser, navigate to <http://localhost:8080/portal> and you should see the portal.

2.4.1. Logging in

You can now use the UserPortlet (top-left) to login. Two default accounts exist:

- user/user
- admin/admin

Customizing your installation

Thomas Heute <theute@jboss.org>

Roy Russo <roy at jboss dot org>

This section is intended to describe some customization features available in JBoss Portal.

3.1. Changing the port

It is common to have a server running on the port 80 instead of the default port 8080.

To change it, you need to edit the file `$JBoss_HOME/server/default/deploy/jbossweb-tomcat50.sar/server.xml` and change the port value of the HTTP Connector. You can also change the value of the SSL port, by default it is set to 8443. Remember to uncomment the following when you have configured it:

```
<!-- SSL/TLS Connector configuration using the admin devl guide keystore
<Connector port="8443" address="{jboss.bind.address}"
    maxThreads="100" strategy="ms" maxHttpHeaderSize="8192"
    emptySessionPath="true"
    scheme="https" secure="true" clientAuth="false"
    keystoreFile="{jboss.server.home.dir}/conf/chap8.keystore"
    keystorePass="rmi+ssl" sslProtocol = "TLS" />
-->
```

Now you can restart JBoss and use the new port that you defined. On systems like Linux, you need privileges to be able to run a server on a port lower than 1000, starting JBoss on the port 80 as a regular user will not work, for testing you can log as root but is not recommended if the server is public as it could be a security breach in your system.

3.2. Changing the context path

By default, the "main" page of JBoss portal will be accessible at `http://localhost:8080/portal/index.html`. You may want to change that either to a different name or to `http://localhost:8080/index.html`.

Note

To do so, edit the file `$PORTAL_HOME/build/local.properties` and change `portal.context-root` to any-

thing you want.

Now you can rebuild JBoss portal and redeploy it for the context path changes to take effect.

3.3. Forcing the DB dialect

If you encounter that the Hibernate dialect is not working properly and would like to override the default behaviour, follow the instructions contained in this section:

Note

Under most common circumstances, the auto-detect feature should work fine.

3.3.1. DB Dialect settings for the portal core

Modify `jboss-portal.sar/conf/hibernate/[module]/hibernate.cfg.xml`. A list of supported dialects for Hibernate3, can be found here [http://www.hibernate.org/hib_docs/v3/reference/en/html/session-configuration.html#configuration-optional-dialects].

```
<?xml version='1.0' encoding='utf-8'?>
<!DOCTYPE hibernate-configuration PUBLIC
"-//Hibernate/Hibernate Configuration DTD//EN"
"http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
<session-factory>
<property name="connection.datasource">java:PortalDS</property>
<property name="show_sql">false</property>
<property name="cache.provider_class">org.hibernate.cache.EhCacheProvider</property>
<property name="cache.use_query_cache">true</property>

<!-- Force the dialect instead of using autodetection -->
<!--
<property name="dialect">org.hibernate.dialect.PostgreSQLDialect</property>
-->

<!-- Mapping files -->
<mapping resource="conf/hibernate/user/domain.hbm.xml"/>
</session-factory>
</hibernate-configuration>
```

3.3.2. DB Dialect settings for the CMS component

Modify `jboss-portal.sar/portal-cms.sar/META-INF/jboss-service.xml`. A list of supported dialects for Hibernate3, can be found here [http://www.hibernate.org/hib_docs/v3/reference/en/html/session-configuration.html#configuration-optional-dialects].

You will be searching this file for the commented "`org.hibernate.dialect.`" blocks. Once found, replace with whichever dialect you are trying to force from the Hibernate list. There are **5** such commented blocks in this file. Make sure you uncomment them all!

3.4. Configuring the Content Store Location

By default, the JBoss Portal CMS stores all node properties, references, and binary content in the database, using the portal datasource. The location of some of these items is configurable.

3.4.1. Configuring a FileSystem Store

You will note that the jackrabbit configuration file is set to use the `HibernateStore` and `HibernatePersistenceManager` classes, by default. To have the CMS use 100% file system storage, simply comment these blocks. Then, you should uncomment to use the `LocalFileSystem` and `XMLPersistenceManager` classes. The repository configuration file is located under: *jboss-portal.sar/portal-cms.sar/META-INF/jboss-service.xml*. Remember to perform this switch for the Repository, Workspace, and Versioning attributes.

3.4.2. Configuring External Blobs with a DB Store

Some enterprise deployments that serve large multimedia content prefer to not have the large files stored in the DB, along with all the property and node information. When using the `HibernateStore` and `HibernatePersistenceManager`, you can elect to have binary content stored on the local filesystem. Under the `WorkSpace` and `Versioning` nodes, set *externalBlobs* to *true* to achieve this.

4

Upgrading 2.0 - 2.2

Roy Russo <roy at jboss dot org>

Boleslaw Dawidowicz <boleslaw.dawidowicz@jboss.com>

This chapter addresses migration issues from version 2.0 to 2.2 of JBoss Portal.

4.1. Deployment Descriptors

From version 2.0 to 2.2, the JBoss Portal deployment descriptors have changed when defining pages, portlets, and portal instances.

4.1.1. Example - Assigning a Portlet on a Page

To describe the changes made to the deployment descriptors, we have made available an example that you can download [here](http://labs.jboss.com/file-access/default/members/portletswap/downloads/portlets/samples/HelloWorldPortlet.zip): HelloWorld Portlet [http://labs.jboss.com/file-access/default/members/portletswap/downloads/portlets/samples/HelloWorldPortlet.zip]. After this helloworldportlet.ear is deployed, you should be able to access the new portal page by pointing your browser to <http://localhost:8080/portal/portal/default/HelloWorld>.

All portal, page, and portlet instance deployment is now handled by one file: *-object.xml. You no longer need the *-portal.xml, *-pages.xml, and *-instances.xml found in JBoss Portal 2.0. For our example we make available *helloworld-object.xml* located under *helloworldportlet.war/WEB-INF/*, and it looks like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<deployments>
<deployment>
<if-exists>overwrite</if-exists>
<parent-ref>default</parent-ref>
<properties/>
<page>
<page-name>Hello World</page-name>
<properties/>
<window>
<window-name>HelloWorldPortletWindow</window-name>
<instance-ref>HelloWorldPortletInstance</instance-ref>
<region>center</region>
<height>0</height>
</window>
</page>
</deployment>
<deployment>
<if-exists>overwrite</if-exists>
```

```
<instance>
  <instance-name>HelloWorldPortletInstance</instance-name>
  <component-ref>helloworld.HelloWorldPortlet</component-ref>
</instance>
</deployment>
</deployments>
```

A deployment file can be composed of a set of `<deployments>`. In our example file, above, we are defining a page, placing the `HelloWorldPortlet` as a window on that page, and creating an instance of that portlet. You can then use the Management Portlet (bundled with JBoss Portal) to modify the instances of this portlet, reposition it, and so on...

- **<if-exists>** Possible values are *overwrite* or *keep* . *Overwrite* will destroy the existing object and create a new one based on the content of the deployment. *Keep* will maintain the existing object deployment or create a new one if it does not yet exist.
- **<parent-ref>** Indicates where the object should be hooked in to the portal tree. See the chapter on Dynamicity for the tree-concept within JBoss Portal.
- **<properties>** Properties definition specific to this page, commonly used to define the specific theme and layout to use. If not defined, the default portal layouts/theme combination will be used.
- **<page>** The start of a page definition.
- **<page-name>** The name of the page.
- **<window>** The start of a window definition.
- **<window-name>** The name of the window.
- **<instance-ref>** The instance reference used by this window. Should correspond with the `<instance-name>` variable.
- **<height>** The vertical position of this window within the region defined in the layout.
- **<instance>** The start of an instance definition. page.
- **<instance-name>** Maps to the above `<instance-ref>` variable.
- **<component-ref>** Takes the name of the application followed by the name of the portlet, as defined in the *portlet.xml*

Note

For further explanation of the deployment descriptor, please view the XMLDescriptor section in the Reference Guide

4.2. Content Management System

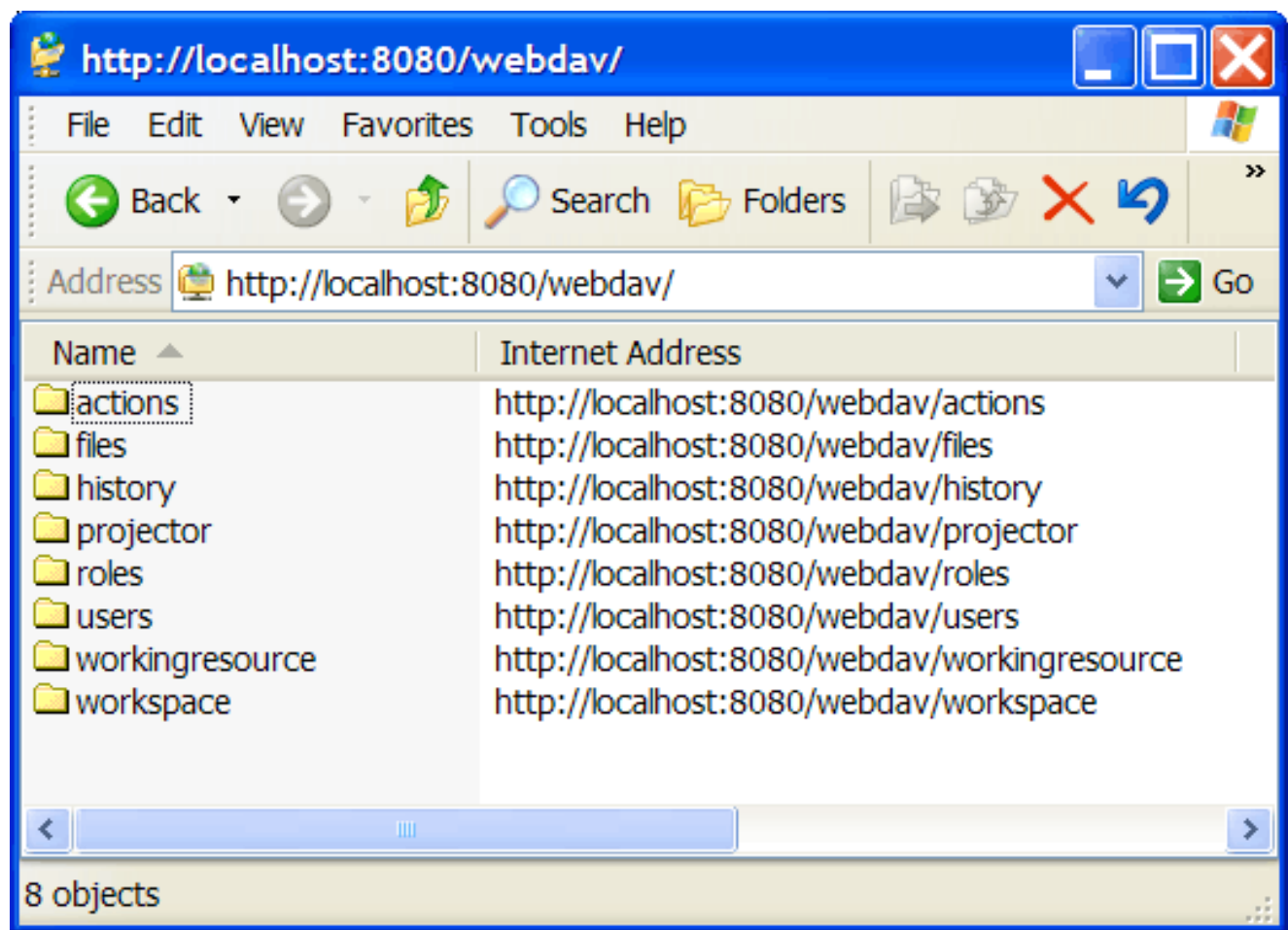
From version 2.0 to 2.2, the JBoss Portal Content Management System changed from using Apache Slide API to the Java Content Repository (JCR), JSR-170 [<http://www.jcp.org/en/jsr/detail?id=170>] using the Apache Jackrabbit [<http://incubator.apache.org/jackrabbit/>] implementation.

4.2.1. Migrating Content

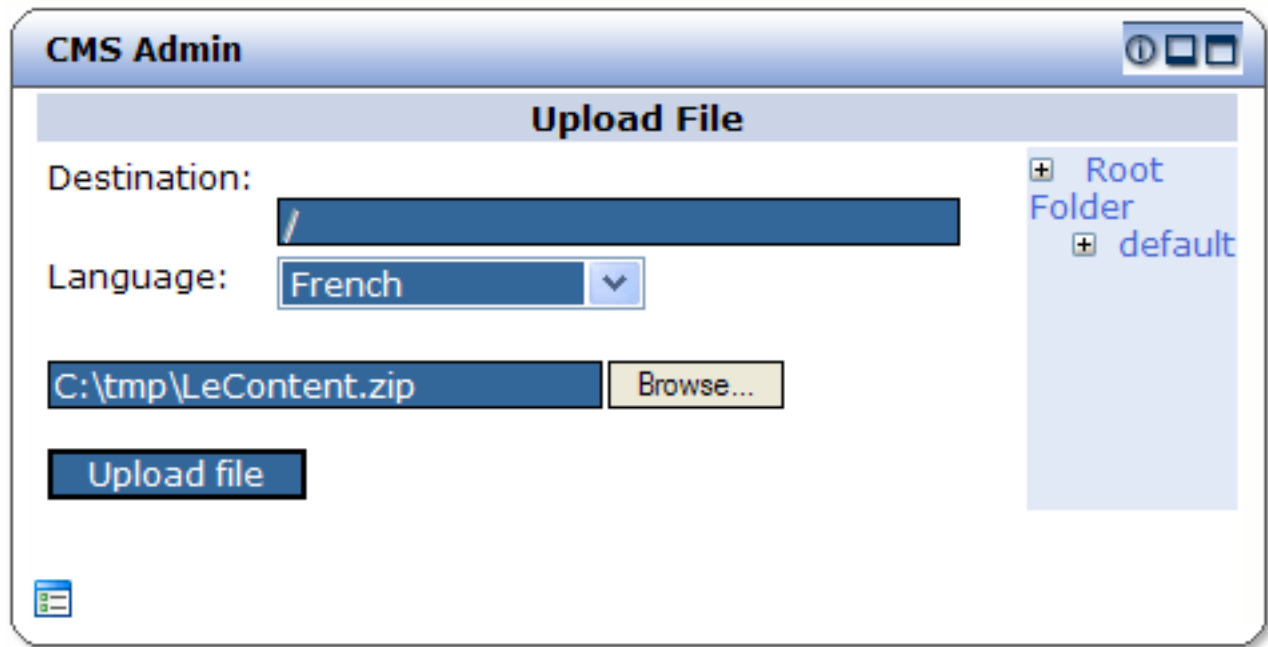
Since the underlying layer of the CMS has changed, it will be necessary for users migrating from 2.0 to move their content, so the following steps describe how to perform this operation.

JBoss Portal v2.0 had native WebDAV support, allowing a user to connect to the content repository via the Operating System, given the proper credentials. You will use this method to extract the content, zip it in an archive, and upload it to the new CMS.

- First, start up your previous installation of JBoss Portal 2.0, and connect to it using MS WebFolders. Using the *Add Network Place* option under *My Network Places*, add a new network place, giving it the path to your webdav repository. By default it is `http://localhost:8080/webdav`. Upon providing the proper credentials, you should see your repository structure.



- Navigate to `http://localhost:8080/webdav/files` and your entire content directory structure with files should be available here. You should be able now to zip these directories and upload them as an archive to the JBoss Portal 2.2 CMS via the CMSAdminPortlet.



Warning

There are two known issues you need to know about when importing content from the old repository using this method:

- Version information will be lost.
- You must verify that pre-existing links to local resources are correct.

4.3. Forums Migration

4.3.1. Forums DB schema issues

Database schema differs slightly between portal 2.0.0 and 2.0.1 versions. Some new tables were added for new functionality. There were few columns removed or type changed also.

From 2.0.1 RC2 version portal performs schema update try during startup/deployment. Hibernate SchemaUpdate hbm2ddl tool is able to add new tables or new columns. What it doesn't do is removing unnecessary columns or column sql-type changes.

Besides of that, it is always good to back up your data as this behaviour might depends on different RDBMS versions.

4.3.2. Portal 2.0.0 to 2.0.1 Forums migration

In portal 2.0.1 there are some changes in db schema related to Forums Portlet

For eg. columns such as:

- jbp_forums_forums --> jbp_last_post_id
- jbp_forums_topics --> jbp_first_post_id
- jbp_forums_topics --> jbp_last_post_id

are now not used. These are retrieved using Hibernate collections storing capabilities.

Column:

- jbp_forums_posts --> jbp_text

had wrong SQL type. It was 'varchar(255)' in 2.0.0 and it is 'text' in 2.0.1.

4.3.3. Nessesary steps to migrate Forums

After upgrading portal to 2.0.1, schema should be updated automaticly and all new nessesary tables/columns created. If this process fail the schema will be dropped/created. Remember to backup your data before doing migration!

After successfull update beware of the fact that you will have:

- a number of unused columns in schema
- texts of messages stored in varchar(255) column - Posts in forums couldn't be longer than 255 chars. In fact longer messages will cause portlet exception...

To deal with second issue we must change jbp_forums_posts-->jbp_text column type. It's very simple to do in MySQL RDBMS:

```
ALTER TABLE jbp_forums_posts CHANGE jbp_text jbp_text text
```

In Postgres it will be:

```
ALTER TABLE portal.jbp_forums_posts ALTER jbp_text TYPE text;
```

This will change column type.

Check in your RDBMS docs if such ALTER TABLE SQL statement works. If not you should probably recreate jbp_forums_posts table with proper SELECT/INSERT statement.

Portal Object Management - Dynamicity

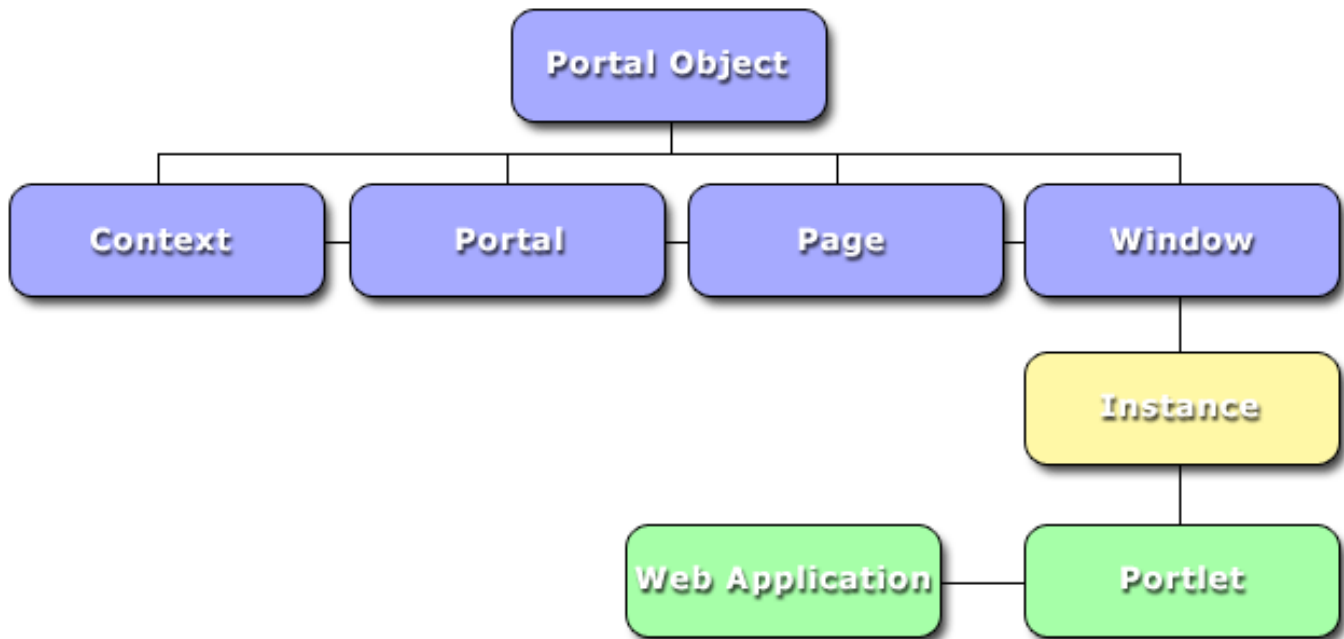
Roy Russo <roy @ jboss dot org>

5.1. Introduction

The concept of dynamicity is that all portal objects are able to dynamically be modified at runtime, eliminating the need to struggle with large xml files, or restarting the application server for changes to take effect. In the scope of dynamicity, Portal objects are defined and can be altered as follows:

- Portal Instances: Multiple Portal instances can be launched at any time, secured, and skinned.
- Pages: Multiple pages, composed of windows, can coexist inside a portal instance, secured and each could have its own theme applied, if desired.
- Portlets: Can be hot-deployed and will instantly register with the portal, appearing in the management portlet.
- Portlet Instances: Instances can be created/destroyed. Their preference variables can be modified at runtime, instances assigned to windows and secured.
- Windows: Windows can be secured, created/destroyed, and assigned to specific pages.
- Themes/Layouts: Can be hot-deployed and will register with the portal, appearing the management portlet, allowing for dynamic customization of any of the above components.

Dynamicity and our new consolidated deployment descriptor work hand-in-hand. Portal objects are all considered branches to the root portal context, and within our new deployment descriptor all branches (and leaves) can be specified.



5.2. Functionalities

Dynamicity allows an administrator to manage the entire portal deployment from within a portlet. Some of the many tasks available to administrators are:

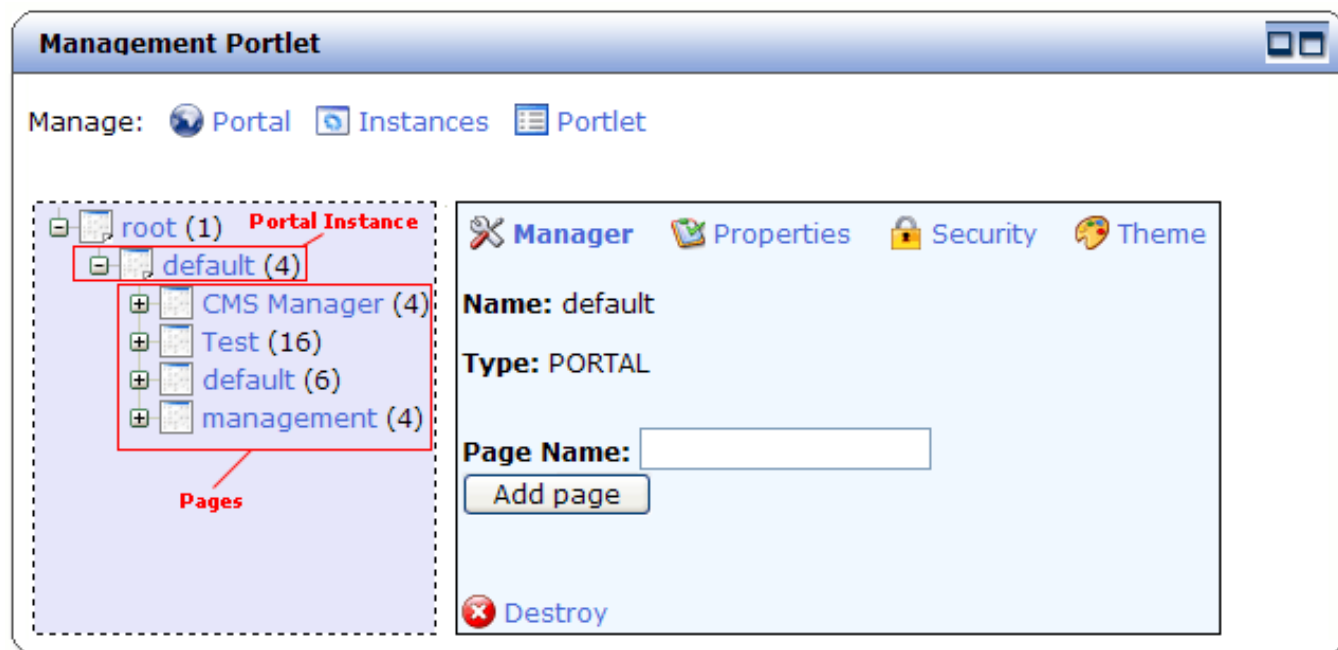
- Create/Destroy portal instances.
- Create/Destroy portlet instances.
- Create/Destroy portal pages and subpages.
- Assign pages to portal instances.
- Assign portlet instances to pages.
- Modify instance variables for portlets.
- Control the order and region in which portlets appear on a page.
- Modify the theme and/or layout for a portal instance or a specific page.
- Modify the security constraints for a portal instance or a specific page.

Note

For the remainder of this chapter, the term **object** will be used to mean any Portal, Page, Portlet, or Window.

5.3. Managing portal instances

Administrators may manage the portal, pages, subpages, and windows at any time, by clicking on the "Portal" tab at the top of the Management Portlet. The components currently deployed in the portal container are displayed in a tree-structure for ease-of-navigation and modification.



5.3.1. Creating portal pages

By selecting a portal instance, and then selecting the *Manage* option, you can create a new page. Simply assign a name to a page and submit the form.

5.3.2. Securing the portal instance

Selecting the *Security* option, allows an administrator to secure the portal instance. On this screen, you can select and unselect portal-wide security settings for a given role.

5.3.3. Modifying the portal instance appearance

Selecting the *Theme* option, allows an administrator to modify the look-and-feel for the chosen portal instance.

5.4. Managing portlets and portlet instances

5.4.1. Creating portlet instances

Management Portlet

Manage: Portal Instances Portlet

Web application id : samples/portal-samples

Name	State	Description
Portlet Test	Started	C'est une portlet de test
HeaderContentPortlet	Started	-
CachedCounterPortlet	Started	-
ThemeSelectorPortlet	Started	-
WsrpSelectorTestPortlet	Started	-
ExceptionPortlet	Started	-
ContentRewritePortlet	Started	-
PortletA	Started	-
SecuredTestPortlet	Started	-
HintPortlet	Started	-
CharsetPortlet	Started	-
PortletB	Started	-
PortletSessionPortlet	Started	-
PreferencesPortlet	Started	-
CounterPortlet	Started	-

Web application id : portal/portal-core

Name	State	Description
ThemeManager	Started	-
ManagementPortlet	Started	-
UserPortlet	Started	-
CatalogPortlet	Started	-
CMSPortlet	Started	-
ThemeSelectorPortlet	Started	-
PolicyConfiguratorPortlet	Started	-
NavigationPortlet	Started	-
CMSAdminPortlet	Started	-

Manager
Security

Name: CMSPortlet
 State: Started
 Description: -

Clicking on the *Portlet* tab and then clicking on a specific portlet allows you to create a new instance of this portlet.

5.4.2. Modifying portlet instance preferences

Management Portlet

Manage:
Portal
Instances
Portlet

Id	Portlet	Action
PolicyConfiguratorPortletInstance	PolicyConfiguratorPortlet	
CMSPortletInstance	CMSPortlet	
CMSAdminPortletInstance	CMSAdminPortlet	
ManagementPortletInstance	ManagementPortlet	
NavigationPortletInstance	NavigationPortlet	
UserPortletInstance	UserPortlet	
CatalogPortletInstance	CatalogPortlet	
ThemePortletInstance	ThemeManager	
ThemeSelectorInstance	ThemeSelectorPortlet	
TestPortletInstance	Portlet Test	
CounterPortletInstance	CounterPortlet	
CachedCounterPortletInstance	CachedCounterPortlet	
PortletSessionPortletInstance	PortletSessionPortlet	
CharsetPortletInstance	CharsetPortlet	
ExceptionPortletInstance	ExceptionPortlet	
MissingPortletInstance	Portlet not deployed	
PreferencesPortletInstance	PreferencesPortlet	
PortletAInstance	PortletA	
PortletBInstance	PortletB	
HeaderContentInstance	HeaderContentPortlet	
ContentRewriteInstance	ContentRewritePortlet	
SecuredTestPortletInstance	SecuredTestPortlet	
HintPortletInstance	HintPortlet	
WsrpSelectorTestPortletInstance	WsrpSelectorTestPortlet	
CMSInstance2	CMSPortlet	

Preferences
Security

Key	Description	ReadOnly	Value	Instance	Portlet
html chars	Description	false	changeme		%3C%26%3E%22
read-write foo	Description	false	bar		bar
read-write foo	Description	false	bar0,bar1		bar0,bar1
read-only foo	Description	true	bar		bar

Clicking on the *Instance* tab and selecting a portlet instance, displays a screen which allows an administrator to edit the portlet instance preferences.

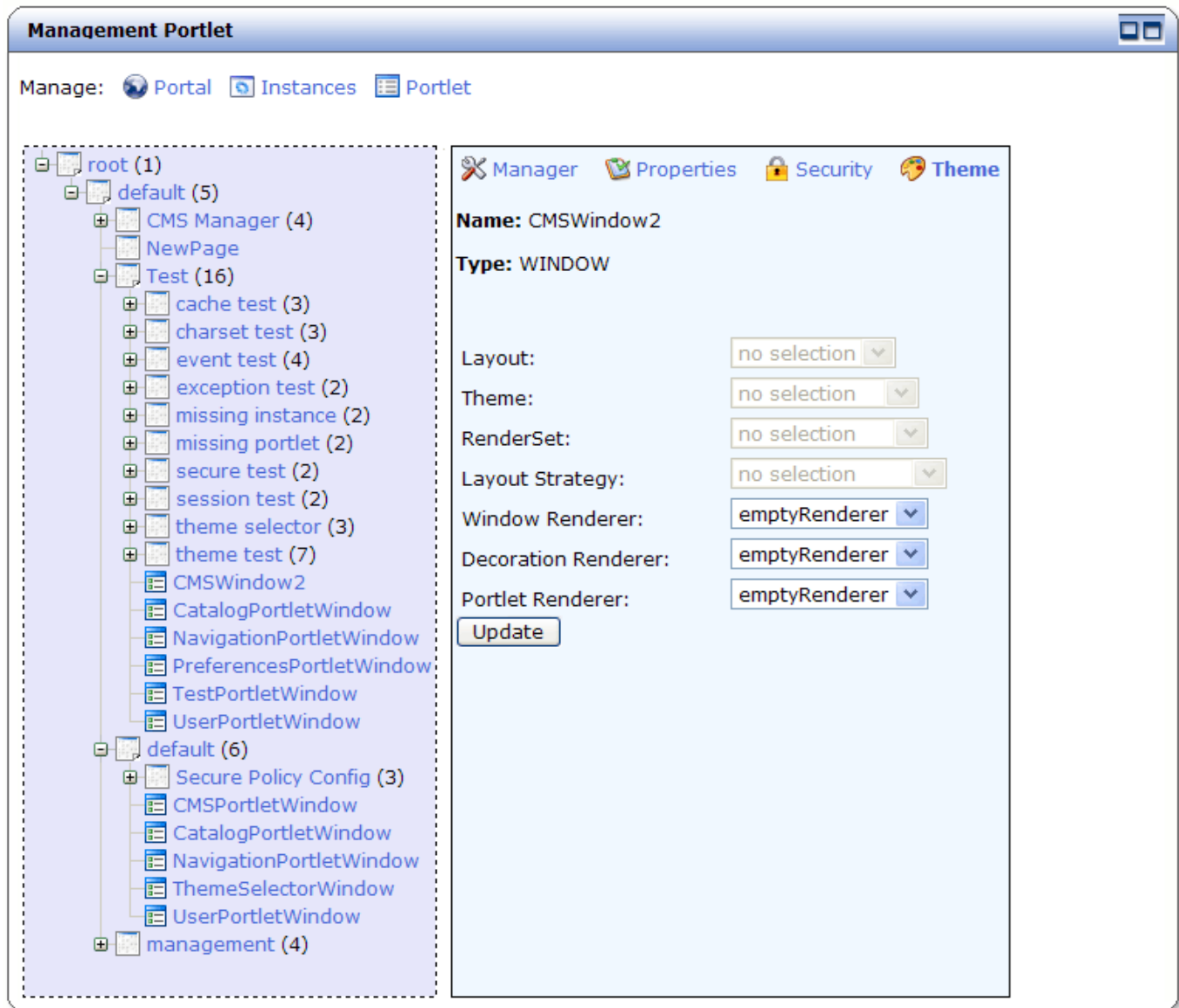
5.4.3. Securing portlets

Selecting the *Security* option, allows an administrator to secure the portal page.

Note

Depending on whether you are securing a portlet or an instance, keep in mind that instance security constraints take precedence over portlet security settings.

5.4.4. Modifying portlet window appearance



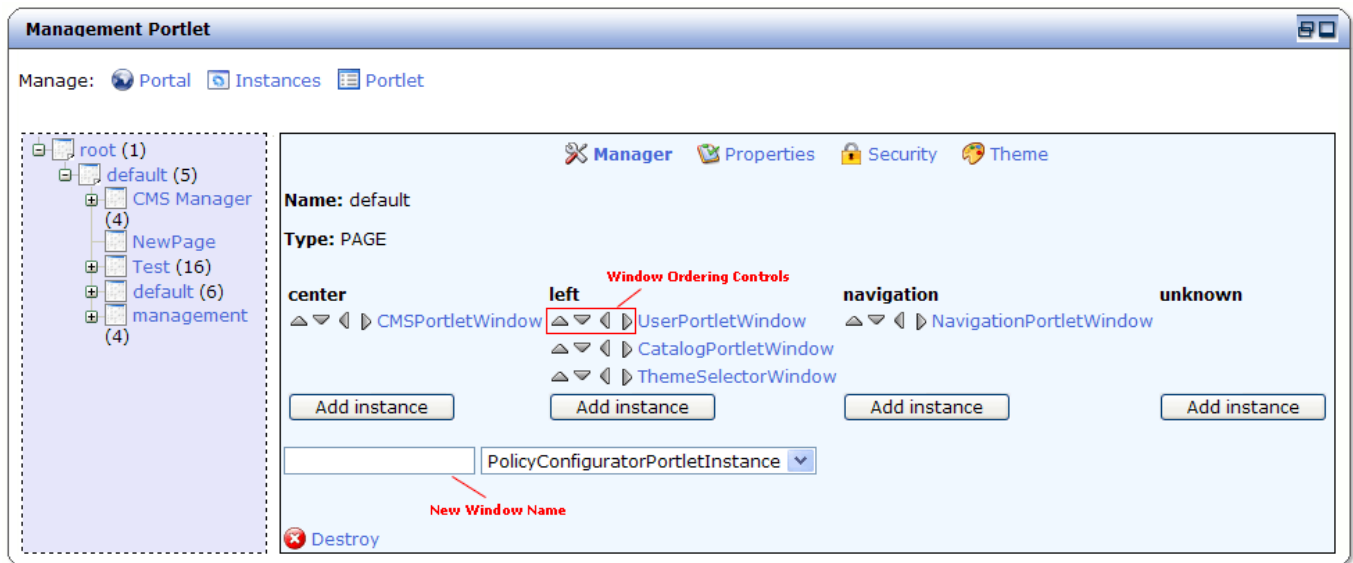
Selecting a portal window under the *Portal* tab, and selecting the *Theme* option, allows an administrator to modify the look-and-feel for the chosen portal page.

Note

From this screen, an administrator can elect to set all values to *emptyrenderer* so the portlet window is displayed with no decorations and appears to be part of the layout.

5.5. Managing portal pages

5.5.1. Creating portlet windows



Selecting a portal page, allows you to *Manage* the order in which windows appear and the layout column in which they will appear. Additionally, you can name and assign portlet instances on the selected page.

5.5.2. Securing portal pages

Selecting the *Security* option, allows an administrator to secure the portal page. On this screen, you can select and unselect page security settings for a given role.

5.5.3. Modifying the portal page appearance

Selecting the *Theme* option, allows an administrator to modify the look-and-feel for the chosen portal page.

6

CMS Portlet

Roy Russo <roy @ jboss dot org>

JBoss Portal packages a Web Content Management System capable of serving and allowing administration of web content. This chapter describes the CMS Portlet which is responsible for serving resources requested, the following chapter describes the CMSAdmin Portlet and all administration functionality.



6.1. Introduction

The CMS Portlet displays content from the file store inside a portlet window, or, in the case of binary content, outside of the portlet window altogether.

6.2. Features

The CMSPortlet handles all requests for all content types.

The methodology of serving content within the CMSPortlet, allows for some beneficial features, like:

1. Search-engine friendly URLs: `http://domain/[portal]/content/company.html`
2. Serve binaries with simple urls independant of the portal: `http://domain/content/products.pdf`
3. Deploy several instances of the CMSPortlet on any page and configure them to display different start pages.
4. Localization support: CMSPortlet will display content based on the user request locale, or display content using the default locale setting.

6.3. CMS Configuration

6.3.1. Portlet Configuration

The CMS Portlet allows an administrator to configure the initial start page displayed via its deployment descriptor, located in *jboss-portal.sar/portal-core.war/WEB-INF/portlet.xml*

```
<init-param>
  <description>Default path to index page.</description>
  <name>indexpage</name>
  <value>/default/index.html</value>
</init-param>
```

6.3.2. Service Configuration

JBoss Portal uses Apache Jackrabbit as its Java Content Repository implementation. Configuration of the service descriptor, allows for changing many of the variables associated with the service.

Here is the default configuration for the CMS repository found under *portal-cms.sar/META-INF-INF/jboss-service.xml* :

```
...
<attribute name="DoChecking">true</attribute>
<attribute name="DefaultContentLocation">portal/cms/conf/default-content/default/</attribute>
<attribute name="DefaultLocale">en</attribute>
<attribute name="RepositoryName">repotest</attribute>
<attribute name="HomeDir">${ jboss.server.data.dir}${ / }portal${ / }cms${ / }conf</attribute>
```

Below is a list of items found in the service descriptor and their definitions. Only items commonly changed are covered here and it is recommended you do not change any others unless you are very brave.

- **DoChecking:** Should the portal attempt to check for the existence of the repository configuration files and de-

fault content on startup?

- **DefaultContentLocation:** Location of the default content used to pre-populate the repository.
- **DefaultLocale:** Two-letter abbreviation of the default locale the portal should use when fetching content for users. A complete ISO-639 list can be found here [<http://ftp.ics.uci.edu/pub/ietf/http/related/iso639.txt>] .
- **HomeDir:** Location of configuration information for the repository when in 100% FileSystem store mode. Otherwise, its in the database. ;-)

Also note that there is an additional parameter that decides under which path the portal should serve requests for content, located in the same jboss-service.xml:

```
...
<mbean
  code="org.jboss.portal.core.cms.CMSObjectCommandMapper"
  name="portal:mapper=CMSObject"
  xmbean-dd="org/jboss/portal/core/cms/CMSObjectCommandMapper.xml">
    <attribute name="Prefix">content</attribute>
    <attribute name="TargetWindowRef">default.default.DefaultCMSPortletWindow</attribute>
    <depends optional-attribute-name="Mapper" proxy-type="attribute">portal:mapper=PrefixDelegating</de
    <depends optional-attribute-name="CMSService" proxy-type="attribute">portal:service=CMS</depends>
  </mbean>
  ...
```

- **Prefix:** This is the context path prefix that will trigger the portal to render content. By default, navigating to a URL such as `http://localhost:8080/[portal_context]/content/Test.PDF` will trigger the portal to display the PDF isolated from the portal pages. The path following the *Prefix* has to be absolute when fetching content.

6.4. Localization Support

The CMS Portlet now serves content based on the user's locale setting. For example: if a user's locale is set to Spanish in his browser, and he requests URL: `default/index.html` , the CMSPortlet will first try and retrieve the Spanish version of that file. If a Spanish version is not found, it will then try and retrieve the default language version set for the CMSPortlet.

CMSAdmin Portlet

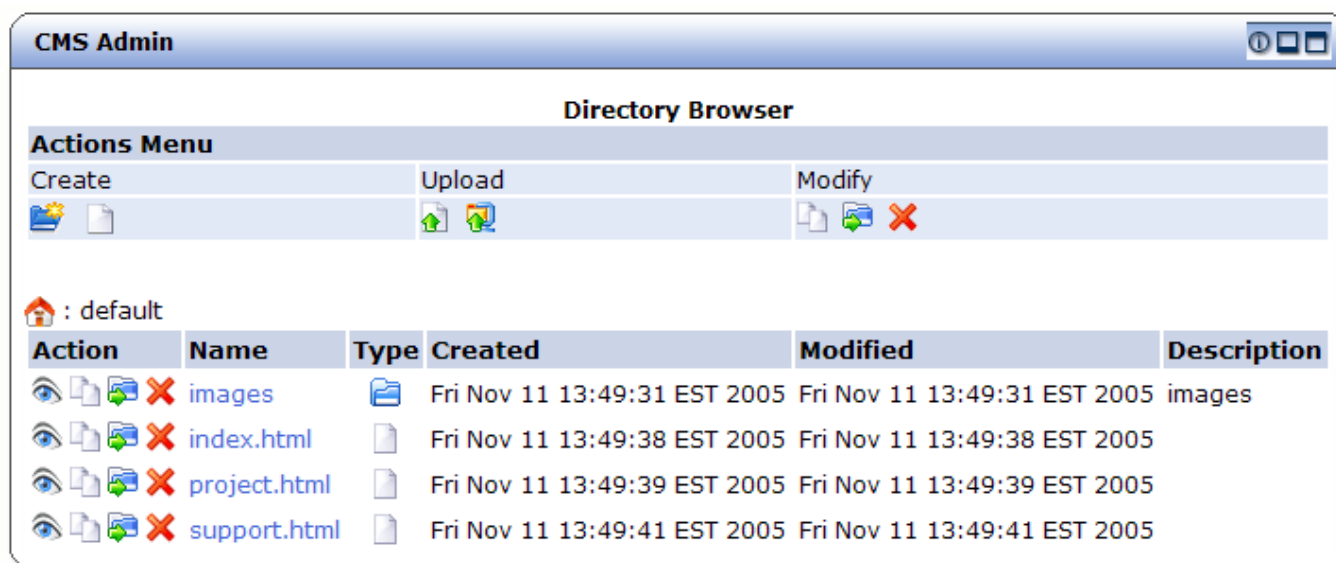
Roy Russo <roy @ jboss dot org>

7.1. Introduction

The CMSAdmin Portlet allows control over the content management system.








Viewing the CMSAdmin Portlet is accomplished by logging in as an admin (admin/admin) and navigating to the CMSManager page.

You should then be presented with a portlet that is similar to this:





It is important for a user to note the action icons used throughout the portlet and their meanings. The action options change depending on what type of resource the user is dealing with. All possible actions are listed here:

- Launches HTML WYSIWYG Editor window for HTML files. Launches upload dialog window for binary type files.
- Opens the copy file/folder dialog window.
- Opens the move file/folder dialog window.
- Launches HTML WYSIWYG Editor window.

-  - Opens the create folder dialog window.
-  - Opens the upload file dialog window.
-  - Opens the upload archive dialog window.
-  - Opens the delete confirmation dialog window.
-  - In the case of files, opens the file properties view. In the case of folders, opens the folder listing.
-  - Moves up the folder tree when clicked on.
-  - Expands directory tree.


Additionally, there are icons that help describe the types of resources present on the page:

-  - Denotes this resource as a file.
-  - Denotes this resource as a folder.


7.2. Features

This section describes common actions a user can perform from within the AdminCMS Portlet.

7.2.1. Viewing Directories

A user can list directory contents by either clicking on the  icon, or clicking on the directory's "DisplayName". All actions are possible from this screen.

7.2.2. Viewing File Properties

Clicking on the  icon or the "DisplayName" of a file brings up the File Properties page.

CMS Admin File Browser

🏠 : default : index.html

Actions Menu

✎ 📄 📁 ✖

📄 English

Edit	Type	Size	Version #	Created	Modified	Title
✎	text/html	2kb	1.0 ★	Fri Nov 11 13:49:39 EST 2005	Fri Nov 11 13:49:38 EST 2005	JBoss Portal

📄 Spanish

Edit	Type	Size	Version #	Created	Modified	Title
✎	text/html	0kb	1.0	Fri Nov 18 08:45:43 EST 2005	Fri Nov 18 08:45:43 EST 2005	espanol
✎	text/html	2kb	1.1 ★	Fri Nov 18 08:46:00 EST 2005	Fri Nov 18 08:46:00 EST 2005	es2

The File Properties window displays all the possible actions available to perform on a file.

Version and Locale Information are also contained on this screen. Note that any version labeled with the ★ is the current "live" version shown to users.

7.2.3. Copying Files/Directories

Clicking on the 📄 icon displays the copy file/directory dialog window.

CMS Admin Confirm Resource Copy

Source: /default/index.html


Destination: /

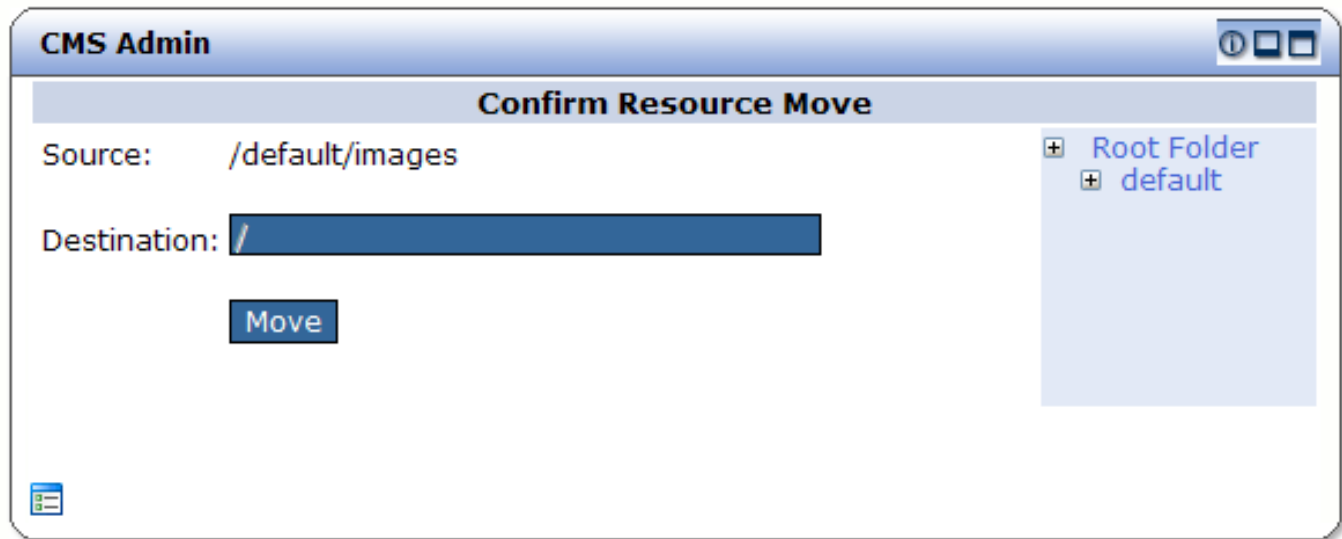
Copy


+ Root Folder
 + default

The copy resource window allows a user to copy files to any folder on the system, as well as copy whole directory trees to any directory on the system. A user can select which destination directory to copy the resource to, by using the directory browser. Clicking the + icon expands the directory tree. Clicking on the name of the directory within the tree, sets it as the destination directory for the copied resource.


7.2.4. Moving Files/Directories

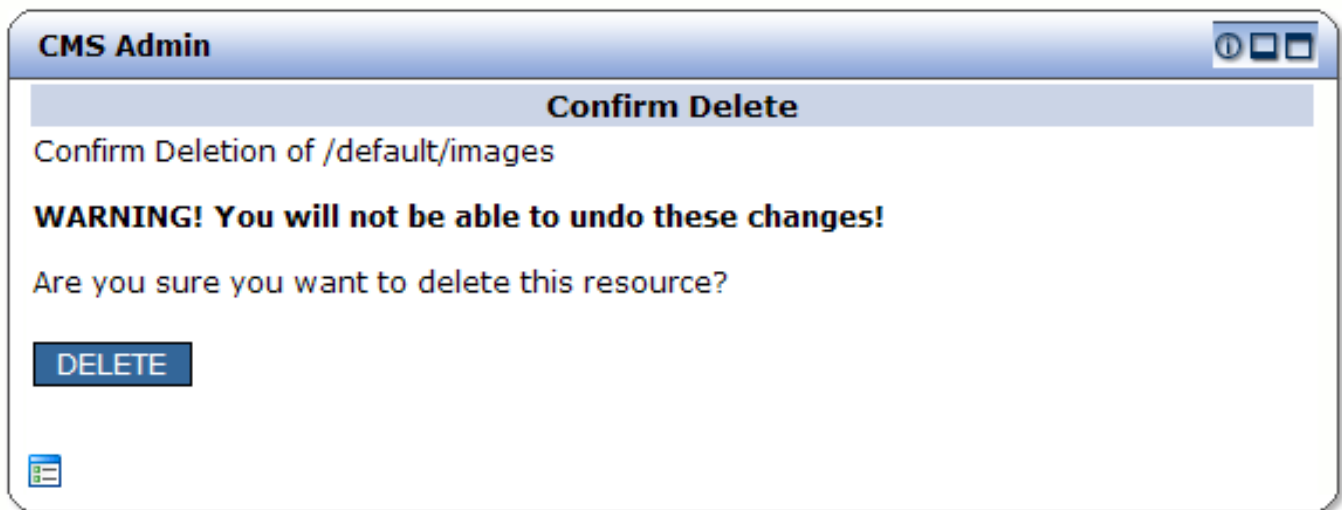
Clicking on the  icon displays the move file/directory dialog window.



The move resource window allows a user to move files to any folder on the system, as well as move whole directory trees to any directory on the system. A user can select which destination directory to move the resource to, by using the directory browser. Clicking the  icon expands the directory tree. Clicking on the name of the directory within the tree, sets it as the destination directory for the moved resource.

7.2.5. Deleting Files/Directories

Clicking on the  icon displays the delete file/directory confirmation window.




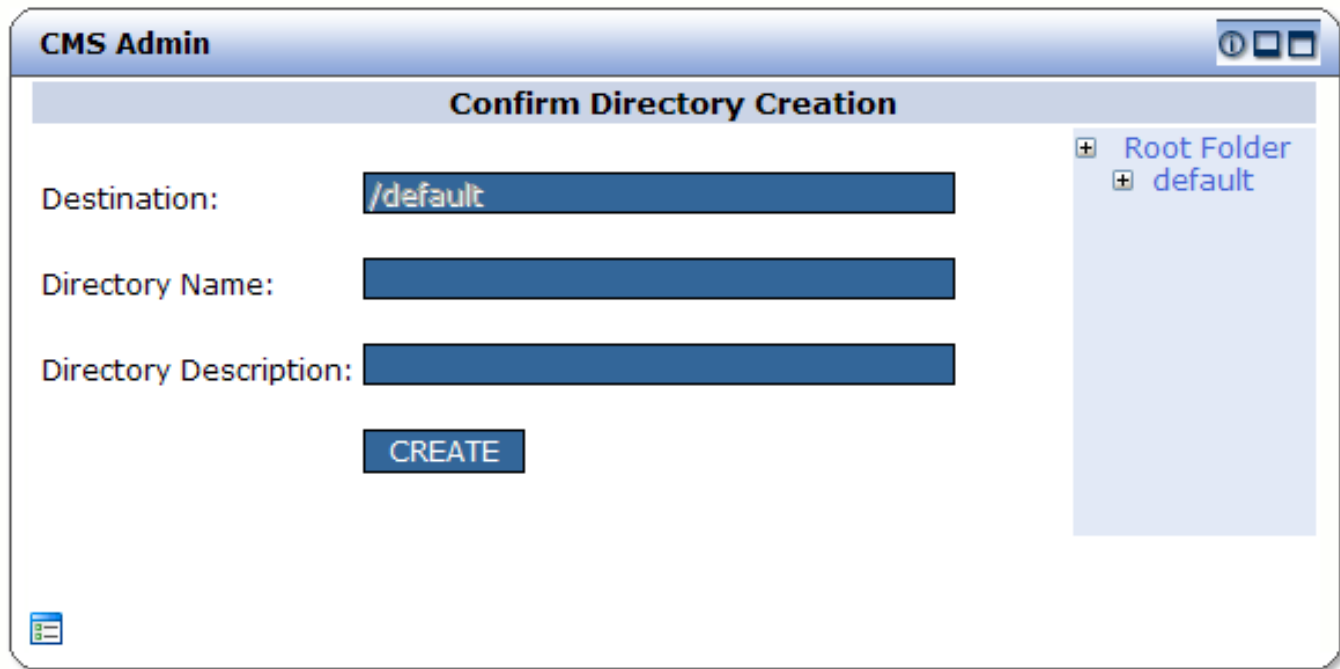
The delete resource confirmation window allows a user to delete a file, or a directory on the system. Note that deleting a directory, will delete the entire tree, so all directories under the deleted one, will also be deleted.

Warning

Currently, there is no way to retrieve deleted files/directories! Deleting a file or directory is permanent!

7.2.6. Creating Directories


Clicking on the  icon displays the create directory dialog window.



The image shows a dialog window titled "CMS Admin" with a subtitle "Confirm Directory Creation". It contains three text input fields: "Destination:" with the value "/default", "Directory Name:", and "Directory Description:". Below these fields is a "CREATE" button. On the right side, there is a tree view showing a folder structure: "Root Folder" (expanded) and "default" (subfolder). The window has standard window controls (minimize, maximize, close) in the top right corner and a small icon in the bottom left corner.


The create directory resource window allows a user to create a directory under chosen path. On this window, a user can specify a name for the new empty directory and assign it a description.

7.2.7. Creating Text/HTML Files

Clicking on the  icon displays the create file dialog window with the embedded WYSIWYG editor and directory browser.



HTML


An in-depth walk-through of the editor is beyond the scope of this document. However, the editor does contain help pages within it, that can be accessed by clicking the  icon.

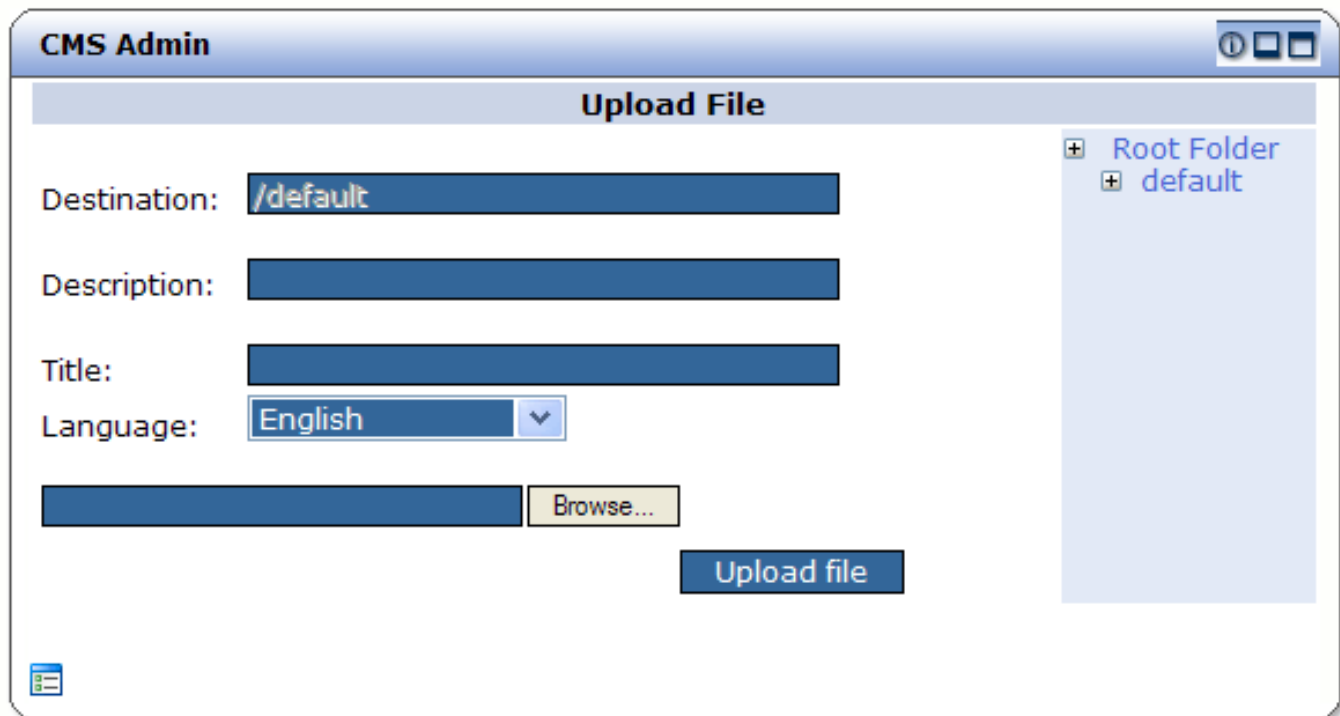
Note


It is important to note here that when creating links to images or other resources within the system, as user must use the relative file path to that resource. ie: images/hello.gif. Keep in mind at all times that the document base is `http://localhost/portal/` by default!

Additionally, a user can set a title for the file that will be used in the portlet title bar, and a language for the file, used in serving localized content.


7.2.8. Uploading Files

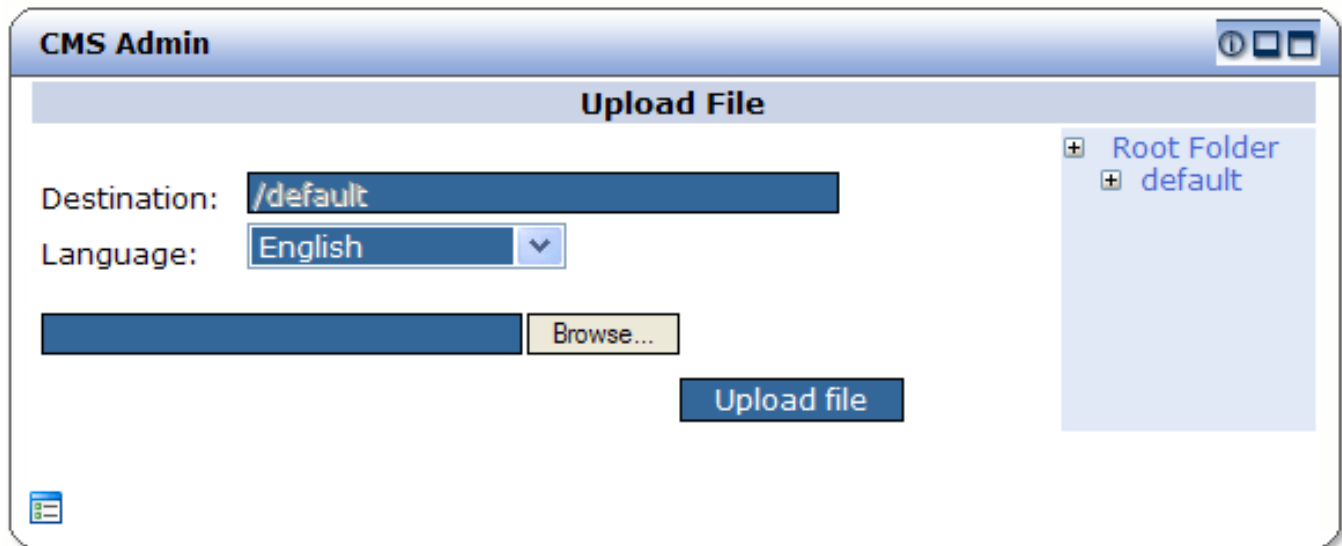
Clicking on the  icon displays the upload file dialog window.




The upload file window allows a user to upload files to any directory on the system. The upload process will work on files up to 1GB and of all types. A user can select which destination directory to upload the resource to, by using the directory browser. Clicking the  icon expands the directory tree. Clicking on the name of the directory within the tree, sets it as the destination directory for the uploaded resource. Additionally, a user can set a title for the file that will be used in the portlet title bar, and a language for the file, used in serving localized content.


7.2.9. Uploading Archives

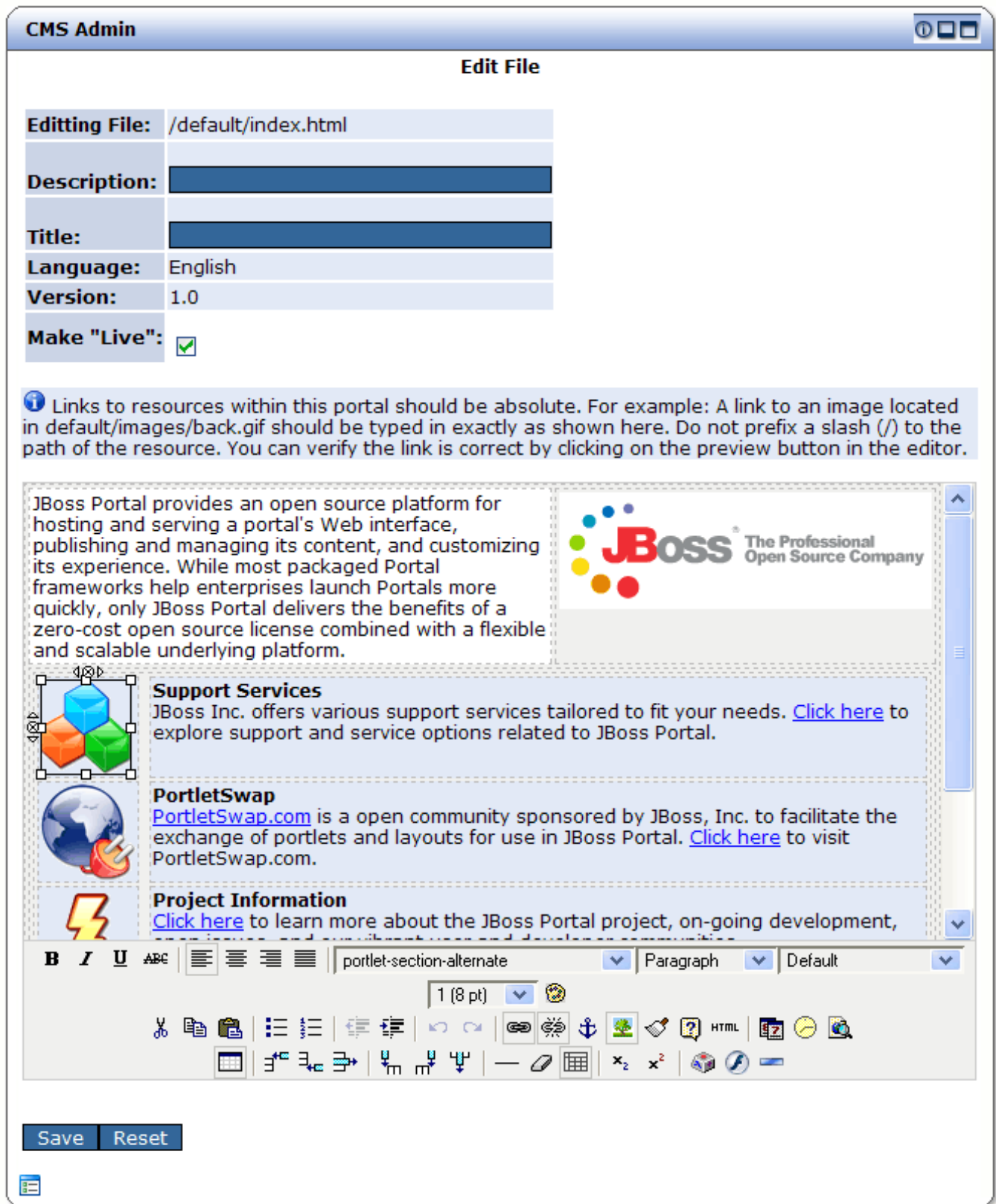
Clicking on the  icon displays the upload archive dialog window.





The upload archive window allows a user to upload archives to any directory on the system. The system will then explode the archive, create versions, and place all the files in the repository. A user can select which destination directory to upload the resource to, by using the directory browser. Clicking the  icon expands the directory tree. Clicking on the name of the directory within the tree, sets it as the destination directory for the uploaded resource. Additionally, a user can set a language for the archive files, used in serving localized content.

7.2.10. Editing Text/HTML Files

Clicking on the  icon displays the edit file dialog window with the embedded WYSIWYG editor and directory browser.



The edit file window allows a user to edit a text or HTML file using the embedded WYSIWYG HTML editor. The editor is a fully-functional HTML editor with a myriad of HTML functions. It also includes a preview  button and a source view  button.

A user may specify at this point if he would like to make the new edit "live", or available in production. Additionally, a user can set a title for the file that will be used in the portlet title bar.

User and Role Administration

Thomas Heute <theute@jboss.org>

Roy Russo <roy@jboss.org>

8.1. User Portlet

1. Introduction

The user portlet is dedicated to create and manage users and their profiles. This portlet is accessible immediately on the default portal homepage. Also, once logged in, you will notice how it changes to allow the user and administrative functions.



2. Functionalities

Managing users using the user module consists in:

1. Allowing a user to register a new profile with a login and password
2. Allowing a user to log in the system and be recognized as member of user groups
3. Allowing a user to update a profile with more information like his real name, instant messenger informations...
4. Allowing an administrator to see the list of users, from there he can assign roles to a user
5. Allowing an administrator to assign roles to a user

6. Allowing a user to logout

3. Configuration

The following xml block is the standard configuration for the UserPortlet found in *portal-core.war/WEB-INF/portlet.xml* :

```
<portlet>
  <portlet-name>UserPortlet</portlet-name>
  <portlet-class>org.jboss.portal.core.portlet.user.UserPortlet</portlet-class>
  <supported-locale>en</supported-locale>
  <supported-locale>fr</supported-locale>
  <resource-bundle>Resource</resource-bundle>
  <supports>
    <mime-type>text/html</mime-type>
    <portlet-mode>VIEW</portlet-mode>
  </supports>
  <portlet-info>
    <title>User portlet</title>
  </portlet-info>
  <init-param>
    <description>Whether we should use ssl on login and throughout the Portal. 1=yes;0=no</description>
    <name>useSSL</name>
    <value>0</value>
  </init-param>
  <init-param>
    <description>Subscription mode</description>
    <name>subscriptionMode</name>
    <!--      <value>emailVerification</value>-->
    <value>automatic</value>
  </init-param>
  <init-param>
    <description>Domain of your website for email verification.</description>
    <name>emailDomain</name>
    <value>JBoss.com</value>
  </init-param>
  <init-param>
    <description>Email displayed in the TO field</description>
    <name>emailFrom</name>
    <value>jbossportal@example.com</value>
  </init-param>
  <init-param>
    <description>Default role of registered users</description>
    <name>defaultRole</name>
    <value>Users</value>
  </init-param>
</portlet>
```

The following attributes can be modified in the xml descriptor:

1. useSSL

Allows for user logins to be passed thru a SSL.

a. 0

Set to zero to disable.

b. 1

Set to 1 to enable. You must have SSL configured properly in tomcat for this to work.

2. `subscriptionMode`

a. `automatic`

The user can register and is automatically enabled

b. `emailVerification`

The user is disabled until he clicks on a link sent to his email address.

3. `emailDomain`

Your domain name or the name of your website for the email verification form text.

4. `emailFrom`

Email address that will appear in the "From" header when the email verification is sent.

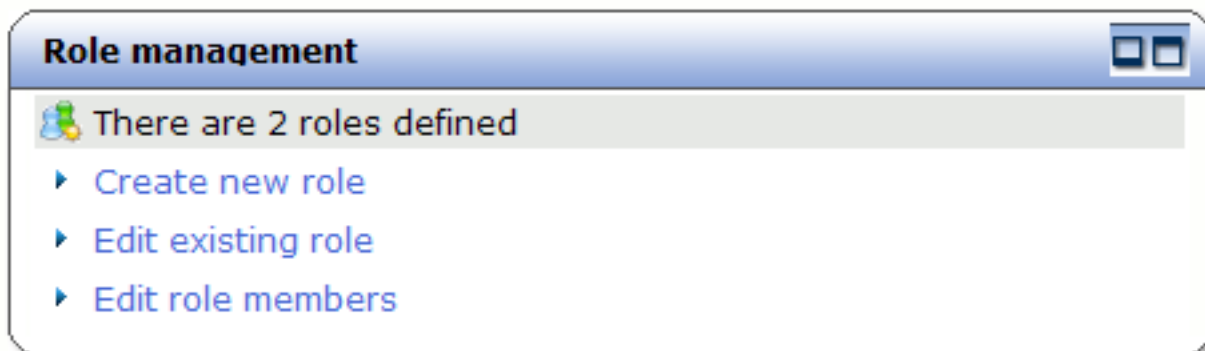
5. `defaultRole`

Default role assigned to new users

8.2. Role Portlet

1. Introduction

The role portlet is dedicated to create and edit roles. A role will be used to grant different permission level to different portlets. A user can have several roles (for example he can be an administrator of a category of forum but only a reader on another category)



2. Functionalities

2.1. Create a role

To create a new role, you just need to define a short name that will be used for reference, and a display name for displaying to the user, for example `admin` would be a good name for the display name `Administrators`, changing the display name will not affect the security rules

2.2. Edit a role

While editing a role, you just need to pick an existing role then change the display name.

2.3. Editing Role Members

Allows for an administrator to search and modify the members' assigned roles.

Forums Portlet

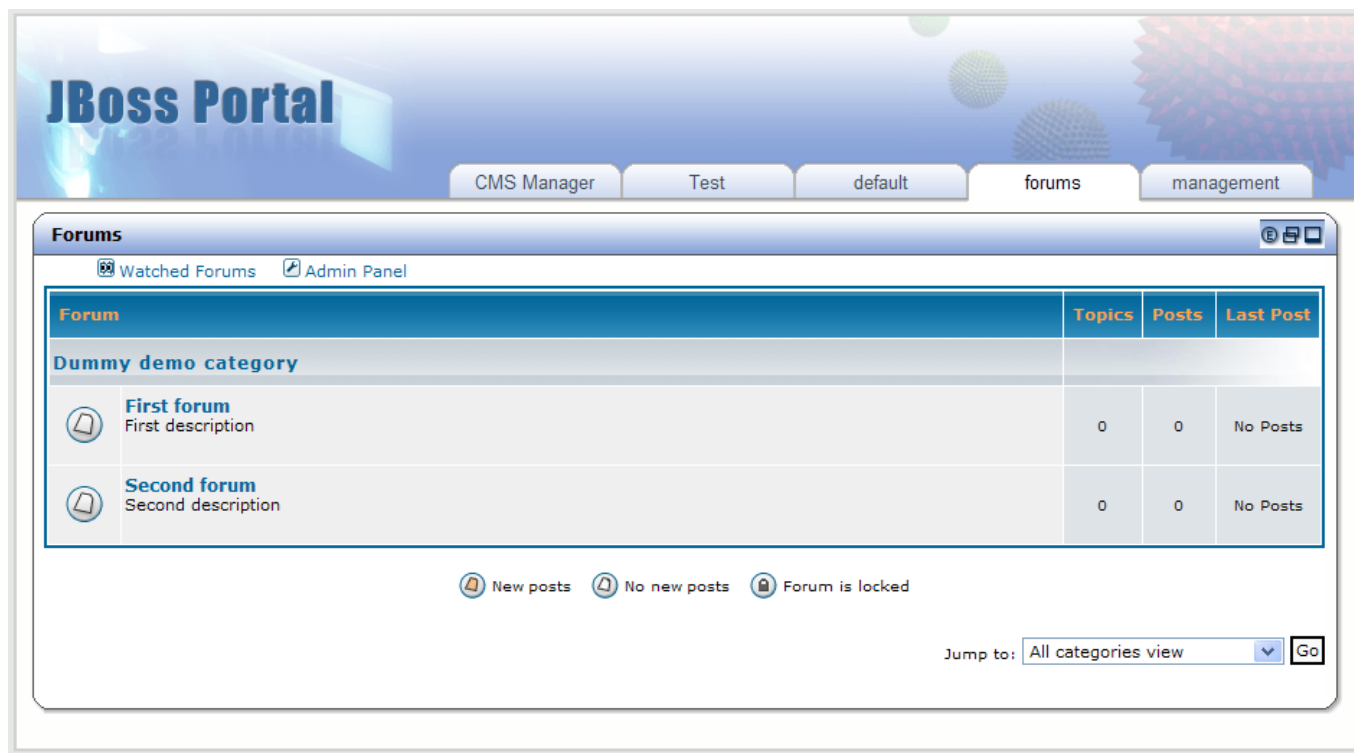
Thomas Heute <theute@jboss.org>

9.1. Introduction

Warning

The forums portlet is GPL licensed

The forums portlet is a port of the phpBB forums as a Java portlet. It is packaged independently of the core, so you can easily use it or not depending on your own needs.



Above is the main window displayed by default to any user. It lists all the forums classified by categories. It is possible to see how many topics and posts were written for each forum and the date and user of the last post. All those categories and forums can be configured if the user has the correct privileges. The next image shows the main administration interface available to users with the correct credentials.

JBoss Portal

CMS Manager Test default **forums** management

Forums [Watched Forums](#) [Admin Panel](#)

Forum Administration

From this panel you can add, delete, edit, re-order and re-synchronise categories and forums

Forum Administration					
Dummy demo category			Edit	Delete	Move up Move down
First forum	0	0	Edit	Delete	Move up Move down
Second forum	0	0	Edit	Delete	Move up Move down

Jump to:

9.2. Functionalities

User features:

- See the list of forums
- Post a new topic
- Read a topic
- Reply to an existing post
- Fast-reply to an existing post in the same page as the thread
- Email notification
- Quote a existing forum posts

Admin features:

- Create a new category of forum
- Edit the name of a category

- Delete a category and move the content to another category
- Create a new forum
- Edit the name and description of a forum
- Delete a forum and move the content to another forum
- Classify categories
- Classify forums
- Edit forum posts

9.3. Installation

If you are deploying from **binary** , just move portal-forums.ear in to your deploy directory.

If you are deploying from **source** :

To install forums, you need to go to the directory `forums` and type `sh build.sh deploy` it will create a file `portal-forums.ear` and copy it to `$JBOSS_HOME/server/default/deploy` . If JBoss is already running you have nothing to do but to go to a page where the forums should be displayed (see your configuration).

To have the mail notification working, make sure that you correctly configure the mail service with an existing SMTP account in the file:
`$PORTAL_HOME/core/src/resources/portal-server-war/WEB-INF/jboss-service.xml`

9.4. Configuration

In `$FORUMS_HOME/src/resources/portal-forums-war/WEB-INF/portlet.xml` you can configure the following options:

- `floodInterval` : Minimal time in seconds between two messages by a user.
- `fromAddress` : Email address appearing in the `From` field of notification emails.

9.5. Security

You can restrict access to the forums for certain roles, to do so edit the file `$FORUMS_HOME/src/resources/portal-forums-war/WEB-INF/jboss-portlet.xml` . You should see the existing part:

```
<scheme>
<domain></domain>
<item>
<path></path>
```

```
<permission>
<permission-name>Add</permission-name>
<role-name>Users</role-name>
</permission>
<permission>
<permission-name>Admin</permission-name>
<role-name>Admins</role-name>
</permission>
<!-- For non logged users -->
<permission>
<permission-name>Read</permission-name>
<role-name></role-name>
</permission>
</item>
</scheme>
```

This means that a user with role Users has the permission to add posts in forums, a user with role Admins has the permissions to Admin anything, while an anonymous user (not logged on), can only read.

If you want users to only view a category named "myCategory" to a certain role "myRole", here is an `item` that you can add:

```
<item>
<path>/myCategory</path>
<permission>
<permission-name>ReadCategory</permission-name>
<role-name>myRole</role-name>
</permission>
</item>
```

10

Security

Martin Holzner <mholzner@novell.com>

10.1. Introduction

New in version 2.2, a role based access control model was introduced into the portal. As before, all portal resources are not secured by default. However, they can be protected via security constraints that tie a role and allowed actions to the resource. Portal resources are portlets, instances of those portlets, portlet windows, portal pages, and portals. As long as there is no security constraint defined for a resource, full access is granted to every user. Once a constraint is defined, only the allowed actions from that constraint are allowed for the roles defined in the constraint. All other access is denied. Access is checked from the top down, so that if a user has no access privileges to a portal, none of the pages are accessible either, and so on.

There are two ways to define security constraints for portal resources. Constraints can be defined in the relevant deployment descriptor, or via a management UI that is provided to admins, once the portal is up and running. To define a constraint that is active immediately after the portal is deployed, once can add a security constraint tag in the relevant descriptor.

The relevant descriptors are:

- jboss-portlet.xml
- *-object.xml

Here is an example of a constraint to secure access to a page and only allow members of the 'Admin' role to 'view' the page (taken from a *-object.xml descriptor):

```
<deployment>
  <if-exists>keep</if-exists>
  <parent-ref>default</parent-ref>
  <page>
    <page-name>management</page-name>
    <window>
      <window-name>NavigationPortletWindow</window-name>
      <!-- more window defs here -->
    </window>
    <security-constraint>
      <policy-permission>
        <role-name>Admin</role-name>
        <action-name>view</action-name>
      </policy-permission>
    </security-constraint>
  </page>
```

```
</deployment>
```

To complete the example, here is a snippet from a `jboss-portlet.xml` descriptor constraining access to the `PolicyConfiguratorPortlet` to the `Admin` role, allowing only to view the portlet.

```
<portlet>
  <portlet-name>PolicyConfiguratorPortlet</portlet-name>
  <security-constraint>
    <policy-permission>
      <role-name>Admin</role-name>
      <action-name>view</action-name>
    </policy-permission>
  </security-constraint>
</portlet>
```

To achieve the same results after the portal was deployed, one can login with an `Admin` role, and navigate to the management tab. There navigate to the desired portal resource in the tree, and click the security menu image. Pick from the offered Roles and actions and save your choice. The changes will immediately be active. Note that you can provide more than one security constraint per portal resource. This allows you to provide different roles with different access privileges for the same resource.

10.2. Securing Portlets

In order to not have to limit access to each instance or window of a particular portlet, the portal offers the possibility to secure the portlet itself. All children (instances and/or windows) will then automatically be secured as well. To secure a portlet add a security constraint to the `jboss-portlet.xml` entry for the portlet in question, and provide the desired role and action information. Of course, this function can be delayed until the portlet is deployed. The portlet can also be secured via the management UI.

10.3. Securing Portlet Instances

To allow to secure all portlet windows that point to the same portlet instance with one constraint, the portal allows to define a constraint for a portlet instance. The constraint can be provided via the instance definition in the `*-object.xml` descriptor, or after the portlet is deployed via the management UI. The result is the same: all portlet windows of this instance will be protected.

10.4. Securing Portlet Windows

To secure an individual portlet window, the `*-object.xml` descriptor allows to provide a security constraint for any individual window.

10.5. Securing Pages

Analogous to the previous portal resources, pages can be secured via a security constraint as well.

10.6. Securing Portals

If an entire portal needs to be secured, one security constraint placed for the portal will take care of that. All pages, and portlet windows will be secured automatically as well.

Clustering Configuration

Roy Russo <roy@jboss.org>

Julien Viet <julien.viet@jboss.com>

This section covers configuring JBoss Portal to function in a clustered environment.

Warning

As of JBoss Portal 2.2.1RC2, the CMS feature is not reliably clustered. Look for future versions for full clustering capabilities in the CMS.

11.1. Introduction

JBoss Portal leverages various clustered services that are found in JBoss Application Server. This section briefly details how each is leveraged:

- **JBoss Cache:** Used to replicate data among the different hibernate session factories that are deployed in each node of the cluster. It is also used by the authorization framework to replicate data among the different policies.
- **JBoss HA Singleton:**
 1. Used to make the deployer a singleton on the cluster, in order to avoid concurrency issues when deploying the various *-object.xml files. Without that, each node would try to create the same objects in the database.
 2. Used with JCR. The jackrabbit server is not able to run in a cluster by itself, therefore we make a singleton on the cluster. This provides HA-CMS, which is similar to the current HA JBossMQ provided in JBoss AS.
- **HA-JNDI:** Used to replicate a proxy that will talk to the HA CMS on the cluster.

Note

JBoss Clustering details can be found in the Wiki [<http://wiki.jboss.org/wiki/Wiki.jsp?page=JBossHA>] or the clustering documentation [<http://www.jboss.com/products/jbossas/docs>] .

11.2. Configuring from Source

If you have obtained the JBoss Portal source files, you can build JBoss Portal to work on a cluster in a simple pro-

cess:

- Edit the file *build/local.properties* and set the property *portal.clustered* to true, then you build JBoss Portal as you normally would .

```
# Build the portal for single or clustered environment
portal.clustered=false
```

- Deploy the jboss-portal.sar and the appropriate datasource descriptor to *JBOSS_HOME/server/all/deploy/**

11.3. Configuring from Binary

If you have the binary version of JBoss Portal, you will need to edit several files manually that would normally be edited for you by the build. The list below details the files and code blocks that need to be modified in each:

- *jboss-portal.sar/META-INF/jboss-service.xml*

```
<!--
| Uncomment in cluster mode : have the deployment of objects run as a clustered singleton
| @portal.single.xml.close@
<mbean
  code="org.jboss.ha.singleton.HASingletonController"
  name="portal:service=Controller,target=ObjectDeploymentFactory">
  <depends>jboss:service=${jboss.partition.name:DefaultPartition}</depends>
  <depends>portal:deploymentFactory=Object</depends>
  <attribute name="TargetName">portal:deploymentFactory=Object</attribute>
  <attribute name="TargetStartMethod">registerFactory</attribute>
  <attribute name="TargetStopMethod">unregisterFactory</attribute>
</mbean>
@portal.single.xml.open@
-->
```

- *jboss-portal.sar/portal-cms.sar/META-INF/jboss-service.xml*

```
<!--
| Uncomment in clustered mode : replicated cache for hibernate
| @portal.single.xml.close@
<mbean
  code="org.jboss.invocation.jrmp.server.JRMPProxyFactory"
  name="portal:service=ProxyFactory,type=CMS">
  <depends optional-attribute-name="InvokerName">jboss:service=invoker,type=jrmp</depends>
  <attribute name="TargetName">portal:service=CMS</attribute>
  <attribute name="ExportedInterfaces">org.jboss.portal.cms.ha.HASingletonInvokerMBean$Proxy</attribute>
  <attribute name="InvokeTargetMethod">true</attribute>
  <attribute name="ClientInterceptors">
    <interceptors>
```

```

        <interceptor>org.jboss.proxy.ClientMethodInterceptor</interceptor>
        <interceptor>org.jboss.invocation.InvokerInterceptor</interceptor>
    </interceptors>
</attribute>
</mbean>

<mbean
    code="org.jboss.portal.cms.ha.HASingletonInvoker"
    name="portal:service=HASingletonInvoker,type=CMS">
    <depends>jboss:service=${jboss.partition.name:DefaultPartition}</depends>
    <attribute name="RetryWaitingTimeMS">2000</attribute>
    <attribute name="MaxRetries">5</attribute>
    <attribute name="JNDIName">MyServiceInvokeTarget</attribute>
    <attribute name="JNDIProperties">
        java.naming.factory.initial=org.jnp.interfaces.NamingContextFactory
        java.naming.factory.url.pkgs=org.jboss.naming:org.jnp.interfaces
        java.naming.provider.url=${jboss.bind.address:localhost}:1100
        jnp.disableDiscovery=false
        jnp.partitionName=${jboss.partition.name:DefaultPartition}
        jnp.discoveryGroup=${jboss.partition.udpGroup:230.0.0.4}
        jnp.discoveryPort=1102
        jnp.discoveryTTL=16
        jnp.discoveryTimeout=5000
        jnp.maxRetries=1
    </attribute>
    <depends optional-attribute-name="Callback" proxy-type="attribute">portal:service=CMS</depends>
    <depends optional-attribute-name="ProxyFactory">portal:service=ProxyFactory,type=CMS</depends>
</mbean>
@portal.single.xml.open@
-->

```

- *jboss-portal.sar/portal-server.war/WEB-INF/web.xml*

```

<!--
    | Uncomment in clustered mode : use http session replication
    @portal.single.xml.close@
<distributed/>
@portal.single.xml.open@
-->

```

- *jboss-portal.sar/conf/hibernate/instance/hibernate.cfg.xml*, *jboss-portal.sar/conf/hibernate/portal/hibernate.cfg.xml*, and *jboss-portal.sar/conf/hibernate/user/hibernate.cfg.xml*

```

<!--
    | Uncomment in clustered mode : use transactional replicated cache
    @portal.single.xml.close@
<property name="cache.provider_class">org.jboss.portal.core.hibernate.JMXTreeCacheProvider</property>
<property name="cache.object_name">portal:service=TreeCacheProvider,type=hibernate</property>
@portal.single.xml.open@
-->

<!--
    | Comment in clustered mode
    @portal.clustered.xml.close@

```

```
<property name="cache.provider_class">org.hibernate.cache.EhCacheProvider</property>
@portal.clustered.xml.open@
-->

<!-- Force the dialect instead of using autodetection -->
<!--
<property name="dialect">org.hibernate.dialect.PostgreSQLDialect</property>
-->
```

After modifying the above files, you can deploy the jboss-portal.sar and the appropriate datasource descriptor to JBOSS_HOME/server/all/deploy/*

Troubleshooting and FAQ

Roy Russo <roy@jboss.org>

12.1. Troubleshooting and FAQ

Installation / Configuration

- I am seeing "ERROR [JDBCExceptionReporter] Table not found in statement" in the logfile on first boot. What is this?
- I want to do a clean install/upgrade over my existing one. What are the steps?
- Is my database vendor/version combination supported?
- How do I force the Hibernate Dialect used for my database?

CMS

- How do I change the CMS repository configuration?
- On reboot, the CMS is complaining about a locked repository.
- I created a file in the CMSAdmin. How do I view it?

Miscellaneous

- Is there a sample portlet I can look at to learn about portlet development and JBoss Portal deployments?

I am seeing "ERROR [JDBCExceptionReporter] Table not found in statement" in the logfile on first boot. What is this?

Ignore this error. It is used by the portal to create the initial database tables. On second boot, you should not see them at all.

I want to do a clean install/upgrade over my existing one. What are the steps?

- Shut down JBoss AS

- Delete JBOSS_HOME/server/default/data/portal
- Delete all JBoss Portal tables from your database
- Start JBoss AS.

Is my database vendor/version combination supported?

See Section 1.4

How do I force the Hibernate Dialect used for my database?

See Section 3.3

How do I change the CMS repository configuration?

There are 3 supported modes: 100% DB (default), 100% Filesystem, and Mixed (Blobs on the Filesystem and metadata in the DB). You can see configuration options here: Section 3.4

On reboot, the CMS is complaining about a locked repository.

This occurs when JBoss AS is improperly shutdown or the CMS Service errors on startup. To remove the lock, shutdown JBoss, and then remove the file under JBOSS_HOME/server/default/data/portal/cms/conf/.lock.

I created a file in the CMSAdmin. How do I view it?

Using the default configuration, the path to the file in the browser would be: <http://localhost:8080/portal/content/path/to/file.ext>. Note that all requests for cms content must be prepended with /content and then followed by the path/to/the/file.gif as it is in your directory structure.

Is there a sample portlet I can look at to learn about portlet development and JBoss Portal deployments?

Sample HelloWorld Portlets can be found at PortletSwap.com [<http://www.portletswap.com>] .