



# BEA WebLogic Portal<sup>®</sup>

## **Guide for Creating Additional BEA Repositories**

Version 8.1  
Revised: November 2003



# Copyright

Copyright © 2003 BEA Systems, Inc. All Rights Reserved.

## Restricted Rights Legend

This software and documentation is subject to and made available only pursuant to the terms of the BEA Systems License Agreement and may be used or copied only in accordance with the terms of that agreement. It is against the law to copy the software except as specifically allowed in the agreement. This document may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form without prior consent, in writing, from BEA Systems, Inc.

Use, duplication or disclosure by the U.S. Government is subject to restrictions set forth in the BEA Systems License Agreement and in subparagraph (c)(1) of the Commercial Computer Software-Restricted Rights Clause at FAR 52.227-19; subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, subparagraph (d) of the Commercial Computer Software--Licensing clause at NASA FAR supplement 16-52.227-86; or their equivalent.

Information in this document is subject to change without notice and does not represent a commitment on the part of BEA Systems. THE SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. FURTHER, BEA Systems DOES NOT WARRANT, GUARANTEE, OR MAKE ANY REPRESENTATIONS REGARDING THE USE, OR THE RESULTS OF THE USE, OF THE SOFTWARE OR WRITTEN MATERIAL IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE.

## Trademarks or Service Marks

BEA, Jolt, Tuxedo, and WebLogic are registered trademarks of BEA Systems, Inc. BEA Builder, BEA Campaign Manager for WebLogic, BEA eLink, BEA Liquid Data for WebLogic, BEA Manager, BEA WebLogic Commerce Server, BEA WebLogic Enterprise, BEA WebLogic Enterprise Platform, BEA WebLogic Express, BEA WebLogic Integration, BEA WebLogic Personalization Server, BEA WebLogic Platform, BEA WebLogic Portal, BEA WebLogic Server, BEA WebLogic Workshop and How Business Becomes E-Business are trademarks of BEA Systems, Inc.

All other trademarks are the property of their respective companies.



# Contents

## Creating Additional BEA Repositories

Step 1: Create Database Objects for the New Repository . . . . .	2
Step 2: Connect the New Repository to the Server . . . . .	3
Step 3: Associate your Portal Application with the New Repository . . . . .	4
Step 4: Start the Server. . . . .	7
Step 5: Create the New BEA Repository . . . . .	7



# Creating Additional BEA Repositories

Your content management system can be made up of multiple BEA Systems repositories and/or multiple third-party repositories. You can create multiple content repositories to meet your unique business needs. For example, if you need a physical separation of your content data then you can create multiple BEA Repositories.

**Note:** For large projects with several thousand content items, you may need to use separate database instances and minimize changes done in the production environment.

This process has five major steps:

- [Step 1: Create Database Objects for the New Repository](#)
- [Step 2: Connect the New Repository to the Server](#)
- [Step 3: Associate your Portal Application with the New Repository](#)
- [Step 4: Start the Server](#)
- [Step 5: Create the New BEA Repository](#)

## Step 1: Create Database Objects for the New Repository

In this step you will create database objects for your new content management additional database.

1. For Oracle, SQL Server, Sybase, or DB2 databases, create a database user for your additional content management database.
  - For **Oracle**, grant that user connectivity and resource privileges, and assign a default tablespace for the additional content management database objects.
  - For **SQL Server** and **Sybase**, create a database for the additional content management database objects and make that user the database owner (dbo) user for that database.
  - For **DB2**, grant that user `createTAB BINDADD connect` privileges to create the additional content management database objects.

**Note:** PointBase is not recommended for a production repository.

2. Set up your additional content management database in the `db_settings.properties` file.

- a. Open your domain's `db_settings.properties` file for edit.  
(`<BEA_HOME>/weblogic81/samples/domains/portal`)

- b. Uncomment the database settings for your new target database.

- c. Update the following settings for your additional content management database:

```
- server=<SERVER_NAME>
- dblogin=CONTENT_USER
- password=CONTENT_USER_PASSWORD
```

3. Within the `db_settings.properties` file, find the `p13n_modules`, `portal_modules` and `netuix_modules` lines at the top of the file.

- a. Copy these 3 lines and comment out the original settings by adding a `#` sign.

- b. Replace the original settings as follows:

```
- p13n_modules=p13n au bt ds with
p13n_modules=
- portal_modules=cm wlcs wps collaboration sample_cm with
portal_modules=cm sample_cm.
```



```
- netuix_modules=pf with  
  netuix_modules=cm sample_cm
```

When you are finished, the section should look like this:

```
#p13n_modules=p13n au bt ds  
#portal_modules=cm wlcs wps collaboration sample_cm  
#netuix_modules=pf  
p13n_modules=  
portal_modules=cm sample_cm  
netuix_modules=
```

- c. Save the changes to the `db_settings.properties` file.
4. Create the additional content management database objects by running the `create_db.cmd` or `create_db.sh` script.
  - a. Navigate to the `<BEA_HOME>\user_projects\domains\<YOUR_DOMAIN>` directory, and double-click on the `create_db.cmd` file.  
  
OR
  - b. If you are using the sample domain, run the `create_db.cmd` file from the following directory: `<BEA_HOME>\weblogic81\samples\domains\portal`.

## Step 2: Connect the New Repository to the Server

1. Start WebLogic Server for your domain, and login to the console.
2. Configure a new connection pool for your additional content management database.
  - a. Go to Services -> JDBC -> Connection Pools.
  - b. Right-click the existing Connection Pool, `cgPool` and select Clone `cgPool`.
  - c. Choose a name for the new Connection Pool (For example: `contentPool2`).
  - d. In the General tab, edit the settings for the cloned pool to match those of the new database you will be using for the additional repository.
  - e. Click Clone, and apply the changes.
3. Update your data sources.
  - a. From Services -> JDBC -> Data Sources, right-click the `cgPool` data source.
  - b. Select “`cgPool Data Source`” from the drop-down menu.

- c. Change the name of the data source from “Clone of cgPool Data Source” to something meaningful (For example: newContentDataSource).
- d. Set the JNDI Name to match the new Data Source name.
- e. Using the Pool Name drop down menu, set the pool to the new connection pool name (For example, contentPool2).
- f. Click Clone, and apply the changes.

### Step 3: Associate your Portal Application with the New Repository

The BEA Repository uses Entity Beans (EJBs) to associate a portal application with its data source(s). Because BEA’s default configuration is configured for just one data source (or repository), you’ll need to modify the XML files that maintain the EJB definitions for your data source(s). In short, you need to create new EJBs in these files that can be utilized by the additional repository you have created.

The easiest way to do this is to use the existing XML files as templates for creating the new EJB definitions and then add the new EJB definitions to the original file.

Redeploy Content EJBs to New Data Source (`ejb-jar.xml`, `weblogic-ejb-jar.xml`, and the `weblogic-cmp-rdbms-jar.xml`).

1. Create the workspace in which you will do your editing.
  - a. Create two temporary, working directories to use when completing your edits. For example, `c:\temp\working` and `c:\temp\newfiles`
  - b. Locate the `content_repo.jar` in the portal application directory that you want to associate with the new repository. For example, `<BEA home>/weblogic81/samples/portal/portalApp/content_repo.jar`
  - c. Place a copy of the `content_repo.jar` file in each of the working directories you have created.
  - d. Using a compression utility such as WinZip or JavaJar, unzip each `content_repo.jar` in its respective directory. At this point, the `content_repo.jar` files and their contents are identical.
  - e. You will now work with three files:
    - `ejb-jar.xml`

### Step 3: Associate your Portal Application with the New Repository

- weblogic-ejb-jar.xml
- weblogic-ejb-rdbms-jar.xml

You will make edits to the files contained in your `C:\temp\working` directory and copy those changes to the files contained in your `C:\temp\newfiles` directory.

2. After creating the temp directories and files to work with, configure the `ejb-jar.xml` file.
  - a. Open the `C:\temp\ejb-jar.xml` file and open the `C:\temp\newfilesejb-jar.xml` file. You will have two files open.
  - b. In the `C:\temp\working\ejb-jar.xml` file, find and replace, EJB (EJB is case sensitive) with a new name. For example: `EJBNew`.

**Note:** You will need to use this new name throughout all of the files.
  - c. Copy the following from the `C:\temp\working\ejb-jar.xml` file to the `C:\temp\newfiles\ejb-jar.xml` file.

---

**Copy everything between . . . from your temporary file**

---

```
<enterprise-beans>  
</enterprise-beans>
```

---

```
<relationships> </relationships>
```

---

```
<assembly-descriptor>  
</assembly-descriptor>
```

---

**and paste before . . . in your original file**

```
</enterprise-beans>
```

```
</relationships>
```

```
</assembly-descriptor>
```

---

- d. Save the `C:\temp\newfiles\ejb-jar.xml`.
3. Using the temporary directories and files you created in [Step 1](#), configure the `META-INF\weblogic-ejb-jar.xml` file.
    - a. Open the `C:\temp\working\META-INF\weblogic-ejb-jar.xml` file and the `C:\temp\newfiles\META-INF\weblogic-ejb-jar.xml` file. You will have two files open.
    - b. In the `C:\temp\working\META-INF\weblogic-ejb-jar.xml` file, find and replace, EJB (EJB is case sensitive) with a new name. For example: `EJBNew`.

**Note:** Use the same name you used in Step 2b.

- c. In the C:\temp\working\META-INF\weblogic-ejb-jar.xml file , find and replace, (case sensitive) BEA\_content with a new name (for example, BEA\_content\_new.)
  - d. In the C:\temp\working\META-INF\weblogic-ejb-jar.xml file, find and replace <jndi-name>contentDataSource </jndi-name> (case sensitive), and replace with DataSource you defined in [Step 2: Connect the New Repository to the Server](#). (For example, <jndi-name>newContentDataSource</jndi-name>).
- Note:** Do not to do a global search and replace because it should be updated in <jndi-name>newContentDataSource</jndi-name>. <help!!!>
- e. Copy the following from the C:\temp\working\META-INF\weblogic-ejb-jar.xml file to the C:\temp\newfiles\META-INF\weblogic-ejb-jar.xml:

<b>Copy everything between . . . from your</b>	<b>and paste before . . . in your</b>
C:\temp\working\META-INF\weblogic-ejb-jar.xml <b>file</b>	C:\temp\newfiles\META-INF\weblogic-ejb-jar.xml <b>file</b>
<code>&lt;weblogic-ejb-jar&gt; &lt;/weblogic-ejb-jar&gt;</code>	<code>&lt;/weblogic-ejb-jar&gt;</code>

- f. Save the C:\temp\newfiles\META-INF\weblogic-ejb-jar.xml file.
4. Using the temporary directories and files you created in [Step 1.](#), configure the META-INF/weblogic-cmp-rdbms-jar.xml file.
- a. Open the C:\temp\working\weblogic-cmp-rdbms-jar.xml file and the C:\temp\newfiles\META-INF\weblogic-cmp-rdbms-jar.xml file. You will have two files open.
  - b. In the C:\temp\working\weblogic-cmp-rdbms-jar.xml file, find and replace EJB (EJB is case sensitive) with same name that you used in the //Meta inf/ejb-jar. For example: EJBNew.
- Note:** Use the same name you used in <x-ref>.
- c. In the C:\temp\working\weblogic-cmp-rdbms-jar.xml temp file, find and replace, contentDataSource (case sensitive) with the DataSource you defined in [step 1. in ""](#). (For example: newContentDataSource)
  - d. Copy the following from C:\temp\working\weblogic-cmp-rdbms-jar.xml file to the C:\temp\newfiles\META-INF\weblogic-cmp-rdbms-jar.xml file:

---

<b>Copy everything starting at the first . . . and ending at the last . . . from the</b> C:\temp\working\META-INF\webl ogic-cmp-rdbms-jar.xml <b>file</b>	<b>and paste after the last . . . in the</b> C:\temp\newfiles\META-INF\weblo gic-cmp-rdbms-jar.xml <b>file</b>
---	--

---

```
<weblogic-rdbms-bean>          </weblogic-rdbms-bean>
</weblogic-rdbms-bean>
```

---

```
<weblogic-rdbms-relation>     </weblogic-rdbms-relation>
</weblogic-rdbms-relation>
```

---

5. In the C:\temp\newfiles directory, re-zip or re-compress the content\_repo.jar file. Overwrite the existing content\_repo.jar file with the new file of the same name that contains your edited XML files. Be sure to keep the files in their original directory structure, including the META-INF directory.
6. Copy the new content\_repo.jar file you just created back into the original install directory. For example <BEA home>/weblogic81/samples/portal/portalApp/

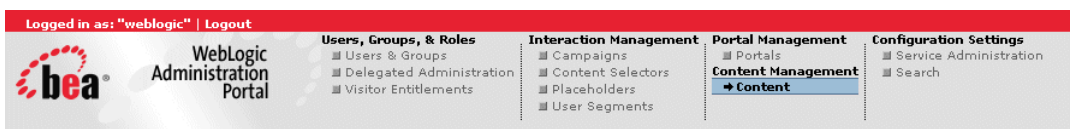
## Step 4: Start the Server

Restart the WebLogic Server to redeploy content\_repo.jar.

## Step 5: Create the New BEA Repository

To add a new repository:

1. Open the Administration Portal. (Start -> Programs -> BEA WebLogic Platform 8.1 -> Examples -> WebLogic Portal -> WebLogic Administration Portal)
2. Select “Content” in the menu at the top of the screen.



3. In the Browse and Edit Content Resource tree to the left, select Repository from the View drop-down menu.

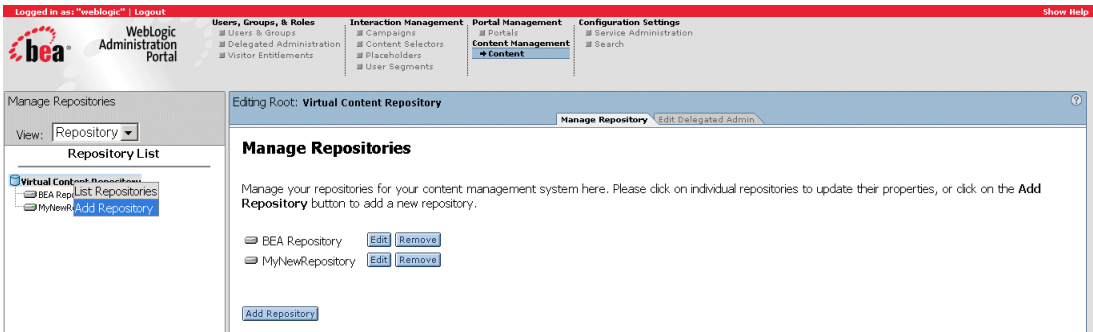
## Creating Additional BEA Repositories



- In the Manage Repositories tree to the left, right click on the Virtual Content Repository.



- Select Add Repository from the pop-up menu (or click the Add Repository button in editor pane).



- In the Editor pane, provide the following information:

Field	Description
Repository Name	The name you give your new repository. For example: MyNewRepository
Connection Class	(com.bea.content.spi.internal.RepositoryImpl)

**Note:** When you create a new repository, if you cut and paste the properties from a text pad document, be sure to delete any trailing blanks.

- Click Add Property, and provide the following parameters:

## Step 5: Create the New BEA Repository

Manage Repositories

View: Repository ▼

Repository List

- Virtual Content Repository
  - BEA Repository
  - MyNewRepository

Editing Root: **Virtual Content Repository**

[Manage Repository](#) [Edit Delegated Admin](#)

### Add Repository

Add a new repository to the content management system. Please fill in all the required fields, which are marked with "\*".

Repository Name: \*

Connection Class: \*

Username:

Password:

Properties	
Key	Value
<input type="text" value="key1"/>	<input type="text" value="value1"/>

[Add Property](#)

Repository Cache Settings			
Node Cache		Binary Cache	
Maximum Entries:	<input type="text" value="50"/>	Maximum Entries:	<input type="text" value="10"/>
Time To Live (seconds):	<input type="text" value="6000"/>	Time To Live (seconds):	<input type="text" value="6000"/>
		Cache Size/Item (bytes):	<input type="text" value="1024"/>
Is Enabled:	<input checked="" type="checkbox"/>	Is Enabled:	<input checked="" type="checkbox"/>

[Create](#)

Key	Value
NODE_OPS_HOME	$\${APPNAME}$ .BEA_content_<the name you chose, in our example, it was new> .RepoNodeOpsHome  For example: $\${APPNAME}$ .BEA_content_RepoNodeOpHome
OBJECT_CLASS_OPS_H OME	$\${APPNAME}$ .BEA_content_<the name you chose, in our example, it was new> .RepoObjectClassOpsHome
SEARCH_OPS_HOME	$\${APPNAME}$ .BEA_content_<the name you chose, in our example, it was new> .RepoSearchOpsHome

8. Click Create.

## Creating Additional BEA Repositories