49

Extending WebCenter Spaces Using JDeveloper

Note: This PDF contains **important updates** to the current version of Chapter 49 Developer's Guide for WebCenter on Oracle Technology Network (OTN). The OTN version will be updated with the same information soon but in the meantime please follow instructions in this PDF.

WebCenter Spaces can be customized using browser-based tools and also using JDeveloper. This chapter describes how to customize WebCenter Spaces resources in JDeveloper and subsequently utilize them in WebCenter Spaces. For example, you may want to deploy additional shared libraries that include custom code or some additional task flows. Or, you might want to edit WebCenter Spaces resources, such as skins, page templates, and navigations, in JDeveloper.

It includes the following sections:

Audience

This section is intended for developers who have basic knowledge of JDeveloper with WebCenter Portal Framework Extensions and want to perform advanced customizations for their WebCenter Spaces deployment.

49.1 Downloading a Workspace for WebCenter Spaces Development

A JDeveloper workspace for developing WebCenter Spaces extensions is available as a download from Oracle Technology Network - this workspace is named **DesignWebCenterSpaces.jws**.

To download the WebCenter Spaces development workspace:

1. Ensure WebCenter Spaces is installed, configured, and working as expected.
2. Download and install Oracle JDeveloper 11g (11.1.1.4.0).

   Oracle JDeveloper 11g (11.1.1.4.0) is available for download from: `http://www.oracle.com/technetwork/developer-tools/jdev/downloads/index.html`

3. Download and install Oracle JDeveloper WebCenter Framework and Services Design Time Extensions (oracle.webcenter.framework_bundle.zip).

   See also, Section 3.2, "Installing the WebCenter Extension Bundle" in the *WebCenter Developer's Guide* (available on OTN at http://download.oracle.com/docs/cd/E17904_01/webcenter.1111/e10148/jpsdg_wcsres.htm.

4. Download the WebCenter Spaces development ZIP file (DesignWebCenterSpaces.zip) from Oracle Technology Network:

   http://download.oracle.com/otndocs/tech/webcenter/files/DesignWebCenterSpaces.zip

5. Unzip the content locally (Table 49–1).

Table 49–1 DesignWebCenterSpaces.zip - Unzip Folders

Unzip Folders	Description
\DesignWebCenterSpaces	Includes DesignWebCenterSpaces.jws. This workspace enables you to configure your shared library list, that is, you can specify one or more shared libraries containing various customizations (such as custom java code, custom taskflows, and so on) that you want to utilize in WebCenter Spaces. The workspace also provides a design environment for editing and uploading WebCenter Spaces resources.
\copy_to_common	Includes a common directory that contains WebCenter WLST commands. WebCenter's WLST commands are required to upload customized resources directly to WebCenter Spaces from DesignWebCenterSpaces.jws.
\copy_to_jdev_ext	Includes oracle.webcenter.portal.jar. Required update to Oracle JDeveloper WebCenter Framework and Services Design Time Extensions.

6. Copy WebCenter's WLST (WebLogic Scripting Tool) commands to the Oracle home directory where JDeveloper is installed:

   a. Navigate to: <DesignWebCenterSpaces_Unzip_Location>\copy_to_common

   b. Make a copy of the common directory.

      The common directory is located under <DesignWebCenterSpaces_Unzip_Location>\copy_to_common.

   c. Navigate to: <JDeveloper_install_directory>\oracle_common

      Tip: The JDeveloper_install_directory is the base directory under which JDeveloper is also installed, that is, the same directory that you configure as the jdeveloper.install.home.directory in config.properties (see also Table 49–2).

   d. Copy the content of the common directory here.

> **Tip:** On Linux, use the following command: `cp -r <DesignWebCenterSpaces_Unzip_location>/copy_to_common/common/.`

7. Copy new version of `oracle.webcenter.portal.jar` to the Oracle JDeveloper extensions directory:

   a. Navigate to: `<DesignWebCenterSpaces_Unzip_Location>\copy_to_jdev_ext`

   b. Make a copy of `oracle.webcenter.portal.jar`.

      The `oracle.webcenter.portal.jar file` is located under `<DesignWebCenterSpaces_Unzip_Location>\copy_to_jdev_ext`.

   c. Navigate to: `<JDeveloper_install_directory>\jdeveloper\jdev\extensions\`

      > **Tip:** The `JDeveloper_install_directory` is the base directory under which JDeveloper is also installed, that is, the same directory that you configure as the `jdeveloper.install.home.directory` in `config.properties` (see also Table 49–2).

   d. Copy the `oracle.webcenter.portal.jar file` here.

49.2 Including Additional Shared Libraries in WebCenter Spaces

This section includes the following subsections:

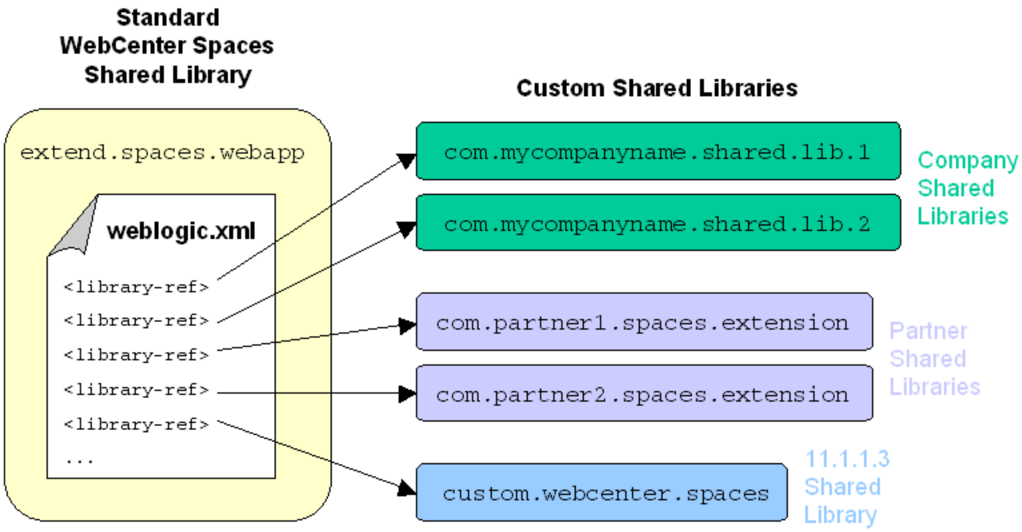
- Section 49.2.1, "What You Should Know About WebCenter Spaces Shared Library Deployment"
- Section 49.2.2, "Deploying Your Own Custom Code and Task Flows in Shared Libraries"
- Section 49.2.3, "Rebuilding the WebCenter Spaces Shared Library List"

49.2.1 What You Should Know About WebCenter Spaces Shared Library Deployment

WebCenter Spaces includes the standard shared library `extend.spaces.webapp.war`. This .WAR file includes a deployment descriptor (`weblogic.xml`) which can reference other libraries that contain WebCenter Spaces customizations.

For example, if you want to use custom code or task flows deployed in several shared libraries from multiple sources you can list them in the `extend.spaces.webapp` shared library, as illustrated in Figure 49–1.

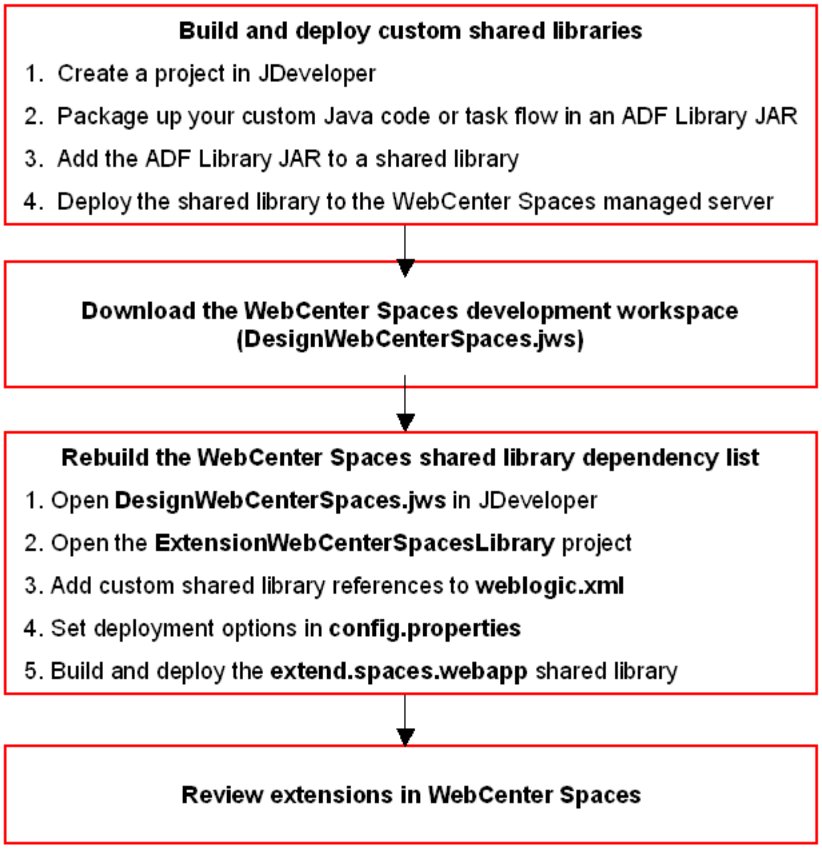
Figure 49–1 Referencing Custom Shared Libraries in extend.spaces.webapp.war



This development model provides an easy way to utilize additional shared libraries in WebCenter Spaces from multiple contributors, including developers, customers, partners. See also, Section 49.2.3, "Rebuilding the WebCenter Spaces Shared Library List".

Whenever you deploy a new shared library that includes WebCenter Spaces extensions you must modify the WebCenter Spaces shared library dependency list and redeploy `extend.spaces.webapp.war`. Figure 49–2 illustrates the process.

Figure 49–2 Overview - Including Additional Shared Libraries in WebCenter Spaces



Build and deploy custom shared libraries

1. Create a project in JDeveloper
2. Package up your custom Java code or task flow in an ADF Library JAR
3. Add the ADF Library JAR to a shared library
4. Deploy the shared library to the WebCenter Spaces managed server

Download the WebCenter Spaces development workspace (DesignWebCenterSpaces.jws)

Rebuild the WebCenter Spaces shared library dependency list

1. Open **DesignWebCenterSpaces.jws** in JDeveloper
2. Open the **ExtensionWebCenterSpacesLibrary** project
3. Add custom shared library references to **weblogic.xml**
4. Set deployment options in **config.properties**
5. Build and deploy the **extend.spaces.webapp** shared library

Review extensions in WebCenter Spaces

49.2.2 Deploying Your Own Custom Code and Task Flows in Shared Libraries

Developers can build extensions for WebCenter Spaces, package them in ADF Library JARs, and then deploy them as shared libraries for use in WebCenter Spaces. For example, you can:

- Develop custom code in JDeveloper and make it available in WebCenter Spaces.
- Develop custom task flows in JDeveloper and make them available to WebCenter Spaces

To help you get started, Oracle provides a sample workspace in which you can develop extensions, and build and deploy WebCenter Spaces shared libraries that contain your custom code.

The sample workspace, `SampleWebCenterSpacesExtension.jws`, contains sample extensions and scripts that enable you to deploy custom code for WebCenter Spaces. For more information, download the accompanying whitepaper, *Using WebCenter Spaces Extension Samples (11.1.1.4.0)*, from Oracle Technology Network at `http://www.oracle.com/technetwork/index.htm`.

49.2.3 Rebuilding the WebCenter Spaces Shared Library List

Whenever you deploy a new shared library that includes WebCenter Spaces extensions you must update the WebCenter Spaces shared library dependency list and redeploy `extend.spaces.webapp.war`.

This section describes:

- Editing the Shared Library List
- Setting Build and Deployment Options (config.properties)
- Building and Deploying the Shared Library List (extend.spaces.webapp.war)
- Troubleshooting Shared Library Deployment
- Reverting to a Previous Shared Library List

49.2.3.1 Editing the Shared Library List

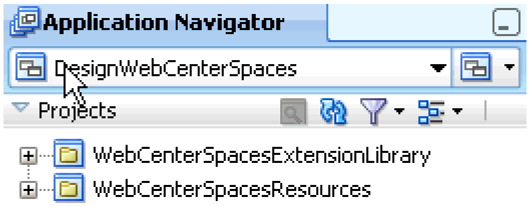
Use the *WebCenterSpacesExtensionLibrary* project to build and deploy `extend.spaces.webapp.war`—a shared library containing `weblogic.xml` in which you configure the WebCenter Spaces shared library dependency list.

1. Download and unzip the WebCenter Spaces development workspace.

   See Section 49.1, "Downloading a Workspace for WebCenter Spaces Development".

2. Navigate to the unzip directory, and open `DesignWebCenterSpaces\DesignWebCenterSpaces.jws` in JDeveloper (Figure 49–3).

Figure 49–3 Projects in the DesignWebCenterSpaces Workspace

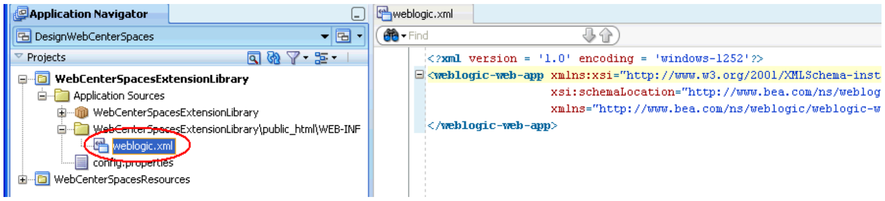


3. Expand the **WebCenterSpacesExtensionLibrary** project.

4. Navigate to **weblogic.xml**, under **Application Sources\WebCenterSpacesExtensionLibrary\public_html\WEB-INF** (Figure 49–4).

   Out-of-the-box, this file is blank, that is, no additional shared libraries are listed in the file.

Figure 49–4 WebCenterSpacesExtensionLibrary - Out-of-the-box weblogic.xml



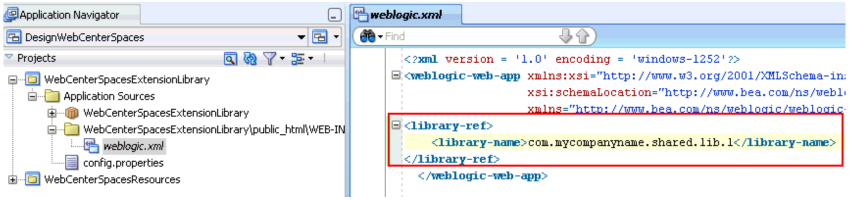
```
<?xml version = '1.0' encoding = 'windows-1252'?>
<weblogic-web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-inst
                  xsi:schemaLocation="http://www.bea.com/ns/weblog
                  xmlns="http://www.bea.com/ns/weblogic/weblogic-w
</weblogic-web-app>
```

5. Add a `<library-ref>` entry for each shared library that you want to use in WebCenter Spaces.

   For example, you can include a single library entry, as shown in Figure 49–5:

```
<library-ref>
  <library-name>com.mycompanyname.shared.lib.1</library-name>
</library-ref>
```

Figure 49–5 weblogic.xml - Single Shared Library Reference

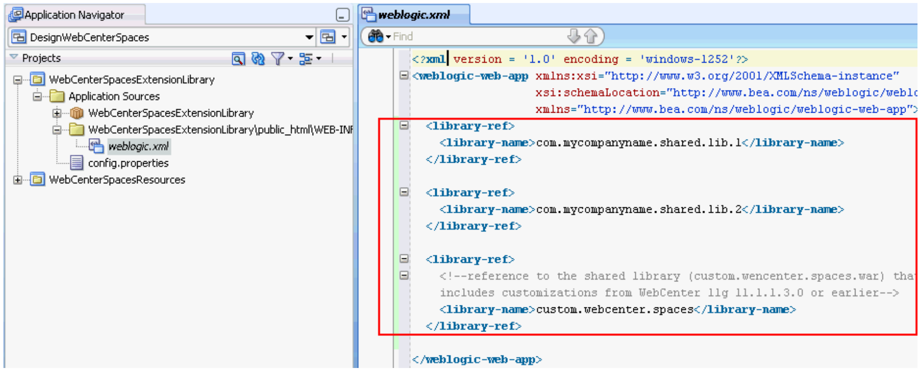


```
<?xml version = '1.0' encoding = 'windows-1252'?>
<weblogic-web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-in
                  xsi:schemaLocation="http://www.bea.com/ns/webl
                  xmlns="http://www.bea.com/ns/weblogic/weblogic
<library-ref>
    <library-name>com.mycompanyname.shared.lib.1</library-name>
</library-ref>
  </weblogic-web-app>
```

Or, reference multiple shared libraries, as shown in Figure 49–6:

```
<library-ref>
    <library-name>com.mycompanyname.shared.lib.1</library-name>
</library-ref>
<library-ref>
    <library-name>com.mycompanyname.shared.lib.2</library-name>
</library-ref>
<library-ref>
     <!--reference to the shared library (custom.wencenter.spaces.war) that
      includes customizations from WebCenter 11gR1 11.1.1.3.0 or earlier-->
    <library-name>custom.webcenter.spaces</library-name>
</library-ref>
```

Figure 49–6 weblogic.xml - Multiple Shared Library References



```
<?xml version = '1.0' encoding = 'windows-1252'?>
<weblogic-web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                  xsi:schemaLocation="http://www.bea.com/ns/weblogic/weblo
                  xmlns="http://www.bea.com/ns/weblogic/weblogic-web-app">
  <library-ref>
    <library-name>com.mycompanyname.shared.lib.1</library-name>
  </library-ref>

  <library-ref>
    <library-name>com.mycompanyname.shared.lib.2</library-name>
  </library-ref>

  <library-ref>
    <!--reference to the shared library (custom.wencenter.spaces.war) tha
    includes customizations from WebCenter 11g 11.1.1.3.0 or earlier-->
    <library-name>custom.webcenter.spaces</library-name>
  </library-ref>

</weblogic-web-app>
```

> **Note:** If you want to use custom extensions developed for a previous WebCenter Spaces release (11.1.1.3.0 or earlier), include the name of the old shared library, **`custom.webcenter.spaces`**, in `weblogic.xml`, as shown in Figure 49–6.

6. Ensure each shared library that you reference is deployed on the WebCenter Spaces managed server.

The next section describes how to set build and deployment options for the shared library list (`extend.spaces.webapp.war`).

49.2.3.2 Setting Build and Deployment Options (config.properties)

Before you can build the shared library list and deploy `extend.spaces.webapp.war` to the WebCenter Spaces managed server, you must provide some information about your environment and your WebCenter Spaces installation in the configuration file `config.properties`.

To set build and deployment properties for the `extend.spaces.webapp` shared library:

1. Open **DesignWebCenterSpaces.jws**.

See also, Section 49.1, "Downloading a Workspace for WebCenter Spaces Development"

2. Expand the **WebCenterSpacesExtensionLibrary** project.
3. Open **config.properties**.
4. Enter information about your JDeveloper environment and WebCenter Spaces installation, as shown in Table 49–2.

   The `config.properties` file describes each property and offers examples. The defaults provided are only samples and must be replaced with your own, installation-specific values.

Table 49–2 Configuring config.properties to Enable Deployment to the Managed Server

Configuration Property	Description
`jdeveloper.install.home.directory`	Base directory where JDeveloper is installed. The directory you specify contains other folders such as `jdeveloper`, `wlserver_10.3`, `modules`, and so on.
`wlst.executable`	Name of the WLST executable file. Either `wlst.cmd` (on Windows) or `wlst.sh` (on Linux).
`extending.spaces.home.dir`	Path to the WebCenter Spaces development workspace directory. For example, `C:\DesignWebCenterSpaces`
`oracle.jdeveloper.ojdeploy.path`	Path to `ojdeploy.exe` (Windows) or `ojdeploy` (Linux).
`wls.port`	Port number on which the WLS Administration Console is running
`wls.host`	Host machine on which WebCenter Spaces is running and where the custom shared library is to be deployed
`wls.userkey`	Name and location of the file storing administrator passwords. `config.properties` describes how to generate the file using WLST. The file can be copied to any suitable, accessible location.
`wls.userconfig`	Name and location of the file storing administrator user details. `config.properties` describes how to generate the file using WLST. The file can be copied to any suitable, accessible location.
`wls.target`	Targets on which the WebCenter Spaces shared library is to be deployed. For example, `WC_Spaces`.
`wls.webcenter.app.target`	Targets on which the `webcenter.ear` file is to be deployed. For example, `WC_Spaces`.
`webcenter.app.name`	Name of the WebCenter Spaces application. Always `webcenter`.
`restart.implementation.version.suffix`	Controls incremental implementation version numbers.

5. Save your updates to `config.properties`.

Updates to `config.properties` are available immediately so there is no need to restart JDeveloper.

The next section describes how to build and deploy the shared library list `extend.spaces.webapp.war`.

49.2.3.3 Building and Deploying the Shared Library List (extend.spaces.webapp.war)

You must rebuild and redeploy the `extend.spaces.webapp.war` shared library whenever you modify the shared library list (`weblogic.xml`).

To build and deploy the shared library list:

1. Open `DesignWebCenterSpaces.jws` in JDeveloper.

   See also, Section 49.1, "Downloading a Workspace for WebCenter Spaces Development".

2. Expand the **WebCenterSpacesExtensionLibrary** project.

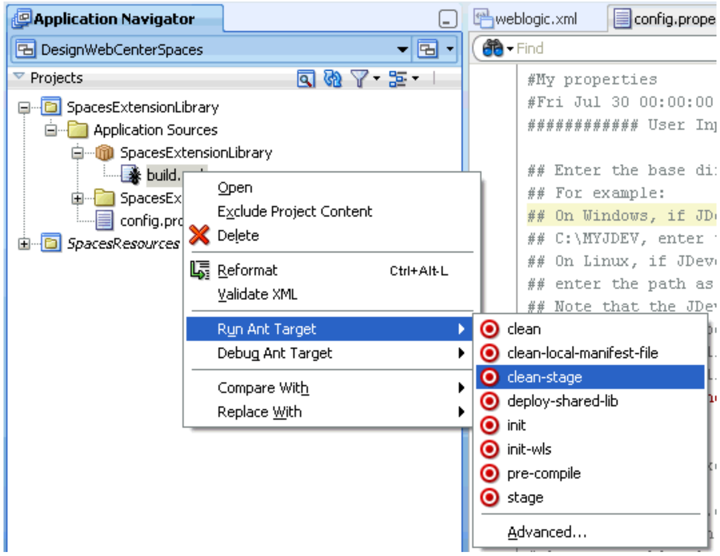
3. If you have not done so already, specify environment details in `config.properties`.

   See also, Section 49.2.3.2, "Setting Build and Deployment Options (config.properties)".

4. Right-click `build.xml`, located under the **Application Sources\WebCenterSpacesExtensionLibrary** (Figure 49–7).

5. Choose **Run Ant Target > clean-stage**.

Figure 49–7 Building a Customized WebCenter Spaces Extension Library



```
#My properties
#Fri Jul 30 00:00:00
############ User In
## Enter the base di
## For example:
## On Windows, if JD
## C:\MYJDEV, enter
## On Linux, if JDev
## enter the path as
## Note that the JDe
```

   This generates a new version of the shared library list `extend.spaces.webapp.war`. The implementation version number associated with the new .WAR is saved to `MANIFEST.MF` as follows:

   ```
   <UnzipDir>/DesignWebCenterSpaces/WebCenterSpacesExtensionLibr
   ary/META-INF/MANIFEST.MF
   ```

6. To deploy the WebCenter Spaces shared library list:

   a. Right-click `build.xml`, located under **Application Sources**.

   b. Choose **Run Ant Target > deploy-shared-lib**.

A new version of the shared library list is deployed to the WebCenter Spaces managed server. To verify the new deployment, login to the WLS Administration Console, navigate to the deployment overview page, and check the implementation version, that is, navigate to:

Deployments> extend.spaces.webapp>Overview

If the latest version is not active, refer to the Section 49.2.3.5, "Troubleshooting Shared Library Deployment".

49.2.3.4 Reverting to a Previous Shared Library List

If there is a problem with the latest shared library list or you want to revert to a previous version for some reason, you can undeploy (remove) the current version and revert to the previous version, using the WLS Administration Console.

You can remove unwanted shared library versions too. If you go through several "change-build-deploy-test" iterations, each incremental version is retained by default. As WebCenter Spaces only uses the latest shared library version you can clean up or delete previous versions if you want.

Before undeploying the latest version, you must shut down the managed server on which WebCenter Spaces is running. Once you have removed the latest version, you can restart the managed server.

Note: Oracle recommends that you do not delete the original `extend.spaces.webapp` shared library (version 11.1.1) as this enables you to revert to the out-the-box version if necessary.

1. Run the following script to stop the managed server on which WebCenter Spaces is deployed:

   - UNIX: DOMAIN_HOME/bin/stopManagedWeblogic.sh
   - Windows: DOMAIN_HOME\bin\stopManagedWeblogic.cmd

2. Login to the WLS Administration Console, navigate to the deployment overview page (**Deployments> extend.spaces.webapp>Overview**), and remove the latest shared library version.

   To revert to the out-of-the-box WebCenter Spaces deployment, revert to the original `extend.spaces.webapp.war` shared library, that is, delete all other WebCenter Spaces shared library versions, except for `extend.spaces.webapp.war` version 11.1.1.

3. Run the following script to start the managed server on which WebCenter Spaces is deployed:

   - UNIX: DOMAIN_HOME/bin/startManagedWeblogic.sh
   - Windows: DOMAIN_HOME\bin\startManagedWeblogic.cmd

49.2.3.5 Troubleshooting Shared Library Deployment

- **Changes are not available after deployment even though deployment successful**.

  WebCenter Spaces always uses the latest shared library version. Check that the implementation version in MANIFEST.MF matches the implementation version displayed in the WLS Administration Console.

  For example, check the value in `<UnzipDir>/DesignWebCenterSpaces/WebCenterSpacesExtensionLibrary/META-INF/MANIFEST.MF` is the same as that displayed in the WLS Admin Console under **Deployments> extend.spaces.webapp>Overview**

- **"Security: 090219" error displays:**

  ```
  weblogic.security.internal.encryption.EncryptionServiceException:
  weblogic.security.internal.encryption.EncryptionServiceException:
  weblogic.security.internal.encryption.EncryptionServiceException:
  [Security:090219]Error decrypting Secret Key com.rsa.jsafe.JSAFE_
  InputException: Invalid input length for decryption. Should be a multiple of
  the block size - 8
  ```

  Check that `wls.userconfig` and `wls.userkey` are both set correctly in the `config.properties` file, and verify that both security files specified are accessible

- **"DeployerException: Task 9 failed" displays:**

  ```
  weblogic.Deployer$DeployerException:
  weblogic.deploy.api.tools.deployer.DeployerException: Task 9 failed:
  [Deployer:149117]deploy library custom.webcenter.spaces
  [LibSpecVersion=11.1.1.2,LibImplVersion=11.1.1.2.5] on AdminServer,WC_Spaces.
  ```

  This error occurs if the implementation version of the new deployment and the existing deployment are the same. Use the WLS Administration Console to verify the current implementation version and then change the restart.implementation.version.suffix property in `config.properties` to this value. When you rebuild and redeploy the shared library, the implementation version should increment by '1'.

- **"java.lang.IllegalArgumentException" displays**:

  Restart the managed server on which WebCenter Spaces is deployed.

- **Deployment hangs:**

  If you encounter issues during the deployment phase (**Run Ant Target > deploy-shared-lib**) and the process hangs, end/kill the Ant process from inside JDeveloper, and then restart the Administration Server and the managed server for WebCenter Spaces. If you encounter further issues restarting either the Administration Server or the managed server, end/kill the associated Java processes and restart both servers.

- **Newly deployed WebCenter Spaces shared library is not active:**

  If a newly deployed shared library is not active, use WLS Administration Console to manually delete the non-active shared library version, and then rebuild (**Run Ant Target>clean-dist**) and redeploy (**Run Ant Target > deploy-shared-lib**) the WebCenter Spaces shared library.

- **"Error occurred attempting to display the list of pages you can access. Please contact the administrator" displays after logging in:**

  Restart the managed server on which WebCenter Spaces is deployed.

- **"WCS#2009.12.22.00.49.35: Fatal error occurred while getting members" displays:**

  Restart WebCenter Spaces.

49.3 Developing WebCenter Spaces Resources

This section describes the WebCenterSpacesResources project—-a design environment for WebCenter Spaces resource development. Through this project, you can upload resources, developed or customized through JDeveloper, to WebCenter Spaces.

This section includes the following subsections:

- Section 49.3.1, "Using the WebCenterSpacesResources Project for Round-Trip Development"
- Section 49.3.2, "Opening and Exploring the WebCenterSpacesResources Project"
- Section 49.3.3, "Enabling Direct Uploads to WebCenter Spaces"
- Section 49.3.4, "Editing WebCenter Spaces Resources in JDeveloper"
- Section 49.3.5, "Uploading Resources Direct to WebCenter Spaces"
- Section 49.3.6, "Exporting WebCenter Spaces Resources to an Archive"
- Section 49.3.7, "Troubleshooting Resource Round-Trip Development"

49.3.1 Using the WebCenterSpacesResources Project for Round-Trip Development

WebCenter's round-trip development features provide a simple, convenient way to modify WebCenter Spaces resources without redeploying the entire application. Round-trip development refers to features and techniques that allow you to download resources from a deployed application and upload them to JDeveloper for maintenance or enhancement. Once modified, you can upload the resource back to WebCenter Spaces for immediate use or for testing.

Out-of-the-box, WebCenter Spaces provides some default resources, such as skins and page templates, for people to use or modify. To keep WebCenter Spaces easy to use, the built-in functionality for creating and editing resources within WebCenter Spaces is purposely somewhat limited. If your resource requirements extend beyond the editing capabilities of WebCenter Spaces, you can further develop resources using a JDeveloper project that is especially designed for that purpose—*WebCenterSpacesResources*.

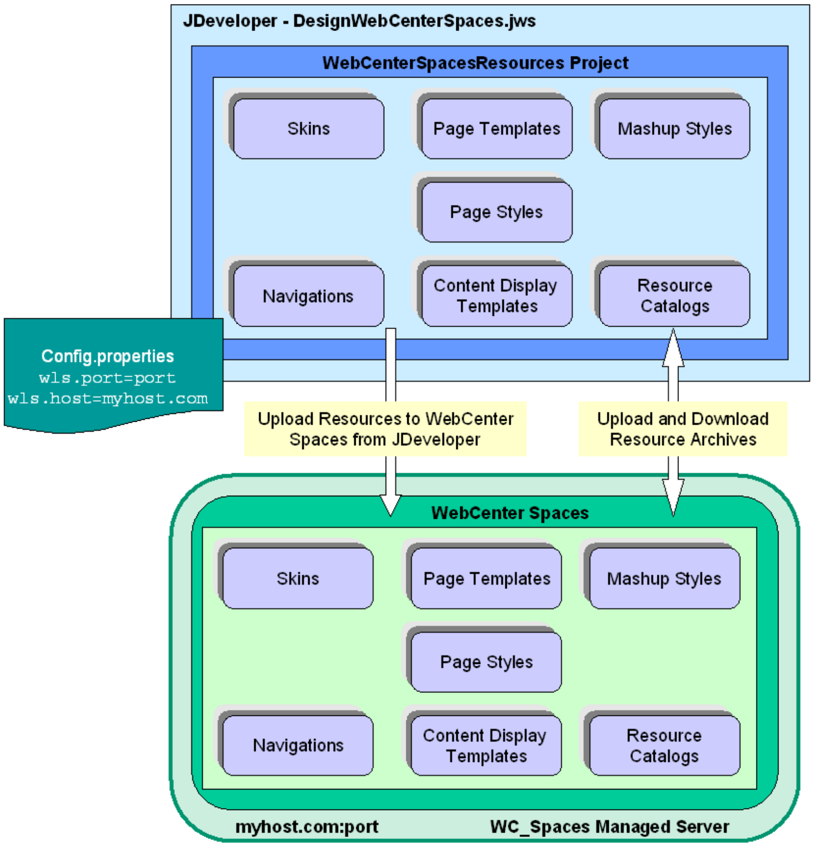
Oracle recommends that you use this project to build and manage custom skins, page templates, page styles, navigations, content display templates, mashup styles, and resource catalogs. The project provides some sample resources, as well as other files and folders you might need. You do not have to develop resources from scratch, you can export existing resources from WebCenter Spaces and edit them in JDeveloper, or use one of the samples as your starting point.

When you are ready to deploy new resources, you can upload them directly from JDeveloper to a live WebCenter Spaces application simply by providing JDeveloper with the name of the host and port where WebCenter Spaces is running.

Note: You do not need to deploy customized resources to a shared library.

Alternatively, you can login to WebCenter Spaces and upload new resources through the Resource Manager. If you decide to use this method you must save the resource to an export archive (.ear).

Figure 49–8 Designing Resources Using the WebCenterSpacesResources Project



49.3.2 Opening and Exploring the WebCenterSpacesResources Project

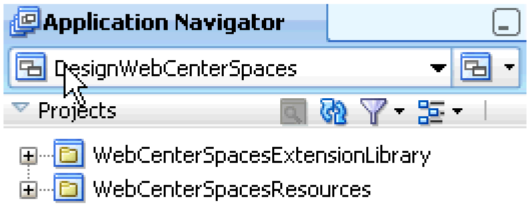
1. Download and unzip the WebCenter Spaces workspace.

   See Section 49.1, "Downloading a Workspace for WebCenter Spaces Development".

2. Navigate to the unzip directory, and open `DesignWebCenterSpaces\DesignWebCenterSpaces.jws` in JDeveloper (Figure 49–9).

   This workspace contains a project dedicated to resource development called **WebCenterSpacesResources**.

Figure 49–9 Projects in the DesignWebCenterSpaces Workspace



3. Expand the **WebCenterSpacesResources** project.

The **WebCenterSpacesResources** project contains several files and folders (Figure 49–10):

Figure 49–10 Exploring the WebCenterSpacesResources Project in JDeveloper

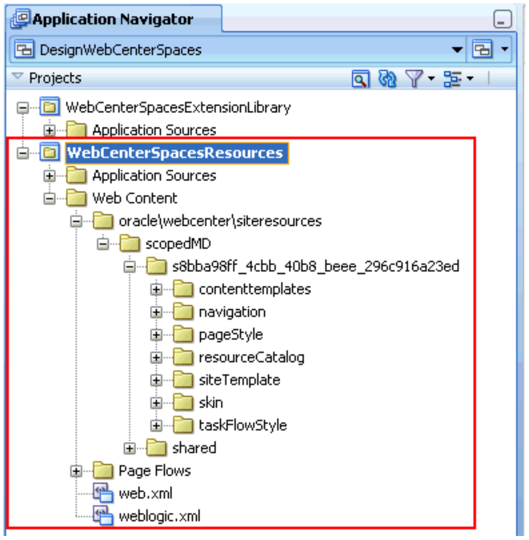


Table 49–3 describes the files and folders in the **WebCenterSpacesResources** project.

Table 49–3 Files and Folders in the WebCenterSpacesResources Project

Configuration Property	Description
\Application Sources	Supporting files and folders.
\Web Content	Contains WebCenter Spaces resources.
\oracle\webcenter\siteresources	Directory containing WebCenter Space resources. You can develop and manage all your application resources under this folder.
\scopedMD	Contains application-level and Space-level resources. ▪ Application-level resources are stored under the *"default scope"* directory `s8bba98ff_4cbb_40b8_beee_296c916a23ed` Out-of-the box, this directory contains several sample application resources. ▪ Space-level resources are stored under Space-specific directories with the same name as the Space's internal ID. No Space-level resources are provided out-of-the-box.
\<Default Scope ID>	Contains sample application-level resources.
\contenttemplates	Includes a sample Content Presenter display template (`content-default-list-item-view.jsff`) - Sample Content List Item View.
\navigation	includes a sample navigation (`Navigation.xml`) - SampleNavigation.
\pageStyle	Includes a blank page style (`TemplateHome.jspx`) - Sample Home Page Style.

Table 49–3 (Cont.) Files and Folders in the WebCenterSpacesResources Project

Configuration Property	Description
\resourceCatalog	Includes a sample catalog (`ResourceCatalog.xml`) - Sample Home Space Catalog
\siteTemplate	Includes a sample page templates (`Template.jspx`) - Sample Side Navigation **Note:** This directory is the correct directory for developing page templates (even though the directory name is siteTemplate).
\skin	Includes a sample skin (`Skin.css`) - Sample Spaces FX Skin.
\taskFlowStyle	Includes files for a sample mashup style (`pform.jsff` and `pform-definition.xml` - Sample Parameter Form.
\shared	Directory for content used by resources, for example, icons, images, and so on.
\Page Flows	Supporting files and folders.

49.3.3 Enabling Direct Uploads to WebCenter Spaces

If you build or modify skins, page templates, and so on, for a WebCenter Spaces application that is live or running in a test environment you can upload your updates directly to the application from JDeveloper. To enable direct updates from the JDeveloper **WebCenterSpacesResources** project you must:

- Specify the name of the host and port where WebCenter Spaces is running in `config.properties`.
- Check that you have the appropriate roles and permissions to upload the resource to WebCenter Spaces. You must have the following:
  - **WebLogic Server role** - `monitor` AND
  - **WebCenter Spaces resource permissions** - For example, to upload a new skin you must have the `Create, Edit, Delete Skins` permission.

Configuring the WebCenter Spaces hostname and port, enables the *Upload Portal Resource* option in your project. The first time you use this option, within a JDeveloper session, you must enter your WebCenter Spaces login name and password. For security reasons, your credentials are not saved for future sessions but they are stored in memory for the current JDeveloper session.

Once you are logged in to WebCenter Spaces you can upload resources, providing, of course, that you have the appropriate permissions to manage WebCenter Spaces resources.

> **Note:** Out-of-the-box, only the default Fusion Middleware administrator (`weblogic`) has all the required permissions.

To set connection properties in `config.properties` and grant permissions:

1. Open `config.properties`, available at:

   `<Unzip_Directory>\DesignWebCenterSpaces\config.properties`

2. Enter details relating to JDeveloper and your WebCenter Spaces installation, as shown in Table 49–4:

The `config.properties` file describes each property and offers examples. The defaults provided are only samples and must be replaced with your own, installation-specific values.

Table 49–4 config.properties Settings to Enable Direct Uploads

Configuration Property	Description
`jdeveloper.install.home.directory`	Base directory where JDeveloper is installed. In addition to `\jdeveloper`, the directory you specify must contain `\wlserver_10.3`, `\modules`, and so on.
`wls.host`	Name of the host machine where WebCenter Spaces is running.
`wls.port`	Port number on which the WLS Administration Console is running.

3. Save your updates to `config.properties`.
4. Verify and, if necessary, request permissions shown in Table 49–5.

Table 49–5 Permissions to Upload and Manage Resources Through JDeveloper

Type	Role or Permissions	Description
WebLogic Server	`monitor`	This role enables you to run the WLST scripts which upload resources from JDeveloper to WebCenter Spaces. See also, "Add users to roles" in *Oracle WebLogic Server Administration Console Help*.
WebCenter Spaces		
Application-level	`Create, Edit, Delete <resourcetype>`	This permission enables you to create and manage application-level resources for WebCenter Spaces.
	`Manage Configuration`	This permission gives you access to the WebCenter Spaces resource administration page. See also, "Managing Application Roles and Permissions" in *Oracle Fusion Middleware User's Guide for Oracle WebCenter Spaces*.
Space-level Standard	`Create, Edit, Delete Resources`	These permissions enable you to create and manage resources for a particular Space. Either standard or advanced permissions will apply, depending on the Space.
Space-level Advanced	`Create, Edit, Delete <resourcetype>`	
	`Manage Configuration`	This permission gives you access to the Space's resource administration page. See also, "Managing Roles and Permissions for a Space" in *Oracle Fusion Middleware User's Guide for Oracle WebCenter Spaces*.

5. Upload a sample resource, to test the connection, and verify that the resource is available in WebCenter Spaces:

   For details, see Section 49.3.5, "Uploading Resources Direct to WebCenter Spaces".

   See also, Section 49.3.7, "Troubleshooting Resource Round-Trip Development"

49.3.4 Editing WebCenter Spaces Resources in JDeveloper

Oracle recommends that you extend WebCenter Spaces resources in the JDeveloper project, *WebCenterSpacesResources*. This project provides everything you need to create, modify, and upload the following WebCenter Spaces resources:

- Page Templates
- Navigations
- Page Styles
- Skins
- Content Display Templates
- Resource Catalogs
- Mashup Styles

Where To Start...

You do not have to develop WebCenter Spaces resources from scratch. Most developers will download an existing resource from WebCenter Spaces and edit it in JDeveloper.

For more information, see:

- How to Download a WebCenter Spaces Resource (from WebCenter Spaces)
- How to Import a WebCenter Spaces Resource into JDeveloper

See, also How to Build WebCenter Spaces Resources.

Application-Level vs Space-Level Resources

The difference between application- and Space-level resources is one of scope. In WebCenter Spaces:

- **Application-level resources are available to all Spaces**.

  You must develop all application-level resources under the WebCenterSpacesResources project directory `oracle/webcenter/siteresources/scopedMD/s8bba98ff_4cbb_40b8_beee_296c916a23ed`.

  To upload application resources to WebCenter Spaces you must have WebCenter Spaces permissions: `Application-Manage Configuration` plus `Create, Edit, Delete <Resource_Type>`.

- **Space-level resources are only available within a particular Space**.

  You must develop a resource for a specific Space under the appropriate namespace, so that it can be uploaded to that Space:

  `oracle/webcenter/siteresources/scopedMD/<space_internal_id>`

  Similarly, resources downloaded from an existing Space and imported into JDeveloper, can only be uploaded back to the same Space. You cannot upload such resources to a different Space.

  To upload resources to a particular Space you must have the `Space-Manage Configuration` permission, plus one of:

  `Create, Edit, Delete Resources` (standard permission model)
  `Create, Edit, Delete <Resource_Type>` (advanced permission model)

If you want to upload resources directly from JDeveloper, you must also have the WebLogic Server `monitor` role. See Section 49.3.3, "Enabling Direct Uploads to WebCenter Spaces".

49.3.4.1 How to Import a WebCenter Spaces Resource into JDeveloper

When you import a WebCenter Spaces resource archive (.ear) into the *WebCenterSpacesResources* project the resource is uploaded to the appropriate project directory.

- **Application-level resources are imported to:**

  ```
  WebCenterSpacesResources\oracle\webcenter\siteresources\scopedMD\
  application_ID\resource_type\resource_ID
  ```

  For example, an application-level skin with the resource ID `gsr5a8c2fcc_bc7f_4cba_9254_36df58d66e60` is created under the application directory `s8bba98ff_4cbb_40b8_beee_296c916a23ed`:

  ```
  WebCenterSpacesResources\oracle\webcenter\siteresources\scopedMD\
  s8bba98ff_4cbb_40b8_beee_296c916a23ed\skin\gsr5a8c2fcc_bc7f_
  4cba_9254_36df58d66e60
  ```

- **Space-level resources are imported to:**

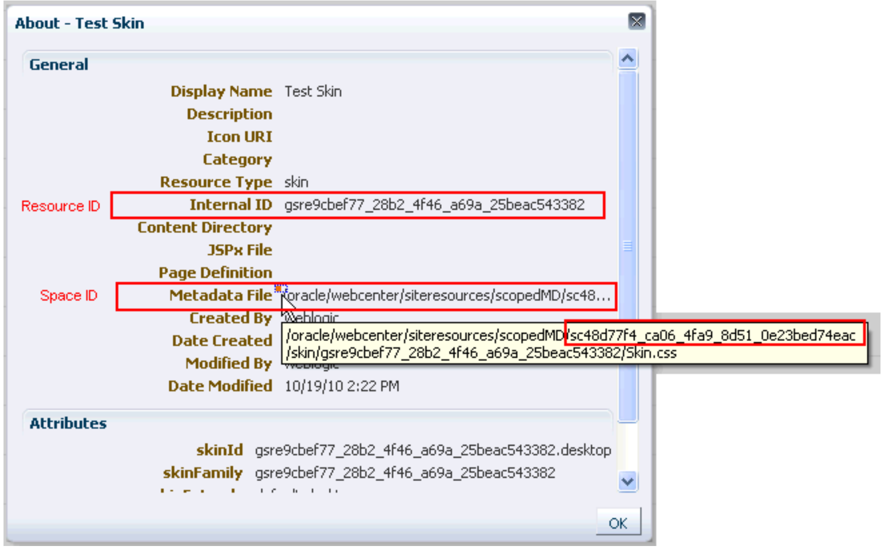
  ```
  WebCenterSpacesResources\oracle\webcenter\siteresources\scopedMD\
  Space_ID\resource_type\resource_ID
  ```

  For example, a Space-level skin with the resource ID `gsre9cbef77_28b2_4f46_a69a_25beac543382` is created under the Space directory `sc48d77f4_ca06_4fa9_8d51_0e23bed74eac`:

  ```
  WebCenterSpacesResources\oracle\webcenter\siteresources\scopedMD\
  sc48d77f4_ca06_4fa9_8d51_0e23bed74eac\skin\gsre9cbef77_28b2_
  4f46_a69a_25beac543382
  ```

Its important to know the internal IDs of WebCenter Spaces resources and the parent Space (Space-level resources only) so that you can locate the appropriate resource folders in JDeveloper. In WebCenter Spaces, internal IDs are published in the resource's About dialog (Figure 49–11). For details, see "Viewing Information About a Resource" in *Oracle Fusion Middleware User's Guide for Oracle WebCenter Spaces*.

Figure 49–11 Checking Resource IDs in WebCenter Spaces



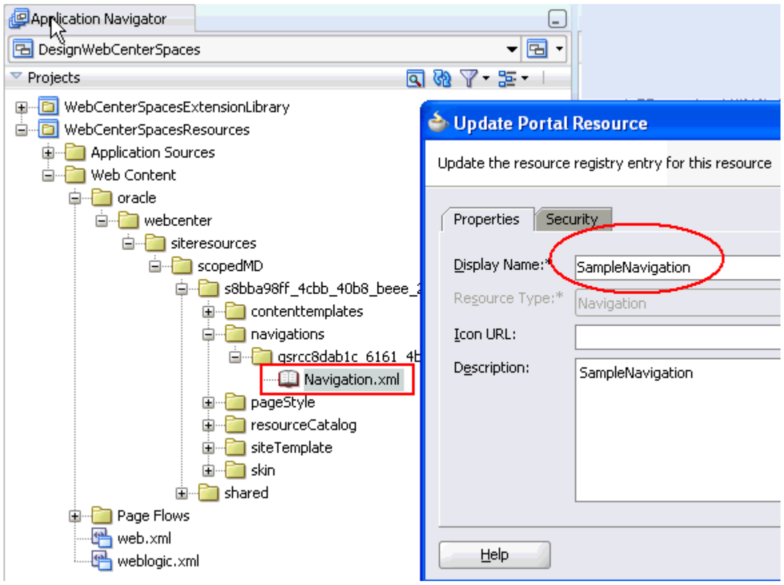
Display Name Test Skin
Description
Icon URI
Category
Resource Type skin
Internal ID gsre9cbef77_28b2_4f46_a69a_25beac543382
Content Directory
JSPx File
Page Definition
Metadata File /oracle/webcenter/siteresources/scopedMD/sc48...
Created By weblogic
Date Created
Modified By weblogic
Date Modified 10/19/10 2:22 PM

/oracle/webcenter/siteresources/scopedMD/sc48d77f4_ca06_4fa9_8d51_0e23bed74eac
/skin/gsre9cbef77_28b2_4f46_a69a_25beac543382/Skin.css

skinId gsre9cbef77_28b2_4f46_a69a_25beac543382.desktop
skinFamily gsre9cbef77_28b2_4f46_a69a_25beac543382

Internal IDs are not easy to remember like display names. If you want to identify an imported resource you can check its display name using the **Update Resource** option. Navigate to the resource directory, drill down to the resource file (for example, `mynavigation.xml`, `myskin.css`, `mypagetemplate.jpsx`, `mycatalog.xml`), and then click **Update Resource** from the right mouse menu (Figure 49–12).

Figure 49–12 Checking Resource Display Names in JDeveloper



Update Portal Resource

Update the resource registry entry for this resource

Display Name:* SampleNavigation
Resource Type:* Navigation
Icon URL:
Description: SampleNavigation

To import a WebCenter Spaces resource from an archive:

1. In the Application Navigator, right-click the **WebCenterSpacesResources** project, and choose **Import Portal Resource**.
2. In the Import Portal Resource dialog, enter the name or browse for the archive file (.EAR) that contains the resource that you want to import.

If the archive does not exist yet, login to WebCenter Spaces and download the resource to an .EAR file. See "Downloading a WebCenter Spaces Resource" in the *Oracle Fusion Middleware User's Guide for Oracle WebCenter Spaces*.

3. Click **OK**.

49.3.4.2 How to Build WebCenter Spaces Resources

To learn more about building resources, read the appropriate chapter in the *WebCenter Developer's Guide* (available on OTN at `http://download.oracle.com/docs/cd/E17904_01/webcenter.1111/e10148/jpsdg_wcsres.htm`:

- Page Templates - Working with Page Templates
- Page Styles - Working with Page Styles
- Skins - Creating and Managing Skins
- Navigations - Building a Navigation Model for Your Portal
- Content Display Templates - Creating Content Presenter Display Templates
- Resource Catalogs - Creating and Managing Resource Catalogs
- Mashup Styles - Working with Mashup Styles

49.3.5 Uploading Resources Direct to WebCenter Spaces

If WebCenter Spaces is up and running you can deploy new or updated resources directly to the application from your JDeveloper environment.

You must have appropriate resource management permissions in WebCenter Spaces to upload resources in this way. If you do not have the correct privileges, you are not allowed to upload resources. See also Section 49.3.3, "Enabling Direct Uploads to WebCenter Spaces".

1. If you have not done so already, configure WebCenter Spaces connections details in `config.properties` and request the appropriate resource management permissions.

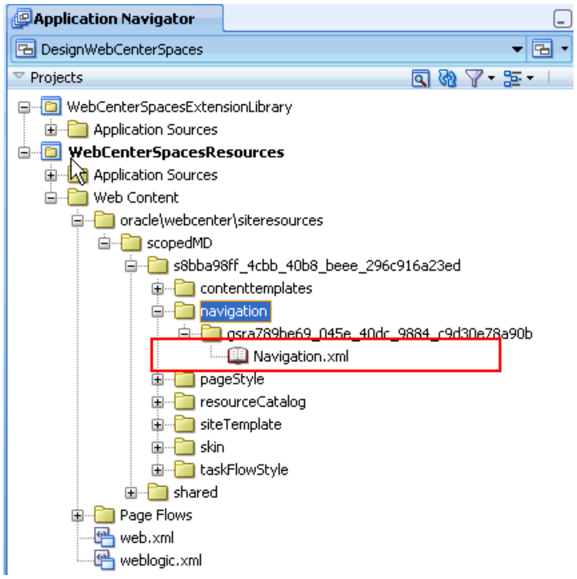
   See also, Section 49.3.3, "Enabling Direct Uploads to WebCenter Spaces".

2. In JDeveloper, open the **WebCenterSpacesResources** project, and navigate to the resource you want to upload.

   For example, to upload the sample application-level navigation *SampleNavigation*, navigate to (Figure 49–13):

   WebCenterSpacesResources\Web Content\oracle\webcenter\siteresources\scopedMD\s8bba98ff_4cbb_40b8_beee_296c916a23ed\navigation\gsrcc8dab1c_6161_4bb8_8764_127b4ecee01b

Figure 49–13 Navigating to WebCenter Spaces Resource Folders



3. Right-click the file associated with the resource you want to upload to WebCenter Spaces, and choose **Upload Portal Resource**.

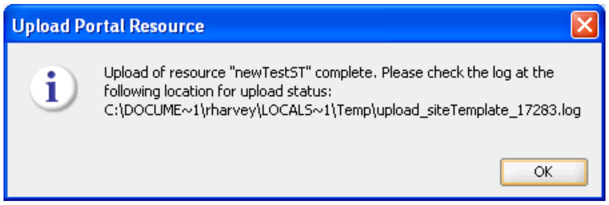
   There is a single file associated with most resource types. The exception is mashup style which has two files: `definition.xml` and `.jsff`. To upload a mashup style you must right-click the `definition.xml` file.

   If the menu option does not display, check that the WebCenter Spaces host name and port you entered in step 1 are correct.

4. If prompted, enter your WebCenter Spaces username and password and click **OK**.

   An information message displays to indicate that the upload process is complete and also provides the name and location of the upload log file—`upload_<resource_type_id>.log`.

Figure 49–14 Resource Upload Complete Message



Upload of resource "newTestST" complete. Please check the log at the following location for upload status:
C:\DOCUME~1\rharvey\LOCALS~1\Temp\upload_siteTemplate_17283.log

5. Review the upload log file.

   The following message indicates that the upload was successful.

   ```
   Imported <temp_log_directory>\<resource_type_id>.ear
   ```

   If you do not see this message, refer to Section 49.3.7, "Troubleshooting Resource Round-Trip Development"

6. Log in to WebCenter Spaces and verify that the resource is available through Resource Manager as expected.

   a. Log in to WebCenter Spaces.

b. Navigate to the Resource Manager as follows:

   To verify an application-level resource, navigate to the following URL:

   `http://host:port/webcenter/spaces/admin/resources`

   To verify a Space-level resources, navigate to the following URL:

   `http://host:port/webcenter/spaces/spaceName/admin/resources`

c. Select the appropriate resource type from the panel on the left, and verify that the resource you just uploaded is available.

49.3.6 Exporting WebCenter Spaces Resources to an Archive

When your resource is ready for use in WebCenter Spaces you can upload it directly to WebCenter Space or you can export the resource to an .EAR file format which you (or someone else) can upload at any time using WebCenter Spaces Resource Manager.

To export a WebCenter Spaces resource:

1. In the Application Navigator, navigate to the resource directory, drill down to the resource file (for example, `myskin.css`, `mypagetemplate.jpsx`, `mycatalog.xml`), and then click **Update Resourc**e from the right mouse menu.
2. In the Export Portal Resource dialog, enter or browse for the path and file name for the export file. For example: `C:\myskin.ear`
3. Click **OK**.

The steps to upload resource archives to WebCenter Spaces are available in "Uploading a WebCenter Spaces Resource" in the *Oracle Fusion Middleware User's Guide for Oracle WebCenter Spaces*.

49.3.7 Troubleshooting Resource Round-Trip Development

This troubleshooting section includes the following subsections:

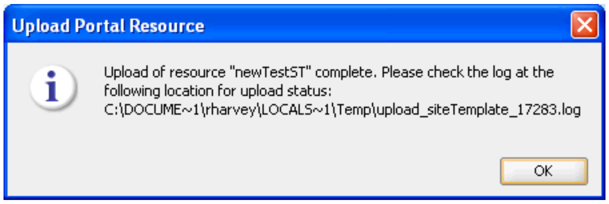
- Examining the Resource Upload Log File
- Unable to Upload Resources to WebCenter Spaces - Access denied
- Unable to Upload a Resource to WebCenter Space - java.lang.Exception

Examining the Resource Upload Log File

When you upload a resource to WebCenter Spaces progress details are recorded in a log file. The name of the log file includes the type of resource you are uploading and a unique ID, in the format—`upload_`*`resource_type_id`*`.log`. For example, `upload_siteTemplate_17283.log`.

The location of the log file depends on your local `temp` directory settings. The exact log file location displays in an information message such as this:

Figure 49–15 Resource Upload Log File Location



Upload of resource "newTestST" complete. Please check the log at the following location for upload status:
C:\DOCUME~1\rharvey\LOCALS~1\Temp\upload_siteTemplate_17283.log

The following log entry indicates that the upload process was successful.

```
Imported <temp_log_directory>\<resource_type_id>.ear
```

Insufficient permissions result in "`Access denied`" or "`java.lang.Exception`" errors, as described below.

Unable to Upload Resources to WebCenter Spaces - Access denied

The following error displays in the *resource upload log file* if you do not have the `monitor` role:

```
Operation: getScopeName(java.lang.String)  Detail: Access denied. Required roles:
Admin, Operator, Monitor, executing subject: principals=[yourusername]
```

The `monitor` role is a Weblogic Server role. This role enables you to run the WLST scripts which upload resources from JDeveloper to WebCenter Spaces. Ask your WebCenter Spaces administrator to grant you this role through the WebLogic Server Administration Console. See also, "Add users to roles" in *Oracle WebLogic Server Administration Console Help*.

Unable to Upload a Resource to WebCenter Space - java.lang.Exception

The following error displays in the *resource upload log file* if you do not have permission to manage any resources in WebCenter Spaces or the particular type of resource you are trying to upload:

```
javax.management.MBeanException at
weblogic.rjvm.ResponseImpl.unmarshalReturn(ResponseImpl.java:234) Root Cause:
Caused by: java.lang.Exception at
oracle.webcenter.portalframework.genericsiteresources.model.lifecycle.SiteResource
sHelper.updateResource(SiteResourcesHelper.java)
```

Ask your WebCenter Spaces administrator to grant you appropriate permissions through WebCenter Spaces Administration. Or, if the resource you are trying to upload is for a particular Space, contact the Space moderator to request the required permission. See also, Table 49–5, " Permissions to Upload and Manage Resources Through JDeveloper"".