

CmsImport Readme

Release 1.0 of January, 2008

© 2008 xanders GmbH

Table of contents

1.	Preface	3
2.	Requirements	3
3.	Installation.....	3
3.1.	Setup of MySQL for Alfresco.....	3
3.1.1.	MySQL Database Connector.....	3
3.1.2.	Create Database	3
3.2.	Alfresco on Tomcat	4
3.3.	Installation and configure of the RMI extension	4
3.4.	Build and deployment of the CmsImport package	4
4.	Screenshots of the portlet running on JBoss Portal Server.....	6
5.	Licence	7

1. Preface

The main purpose is to demonstrate the connectivity between a portal (eg. JBoss Portal Server, JSR 168 specs) and a content management system (eg. Alfresco, JSR 170 specs). The portlet used in the portal can also state as an example how to realize a portlet based on JSF/Facelets and the JBoss Richfaces packages.

The software was developed by Joerg-A. Stoehr. Joerg-A. Stoehr is CEO of xanders (www.xanders.de), a small Germany based consultancy with the focus on realizing enterprise applications.

We welcome your questions or feature requests. Please address them to info@xanders.de.

All product names used in this document are trademarks, which are in the property of their respective owners.

2. Requirements

Our approach is based on the fact that the portal and the content management system are running on different servers. This means to run the CmsImport software the fulfil of the following requirements is necessary:

- The whole software stack is based on Java, so J2SE Runtime Environment (JRE) or the JDK (5.0 or higher) must be installed on the computer.
- For our portal we are the JBoss Portal server (JBoss Portal 2.6.3.GA) is used. The JBoss Portal Server can be downloaded at <http://labs.jboss.com/jbossportal/docs/index.html>.
- We are using the Alfresco CMS running in a Tomcat server. The Tomcat server can be downloaded at <http://tomcat.apache.org>, the Alfresco package (alfresco community war file) is available at http://wiki.alfresco.com/wiki/Download_Alfresco_Community_Network.
- Our approach uses the MySQL Community Server database to store the CMS content. The database is available at <http://dev.mysql.com/downloads>, to establish the connection between Alfresco and MySQL the Connector/J 5.1 is used which is available at <http://dev.mysql.com/downloads/connector/j/5.1.html>
- To build the CmsImport package the tool Ant 1.7, available at <http://ant.apache.org/bindownload.cgi>, has to be used. For editing the package the Eclipse workbench, available at <http://www.eclipse.org>, is a good chance.

3. Installation

To run the CmsImport software successfully several steps have to be done:

3.1. Setup of MySQL for Alfresco

3.1.1. MySQL Database Connector

You may need to add or update the MySQL Java Connector library (required for WAR bundle):

- Download the MySQL Java Database Connector <http://dev.mysql.com/downloads/connector/j/>
- Copy the JAR file into <container>/common/lib or <container>/server/default/lib

3.1.2. Create Database

Navigate to the '<alfresco>/extras/databases/mysql' folder.

Setup the Alfresco database and user manually by loading the 'db_setup.sql' file into MySQL, for example, by running the command:

```
mysql -u root -p <db_setup.sql
```

This creates a MySQL database named 'alfresco' with a user account and password of 'alfresco'. If this fails, it may be because the MySQL service is not running or that the mysql command cannot be found.

3.2. Alfresco on Tomcat

After the download of the Alfresco the package has to be unpacked into a temporary directory. It consists of several directories and files, that have to be handled in the following way:

- Copy the alfresco.war to <tomcat/webapps> deploy directory.
- Copy the alf_data directory to where you want.
- Copy the extensions\extension\custom-hibernate-dialect.properties to <tomcat/shared/classes/alfresco/extension>. In the custom-hibernate-dialect.properties you can change the database dialect, if you want other the hsql. To convert the default installation to MySQL, you simple need to modify these 2 files in the following way:
 - **custom-repository.properties**
 - comment out the HSQL connection lines (using #)
 - uncomment the MySQL connection lines and adjust as appropriate
 - **custom-hibernate-dialect.properties**
 - comment out the HSQL line using #
 - uncomment the MySQL line
- Copy the extensions\extension\custom-repository.properties to <tomcat/shared/classes/alfresco/extension>. In this property file you can overwrite the default properties. Uncomment dir.root and set it to your alf_data location. Here you can also change your database driver and settings.
- Copy the custom-repository-context.xml to <tomcat/shared/classes/alfresco/extension>. If you use windows, copy the bin\Win32NetBIOS.dll to <tomcat/bin>.
- After Alfresco is running on Tomcat import the file **cms_import.acp** located in the <project>/src/resources directory into the Alfresco workbench(Import in the Alfresco Administration Console)

For further information about installing Alfresco look in the file readme_war.txt of the Alfresco distribution.

If you have problems to start the server, because of a very restricted unix installation, you can switch off the ftpserver and the CIFS Server in the <tomcat>\webapps\alfresco\WEB-INF\classes\alfresco\file-servers.xml

3.3. Installation and configure of the RMI extension

- Download the rmi-extension (extension-jcr-rmi-1.0.0-src.zip) from <http://forge.alfresco.com/projects/jcr-rmi> and unpack them into a temporary directory
- Copy extension-jcr-rmi\lib\jcr-rmi-1-impl-classes.jar to <tomcat/common/lib>
- Copy extension-jcr-rmi\build\dist\extension-jcr-rmi.jar to <tomcat/common/lib>
- Copy extension-jcr-rmi\source\web\jsp\jcrRmiTest.jsp to <tomcat/webapp/alfresco/jsp>
- Copy the files in extension-jcr-rmi\config\alfresco\extension to <tomcat/webapp/alfresco/WEB-INF/classes/alfresco/extension>
- Comment out the following line from extension/jcr-rmi-context.xml:
<import resource="classpath:alfresco/jcr-context.xml" />
- Change the jcr-rmi.properties for your environment (eg. servername and port to your rmiregistry). You can use the alfresco avm-rmi-server on port 50500, so you don't need a external rmiregistry. If you want to use your own rmiregistry, set the servername and port to your rmiregistry installation.
- Move jcr-1.0.jar from <tomcat\webapps\alfresco\WEB-INF\lib> to <tomcat/common/lib>
- Download the jackrabbit-jcr-rmi-1.3.3.jar from <http://jackrabbit.apache.org/downloads.cgi/>
- Copy the jackrabbit-jcr-rmi-1.3.3.jar to <tomcat/common/lib>

After configuration the rmiserver can be used to get information from the repository. It's a good way to test with <http://localhost:8080/alfresco/faces/jsp/jcrRmiTest.jsp>

More information concerning the RMI extensions can be retrieved from http://wiki.alfresco.com/wiki/JCR-RMI_Extension.

3.4. Build and deployment of the CmsImport package

The portlet is based on JSF/Facelets and the Richfaces packages.

Before building the CmsImport package some adjustments have to be done. In the file **build.properties** the path of the JBoss Portal Server has to be entered, the compiler options can be adjusted to the individual needs. To see all possible Ant targets, move at command line level to the **<project>/build** directory and type **ant**.

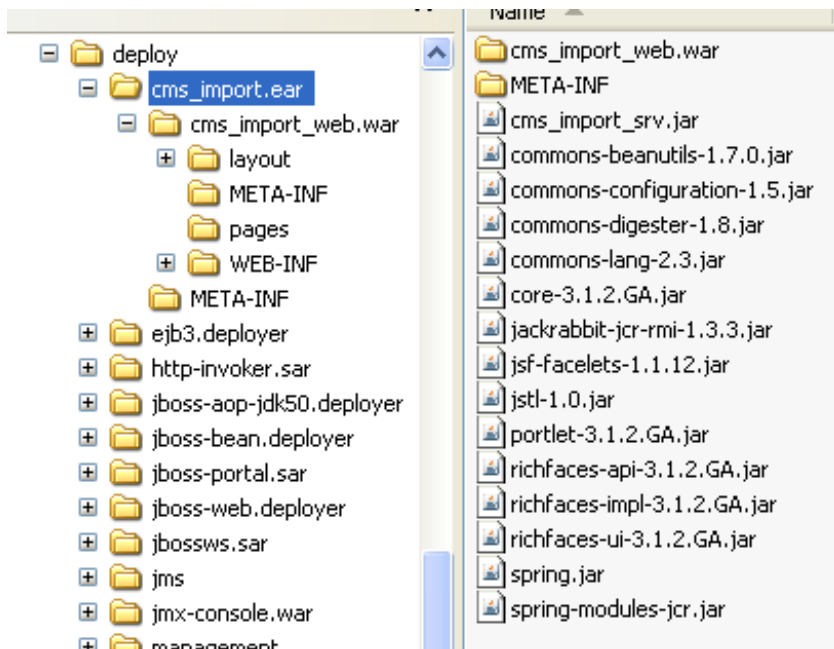
```
usage:
[echo] Usage: CmsImport build file targets description
[echo]
[echo] Preconditions:
[echo]   - JDK 1.5.0 (version 5.0)
[echo]   - JAVA_HOME environment variable
[echo]   - ANTI 1.7.0
[echo]   - ANTI_HOME environment variable
[echo]   - ANTI_OPTS environment variable
[echo]
[echo] Execute 'ant <desired_target>' for target execution
[echo]
[echo] Possible targets are:
[echo]   - jar.get      : download of all the libs needed to build the package
[echo]   - clean       : delete all binary distribution files
[echo]   - compile     : compile all files
[echo]   - jar.srv     : creation of the service distribution
[echo]   - jar.tag     : creation of the JCR tag distribution
[echo]   - war        : creation of the web distribution
[echo]   - assemble.ear : assembles the ear package
[echo]   - deploy.server : deploys the ear file to the server
```

Possible targets of the CmsImport package

In the file **configuration.properties** located in the **<project>/src/resources** directory the JBoss Portal Server has to be adjusted as well.

Before able to build the package all mandatory libraries have to be downloaded by using **ant jar.get** .

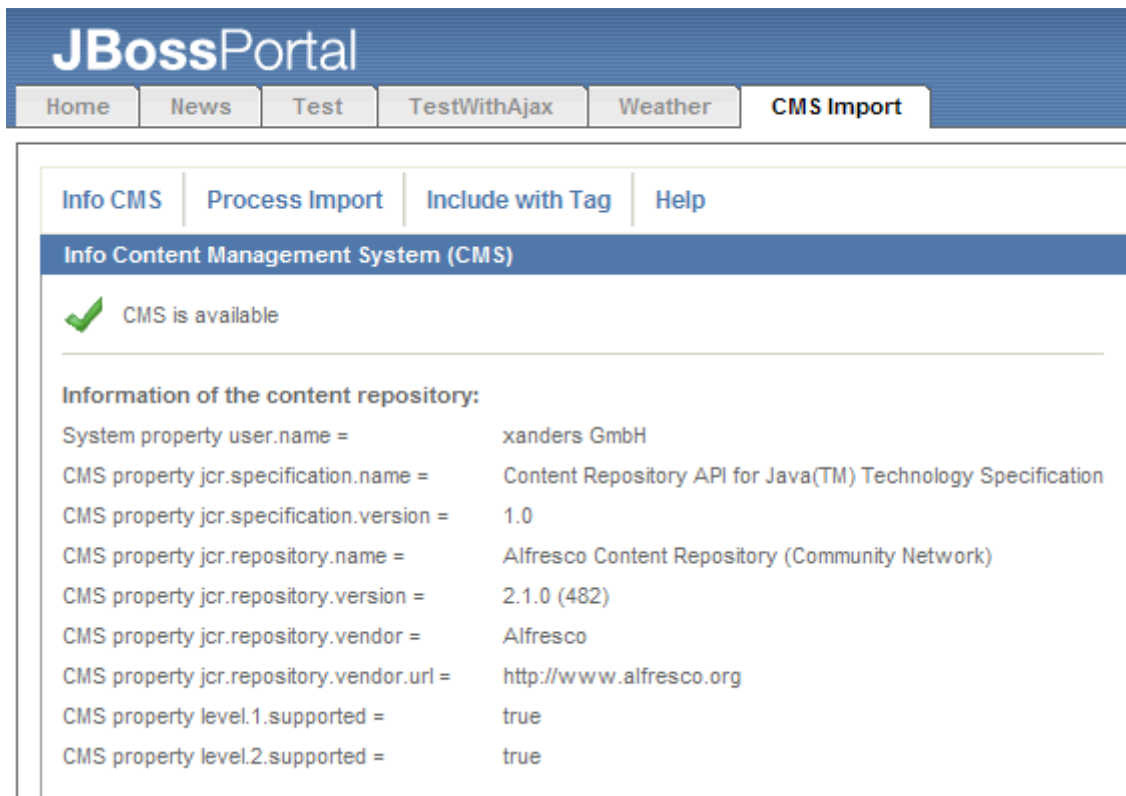
The package can be build with **ant assemble.ear**, the resulting ear file named **cms_import.ear** will be stored in the **<project>/build/dist** directory. To enable the import functionality at JBoss Portal Server, this file has to be unpacked in the JBoss deploy directory. The file **cms_import_web.war** as part of the ear file has to be unpacked as well in the **cms_import.ear** directory, so finally there is the following directory structure:



Directory structure of the cms_import.ear deployed on the JBoss server

After the deployment the package has to be embedded in the portal of JBoss in the usual way.

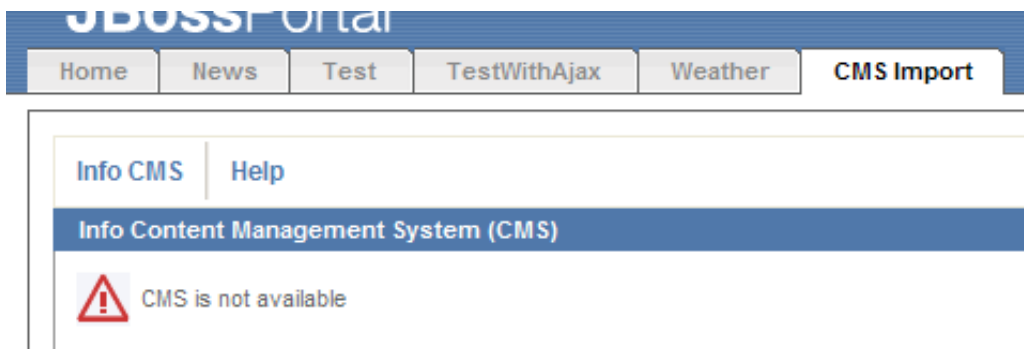
4. Screenshots of the portlet running on JBoss Portal Server



The screenshot shows the JBoss Portal interface. At the top, there is a navigation bar with the following tabs: Home, News, Test, TestWithAjax, Weather, and CMS Import. Below this, there is a sub-navigation bar with tabs: Info CMS, Process Import, Include with Tag, and Help. The main content area is titled "Info Content Management System (CMS)". A green checkmark icon is displayed next to the text "CMS is available". Below this, there is a section titled "Information of the content repository:" followed by a list of system properties and their values:

System property user.name =	xanders GmbH
CMS property jcr.specification.name =	Content Repository API for Java(TM) Technology Specification
CMS property jcr.specification.version =	1.0
CMS property jcr.repository.name =	Alfresco Content Repository (Community Network)
CMS property jcr.repository.version =	2.1.0 (482)
CMS property jcr.repository.vendor =	Alfresco
CMS property jcr.repository.vendor.url =	http://www.alfresco.org
CMS property level.1.supported =	true
CMS property level.2.supported =	true

The connectivity to the CMS (Alfresco) could be established, all panels (Info, Import, Demo) are accessible.



The screenshot shows the JBoss Portal interface. At the top, there is a navigation bar with the following tabs: Home, News, Test, TestWithAjax, Weather, and CMS Import. Below this, there is a sub-navigation bar with tabs: Info CMS and Help. The main content area is titled "Info Content Management System (CMS)". A red warning triangle icon is displayed next to the text "CMS is not available".

The connectivity to the CMS (Alfresco) could not be established (Alfresco is not running).

JBossPortal

Home | News | Test | TestWithAjax | Weather | **CMS Import**

Info CMS | Process Import | Include with Tag | Help

Import content of Content Management System (CMS)

Properties used for the import:

Import content to temp path =	D:/Programme/jboss-portal-2.6.3.GA/cmsimport-temp
Server deploy directory =	D:/Programme/jboss-portal-2.6.3.GA/server/default/deploy/cms_import.ear
Import node =	app:company_home/cm:cms_import
Import sub nodes =	cm:cms_import_web.war

Click button to start the import of the pages out of the Content Management System (CMS):

Start Import

CMS Import panel showing property information. By clicking the Start Import button the sub nodes will be imported into JBoss Server.

JBossPortal

Home | News | Test | TestWithAjax | Weather | **CMS Import**

Info CMS | Process Import | Include with Tag | Help

Include content with a JSF tag

Static content of a Content Management System (CMS) can be included with a JSF tag:

Hello JBoss Portal, I'm a text stored in Alfresco and included with a JSF tag.

CMS content (html snippet) is included with a JSF tag.

5. Licence

The use of the CmsImport package is restricted by the following license. The user of the software agrees to this license when using the software.

THIS SOFTWARE IS PROVIDED AS IT IS AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE THE CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE SOFTWARE PACKAGE CONSISTS OF MATERIALS WHICH IS COPYRIGHTED AND PROTECTED BY WORDWIDE COPYRIGHT LAW.