An Oracle White Paper
December 2011

Using Spaces Extension Samples (11.1.1.6.0)

Executive Overview

WebCenter Portal: Spaces is a pre-built, pre-integrated, and highly customizable WebCenter Portal application. As you begin using the Spaces application you may find there are some aspects that you want to modify or extend beyond the customization capabilities provided in Spaces. For example, you may want to add custom functionality, such as additional task flows, custom properties for task flows, and so on.

To support advanced requirements, Oracle provides a sample JDeveloper workspace and several sample projects to help illustrate the customization process. This document describes the workspaces, the samples, and how to build and deploy shared libraries that contain your custom code.

Introduction

SampleWebCenterSpacesExtensions.jws is a sample JDeveloper workspace in which you can develop extensions, and build and deploy shared libraries that contain custom code for your Spaces application.

Oracle recommends that you copy the sample workspace and use this as a starting point for building Spaces extensions of your own. Within your own workspace version, you can review and deploy the samples, delete the samples, import future samples, and most importantly, add extension projects of your own. The current sample set shows you how to:

- **Support a new language** – Include support for an additional language in the Spaces application.

- **Customize user profile attributes** – Customize the default user attributes derived from the user profile store, and add custom attributes of your own.

- **Search custom profile attributes** – Deploy a custom Oracle SES crawler that searches custom profile attributes.

- **Create rule-based resources** – Change the skin based on custom logic.

- **Deploy custom task flows** – Develop specialized task flows in JDeveloper and make them available to Spaces users.

- **Customize the property pane for task flows** – Add custom properties in the Component Properties dialog for task flows.

- **Add a Web Service connection** – Invoke an external Web Service from within Spaces.

- **Customize the Login task flow** – Configure the default Login task flow to authenticate users against Oracle Access Manager SSO.

Deploying and Undeploying Sample Spaces Extensions

This section describes general tasks that you will perform if you use the samples. It explains how to:

- o Download and explore the Spaces extension samples
- o Build and deploy a custom Spaces shared library (.WAR)
- o Revert to a previous custom Spaces shared library version

Overview

To extend or customize the Spaces application you must download a sample JDeveloper workspace, add your customizations, and then deploy a shared library that contains your custom code. You must also download the Spaces development workspace so you can register the name of your custom shared library in the *Spaces shared library dependency list* and redeploy the updated list (in `extend.spaces.webapp`). The following diagram illustrates the process.

Download the sample workspace for Spaces
(*SampleWebCenterSpacesExtensions_ps5.zip*)

Deploy your custom shared library

1. Open *SampleWebCenterSpacesExtensions.jws* in JDeveloper

2. Create projects in JDeveloper and add your custom code

3. Set deployment options in *config.properties*

4. Modify the deployment profile

5. Build and deploy the custom shared library

Update and redeploy the shared library dependency list

1. Download *DesignWebCenterSpaces_ps5.zip.*

2. Open *DesignWebCenterSpaces.jws* in JDeveloper

3. List your custom shared library in *weblogic.xml*

4. Set deployment options in *config.properties*

5. Build and deploy *extend.spaces.webapp* shared library

Review your extensions in Spaces

This whitepaper describes how to build and deploy custom Spaces shared libraries that contain custom code (highlighted in red boxes in the image). For instructions on how to update and redeploy the Spaces shared library dependency list, refer to "Extending the Spaces Application Using JDeveloper" in *Developer's Guide for Oracle WebCenter* Portal *(11.1.1.6.0)* available at:
http://download.oracle.com/docs/cd/E23943_01/webcenter.1111/e10148/jpsdg_wcsres.htm

Downloading, Copying, and Exploring the Spaces Sample Workspace

Sample Spaces extensions are available from Oracle Technology Network. This section describes how to download and access the extensions.

1. Before you start:

    a) Ensure that the Spaces application is installed, configured, and working as expected.

    b) Download and install **Oracle JDeveloper 11g (11.1.1.6.0)**, which is available from:

        http://www.oracle.com/technetwork/developer-tools/jdev/downloads/index.html

    c) Download and install **Oracle JDeveloper WebCenter Framework and Services Design Time Extensions 11.1.1.6.0**
        (`oracle.webcenter.framework_bundle.zip`).

        Tip: You can install extensions from JDeveloper's Help menu. Select **Help > Check for Updates**.

2. Download **SampleWebCenterSpacesExtensions_ps5.zip** from Oracle Technology Network:

    http://download.oracle.com/otndocs/tech/webcenter/files/SampleWebCenterSpacesExtensions_ps5.zip

    This download includes a JDeveloper workspace, several sample extension projects, and other resources you might need to support your extensions.

3. Unzip the content locally. The .zip file contains the following files and folders:

\SampleWebCenterSpacesExtensions	
SampleWebCenterSpacesExtensions.jws	Use this workspace to review samples, develop extensions, as well as build and deploy shared libraries that contain your custom code.
\LanguageAdditions	Project containing a sample resource bundle for the Catalan language. Use this project to enable users to access the Spaces application using the Catalan language.

\ProfileAttributes	Project containing code that overrides default profile attributes and enables you to extend user profiles by adding your own attributes.
\ProfileCrawler	Project containing an Oracle SES crawler that searches custom user profile attributes.
\RuleBasedResources	Project containing a custom class that determines which skin to use based on the 'organization' to which the current user belongs (`UserSkinBean.java`).
\SampleTaskFlows	Project containing a custom task flow that displays a list of pages in a bulleted list.
\TaskFlowCustomPane	Project containing code that enables you to attach a custom property pane for a task flow. The custom pane is displayed in Composer's Component Properties dialog.
\WebServiceConnection	Project containing code to invoke an external Web Service from the Spaces application.
\WCAppEarSharedLibExtension	Project that generates an application-level shared library (EAR file) containing custom code. (Only required for the profile crawler project).
\WebCenterSpacesSharedLibExtension	A customizable Spaces project that generates an application-level shared library (WAR file) containing your custom code.
\SourceFiles	Out-of-the-box content for you to edit/customize.
\langs	Default resource bundles for all languages
\deploy	Required JDeveloper directory
\mds	Required JDeveloper directory
\resourcebundles	Required JDeveloper directory
\src	Required JDeveloper directory

4. Make a copy of the sample workspace in which to develop your extensions:

   a) Rename the **SampleWebCenterSpacesExtensions** folder, for example, as `AcmeWebCenterSpacesExtensions`.

   b) Rename the **SampleWebCenterSpacesExtensions.jws** workspace, for example, as `AcmeWebCenterSpacesExtensions.jws`.

   c) Open the following files, and replace "**SampleWebCenterSpacesExtensions**" with the name of your workspace copy:

- `config.properties`

- `WebCenterSpacesSharedLibExtension/build.properties`

For example, replace "SampleWebCenterSpacesExtensions" with "AcmeWebCenterSpacesExtensions".

d) Save the changes.

5. Open your copy of **SampleWebCenterSpacesExtensions.jws** in JDeveloper.

This is the workspace that you will use to review the samples, develop Spaces extensions, as well as build and deploy shared libraries that contain your custom code.
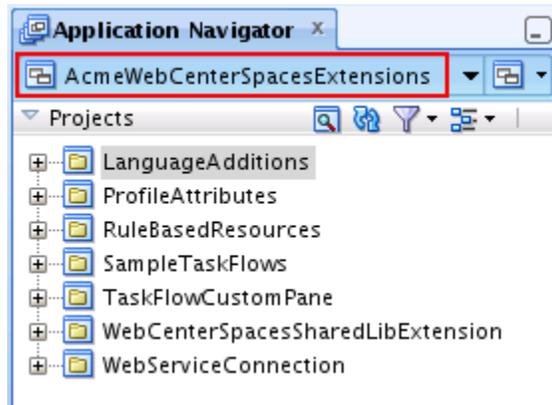
The following projects display in the Application Navigator:

- **WebCenterSpacesSharedLibExtension** – Project used to generate a customized shared library WAR file for Spaces.

  You can include all the samples in this WAR file or exclude samples that you do not want by editing the **WebCenterSpacesSharedLibExtension** deployment profile. You can also add new projects of your own.

  The name of the shared library WAR file is customizable through the customer.library.name property in **config.properties**. For details, see the next section, *Setting Build and Deployment Options (config.properties).*

- **Six sample projects**:

  - **LanguageAdditions** – Project containing a sample resource bundle for the Catalan language.

  - **ProfileAttributes** – Project containing code that overrides the default profile attributes, Department and Photo, and enables you to extend user profiles by adding your own attributes.

- **RuleBasedResources** – Project containing a sample class that determines which skin to use based on the 'organization' to which the current user belongs (`UserSkinBean.java`).

- **SampleTaskFlows** – Project containing a sample task flow that displays a list of pages in a bulleted list.

- **TaskFlowCustomPane** – Project containing code to attach a custom property pane, with a single parameter, for a task flow.

- **WebServiceConnection** – Project containing code to invoke an external Web Service from the Spaces application. The Web Service accepts an IP address and returns the country code.

Note: The sample also contains a seventh project, **ProfileCrawler**, which you can use to deploy a custom profile attribute crawler. By default, this project is not displayed and must be exposed explicitly. For more information, see "*Adding a Crawler for Custom Profile Attributes*"*.*

6. (Optional) Modify the WAR file deployment profile.

All six sample projects are *included* in the custom Spaces shared library WAR file. If you do not want to deploy all the samples, you can edit the deployment profile and select only the projects you want.

> **Note:** If you want to deploy the **ProfileCrawler** project, read the section "*Adding a Crawler for Custom Profile Attributes"*.

a) In the Application Navigator, right-click the **WebCenterSpacesSharedLibExtension** project and choose **Project Properties**.

b) Select **Deployment**, highlight **custom_webcenter_spaces_war**, and then click **Edit**.

c) Select **Library Dependencies**.

By default, all six sample projects are *included* in the custom Spaces shared library (WAR file):
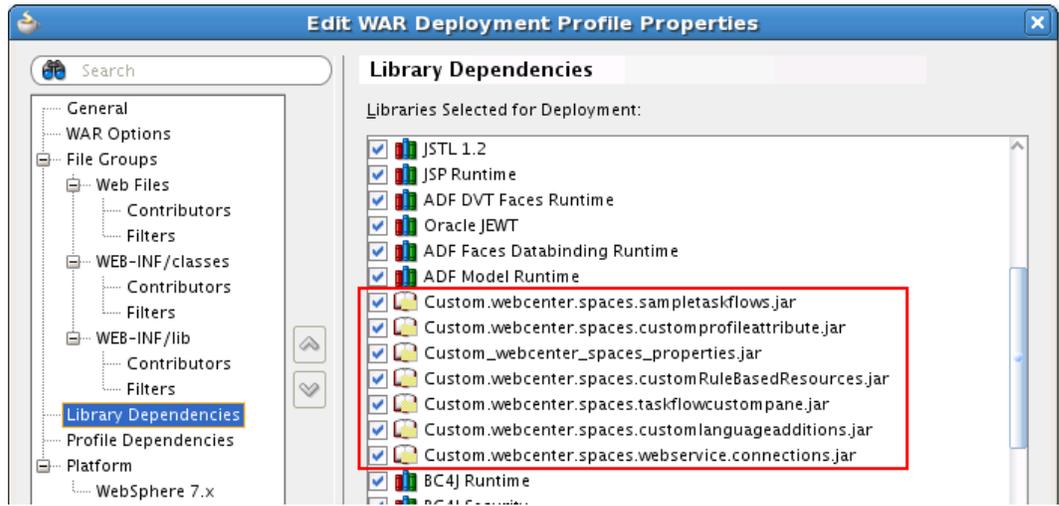
If you want to exclude one or more samples you must edit the library dependency list (see next step).

d) (Optional) Exclude JARs associated with unwanted sample projects.

For example, only select **Custom.webcenter.spaces.sampletaskflows.jar** and **Custom.webcenter.spaces.customRuleBasedResources.jar** to include these projects.
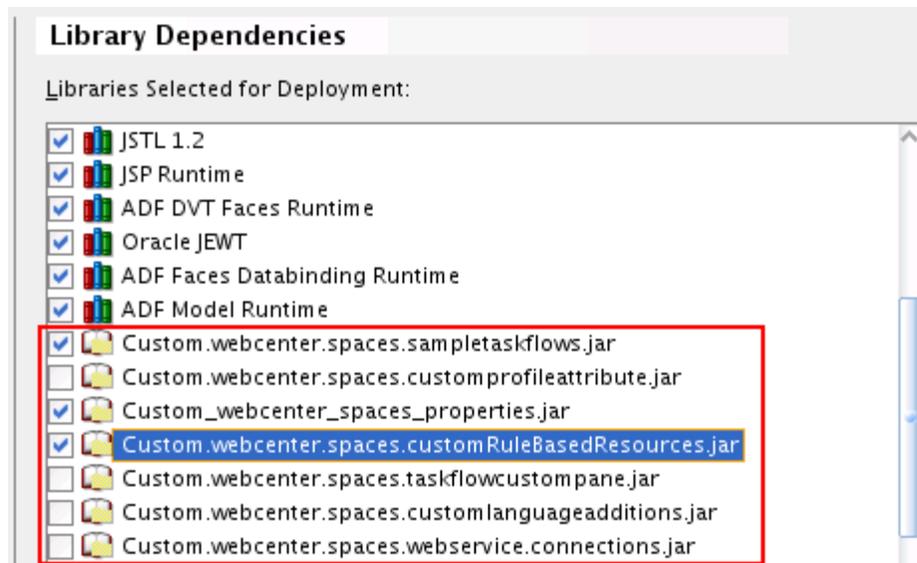
Note: `Custom_webcenter_spaces_properties.jar` is provided for backward compatibility. You can ignore this JAR if you are customizing Spaces for the first time.

The next section describes how to set options that enable you to deploy your custom shared library WAR file to a Spaces managed server.

Setting Build and Deployment Options (config.properties)

Before building and deploying shared library extensions to the Spaces managed server you must provide some information about your environment and your Spaces installation.

Use the configuration file `config.properties` to configure build and deployment properties for your custom shared library:

1. In JDeveloper, open your copy of **SampleWebCenterSpacesExtensions.jws**.

2. Expand **WebCenterSpacesSharedLibExtension**, and open the **Resources** folder.

3. Open **config.properties**.

4. Enter information about your JDeveloper environment and Spaces installation as shown in the table.

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

| PROPERTY NAME	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 WebLogic Scripting Tool WLST executable file.
	Either `wlst.cmd` (on Windows) or `wlst.sh` (on Linux).
extending.spaces.home.dir	Path to your copy of the `SampleWebCenterSpacesExtensions` workspace.
	For example, C:\AcmeWebCenterSpacesExtensions
oracle.jdeveloper.ojdeploy.path	Path to `ojdeploy.exe`. (Windows) or `ojdeploy` (Linux).
wls.port	Port number on which the WebLogic Administration Console is running.
wls.host	Host machine on which the Spaces application is running and where the custom Spaces 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	Managed server on which the Spaces shared library is to be deployed. For example: `WC_Spaces`
	In a clustered environment, enter a comma separated list of all the target servers, for example: `WC_Spaces1,WC_Spaces2`

wls.webcenter.app.target	Managed server on which the Spaces application (`webcenter.ear`) is deployed. For example: `WC_Spaces` In a clustered environment, enter a comma separated list of servers, for example: `WC_Spaces1,WC_Spaces2`
webcenter.app.name	Name of the Spaces application. Always `webcenter`.
customer.library.name	Name for your custom shared library WAR file. For example: `com.acme.custom.webcenter.spaces`
customer.ear.library.name	Name for your custom shared library EAR file. For example: `com.acme.custom.webcenter.spaces.fwk` (Only required if you deploy the ProfileCrawler project).
customer.name	Company name or identifier.
restart.implementation. version.suffix	Controls incremental implementation version numbers for the custom shared library WAR file.
restart.implementation. version.suffix.ear	Controls incremental implementation version numbers for the custom shared library EAR file. (Only required if you are using the profile crawler).

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

Building and Deploying a Custom Shared Library for the Spaces Application

After setting up a deployment profile for the **WebCenterSpacesSharedLibExtension** project and configuring **config.properties**, you can build and deploy your custom Spaces shared library WAR file:

1. Open your copy of **SampleWebCenterSpacesExtensions.jws**.

2. In the Application Navigator, select **WebCenterSpacesSharedLibExtension**.

3. To build a new version of the Spaces shared library WAR, open the **Run Ant** dropdown menu, and then select **clean-stage**:
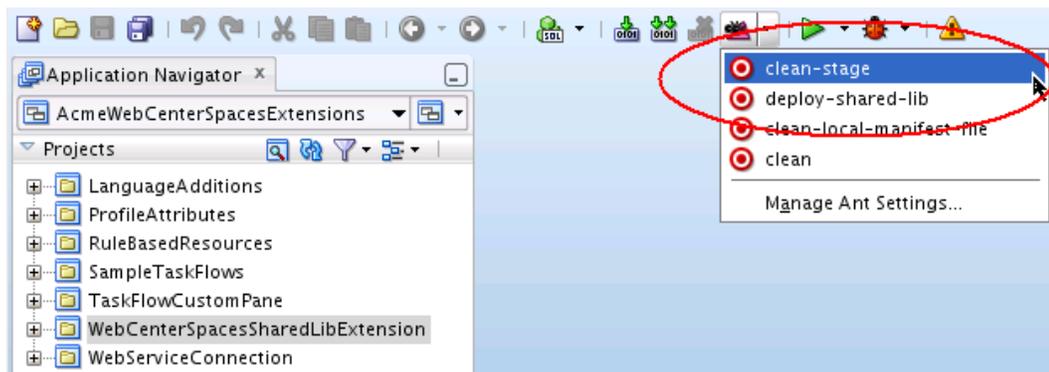
This generates a new version of the shared library WAR file. The implementation version number associated with the new .WAR is saved to `<UnzipDir>/<Your>WebCenterSpacesExtensions/WebCenterSpacesSharedLibExtension/META-INF/MANIFEST.MF`.

4. To deploy the custom Spaces shared library, open the **Run Ant** 🐜 dropdown menu, and then select **deploy-shared-lib**:
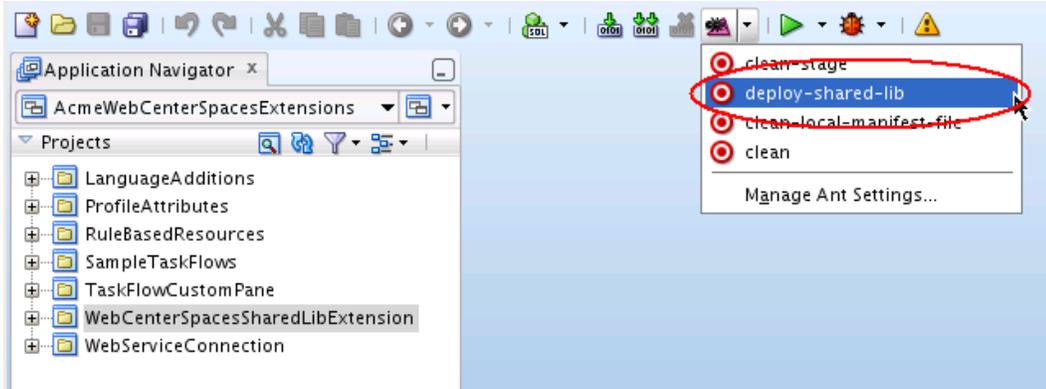
A new version of the shared library is deployed. To verify the new deployment, login to the WLS Administration Console, navigate to the Deployment Overview page, and check the implementation version displayed. For example, if you set the customer.library.name property (in config.properties) to com.acme.webcenter.spaces, navigate to **Deployments > com.acme.webcenter.spaces >Overview**.

The Spaces application only uses the latest shared library version. If you go through several "change-build-deploy-test" iterations, incremental versions are retained by default. You can use the WLS Administration Console to remove unwanted shared library versions. For details, see "Reverting to a Previous Shared Library Version" in *Developer's Guide for Oracle WebCenter Portal*.

If the latest version is not active, see "Troubleshooting Shared Library Deployment" in *Developer's Guide for Oracle WebCenter Portal*.

5. Include the name of your custom shared library WAR file in Space's shared library list.

For detailed steps, refer to "Rebuilding the Spaces Shared Library List" in *Developer's Guide for Oracle WebCenter Portal*.

Note: In the latest version of DesignWebCenterSpaces.jws, use the **Run Ant** 🐜 dropdown menu to build and deploy the shared library list.

📖 *Developer's Guide for Oracle WebCenter* Portal *(11.1.1.6.0)* is available from Oracle Technology Network:
http://download.oracle.com/docs/cd/E23943_01/webcenter.1111/e10148/jpsdg_wcsres.htm

Removing Spaces Extensions

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 do so at any time. For details, see "Reverting to a Previous Shared Library Version" in *Developer's Guide for Oracle WebCenter Portal*.

This section only describes the steps specific to the samples discussed in this document.

Remove Changes Related to the Customized Login Task Flow

If you configured OAM and customized the Login task flow (as described in the section "*Sample – Customizing the Login Task Flow*"), you must perform the following steps to remove the changes you made:

1. In `setDomainEnv.sh`, remove the following `EXTRA_JAVA_PROPERTIES`:

   `EXTRA_JAVA_PROPERTIES` "**-Doracle.webcenter.spaces.osso=true**"

   `export EXTRA_JAVA_PROPERTIES`

2. Remove `OBACCESS_INSTALL_DIR`:

   `OBACCESS_INSTALL_DIR="<Path to Access Server SDK Install Dir>/AccessServerSDK"`

   `export OBACCESS_INSTALL_DIR`

3. Remove `jobaccess.jar` from the `CLASSPATH`:

   `if [ "${POST_CLASSPATH}" != "" ] ; then`

   `POST_CLASSPATH="${COMMON_COMPONENTS_HOME}/modules/oracle.jrf_11.1.1/jrf.jar${CLASSPATHSEP}${OBACCESS_INSTALL_DIR}/oblix/lib/`**`jobaccess.jar`**`${CLASSPATHSEP}${POST_CLASSPATH}"`

   `export POST_CLASSPATH`

   `else`

   `POST_CLASSPATH="${COMMON_COMPONENTS_HOME}/modules/oracle.jrf_11.1.1/jrf.jar${CLASSPATHSEP}${OBACCESS_INSTALL_DIR}/oblix/lib/`**`jobaccess.jar`**`"`

   `export POST_CLASSPATH`

4. Remove the `LD_LIBRARY_PATH` setting:

   `LD_LIBRARY_PATH="${OBACCESS_INSTALL_DIR}/oblix/lib:${WL_HOME}/server/native/linux/i686${CLASSPATHSEP}${LD_LIBRARY_PATH}"`

   `export LD_LIBRARY_PATH`

5. In `commEnv.sh`, remove the `WEBLOGIC_CLASSPATH` entry for `webcenter_oam.jar`:

```
WEBLOGIC_CLASSPATH="${WEBLOGIC_CLASSPATH}${CLASSPATHSEP}/scr
atch/netpoint/webcenter_oam.jar"

export WEBLOGIC_CLASSPATH
```

6. In WebLogic Server Administration Console, navigate to **Home > Startup and Shutdown Classes** and delete the startup and shutdown classes (`OAMConfigShutdown` and `OAMConfigStartup`).

7. Remove SSO configuration for the Spaces application (if required).

   For detail, see *Administrator's Guide for Oracle WebCenter Portal*. Available from Oracle Technology Network:
   http://download.oracle.com/docs/cd/E23943_01/webcenter.1111/e12405/wcadm_security_sso.htm

8. Restart the domain.

Sample – Adding a New Language

Spaces provides runtime translations for 27 languages and 100 different locales. If you want your application to support an additional language, you must translate Spaces strings into your new language within a resource bundle, update two language configuration files (`supported-languages.xml` and `faces-config.xml`), and then deploy your language updates to a custom shared library.

The `SampleWebCenterSpacesExtensions` workspace provides the **LanguageAdditions** project that includes a sample resource bundle for the Catalan language (`WebCenterSpacesResourceBundle_ca`) and the configuration file `faces-config.xml`. The steps below describe how to update and deploy these files to a shared library so that Spaces users can work in the Catalan language. The steps also describe how `supported-languages.xml` is updated and the changes uploaded to MDS.

The implementation details are as follows, you can perform similar steps to add any new language:

1. Open your copy of **SampleWebCenterSpacesExtensions.jws** in JDeveloper.

2. Expand the **LanguageAdditions** project.

3. Open **LanguageAdditions\Application Sources\oracle.webcenter.webcenterapp\resource\WebCenterResourceBundle_ ca.java**
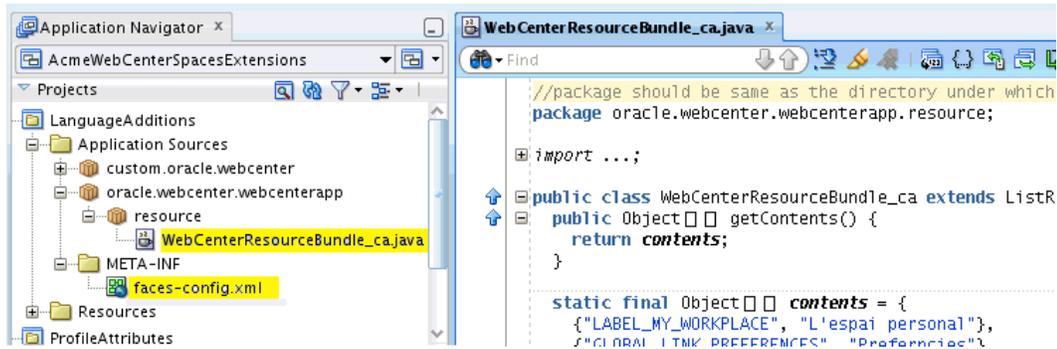
`WebCenterResourceBundle_ca.java` contains Catalan translations for strings in the resource bundle `WebCenterResourceBundle.xrts`.

Note: String translations for new languages, such as those in `WebCenterResourceBundle_ca.java`, must always be placed in the same directory as the original resource bundle (.xrts). In the LanguageAdditions project, for example, `WebCenterResourceBundle.xrts` is located at `oracle\webcenter\webcenterapp\resource` and therefore, the sample Catalan translations for this resource bundle are provided at the same location.

4. To register Catalan as a new language, the appropriate **<supported-locale>** tag is added to `faces-config.xml`.

   a) Open **LanguageAdditions\Application Sources\META-INF\faces-config.xml**.

   b) Note that the code for the new language is included under the **<locale-config>** element.
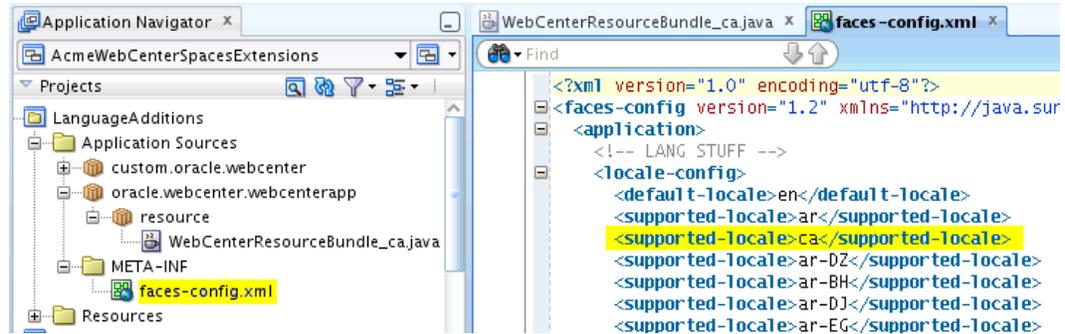
5. To deploy the sample project with translations for the Catalan language, select **LanguageAdditions** in the deployment profile:
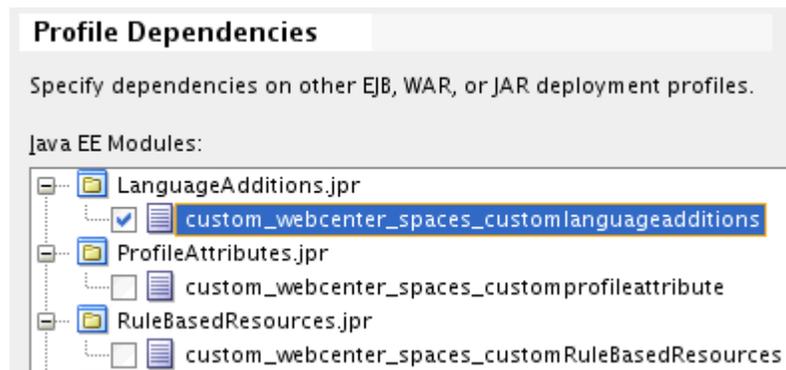
6. Deploy the Catalan translations.

   For details, see section *Building and Deploying a Custom Shared Library for the Spaces Application*.

7. To make the new language available in the Spaces application, add the Catalan **<language>** tag to `supported-languages.xml`.

   a) Start the WebLogic Scripting Tool (WLST) located at `WC_ORACLE_HOME/common/bin`.

      - On UNIX, start WLST using `wlst.sh`.

      - On Windows, use `wlst.cmd`.

b) Use the WLST command **exportMetadata** to download `supported-languages.xml` from MDS:

    For example:

    ```
    exportMetadata(application='webcenter', server='WC_Spaces',
    toLocation='/tmp/myMDSfiles',
    docs='/oracle/webcenter/webcenterapp/metadata/supported-
    languages.xml')
    ```

c) Add the following `<language>` tag to the exported `supported-languages.xml` file:

    ```
    <language name=" Catalan" id="ca" used="true"
    activeicon="nlse_a.gif" inactiveicon="nlse.gif"
    translated="true"/>
    ```

    In the `exportMetadata` example, the file is exported to `/tmp/myMDSfiles`.

d) Use the WLST command **importMetadata** to upload the updated XML file to MDS:

    For example:

    ```
    importMetadata(application='webcenter', server='WC_Spaces',
    fromLocation='/tmp/myMDSfiles',
    docs='/oracle/webcenter/webcenterapp/metadata/supported-
    languages.xml')
    ```

8. Log in to the Spaces application to verify that the new language displays (check Spaces Administration and user Preference screens).

    If the sample with the new language deployed successfully, Catalan displays in the language list:
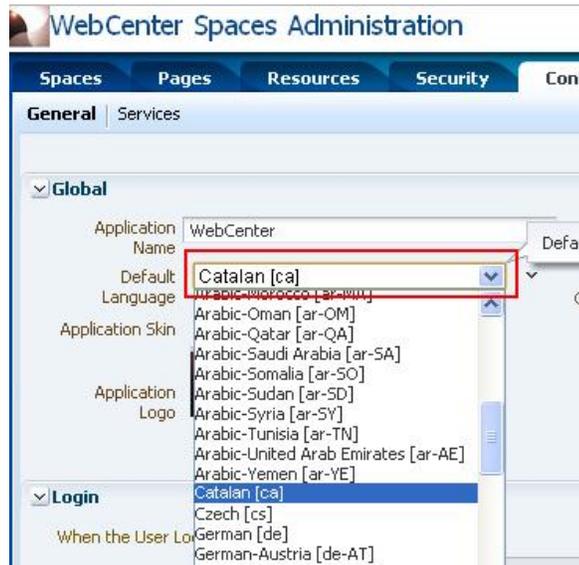
Sample - Customizing User Profiles

In Spaces, the 'profile' feature provides a variety of views into your own and other users' personal profile information. User profiles include information such as your mail address, phone number, office location, department, manager, direct reports, and so on. Most profile attributes are stored and read from the Spaces LDAP identity store; there are three exceptions--the profile photo, status message, and expertise.

The **ProfileAttributes** project in the SampleWebCenterSpacesExtensions workspace demonstrates different ways to customize the default user profile attribute set. For example, you can:

- Override the default profile attributes

- Add custom profile attributes of your own

- Override the default profile page

Overriding the Default Profile Attributes

The ProfileAttributes project includes CustomProfileImpl.java, a sample Java class that overrides a default attribute source, and configures `people-profile-config.xml` to point to the new class.

The sample Java class shows how to override two default LDAP attributes (Department and Photo) as follows:

- o **Department** – Is replaced with the hard coded string **Sales Department** as follows:

  ```
  public String getDepartment()
              throws ProfileException
              {return "Sales Department";}
  ```

- o **Photo** – User photos are replaced with images in an alternative store (file system in this case).

Important considerations:

- **Performance:** When implementing custom logic to fetch values from a database, remember that the JDBC call executes each time any profile is accessed. Spaces does not cache profile data. Keep this in mind when coding your custom class to minimize any impact on performance.

- **Search:** Profile attribute searches are always against the configured identity store. This is still the case even if you customize an attribute source.

For more information, see "Deploying the Sample ProfileAttributes Project".

Adding Custom Profile Attributes

The ProfileAttributes project also includes `ExtendedProfileAttribes.java`, a sample Java class that enables you to include additional profile attributes. The sample includes two additional attributes:

- **Project**

- **SIP Address**

After deploying `ExtendedProfileAttribes.java` you can reference profile attributes defined in the file in Spaces using EL such as:

`#{wcProfileCustomAttribute['smith'].sip}`

If you add your own profile attributes to `ExtendedProfileAttribes.java` you can reference them in the same way. For more information, see "Deploying the Sample ProfileAttributes Project".

Overriding the Default Profile Page

The ProfileAttributes project includes `CustomView.jspx`, a sample profile page to replace the default profile page supplied with the Spaces application. For more information, see "Deploying a Custom Profile Page".

Deploying the Sample ProfileAttributes Project After Customizing or Adding Profile Attributes

The following steps describe how you can customize or add attributes displayed in Spaces user profiles using the **SampleWebCenterSpacesExtension** workspace:

1. Open your copy of **SampleWebCenterSpacesExtensions.jws** in JDeveloper.

2. To override default profile attributes:

   a) Review the sample class by navigating to **ProfileAttributes > Application Sources > custom.webcenter.spaces > CustomProfileImpl.java**

   You can use JDeveloper to implement a customized profile implementation class of your own. If you do, you can specify any valid name for your custom class and package name.

   Tip: Use Ctrl+Space to view/add all the attribute implementation methods that are available, using JDeveloper's Code Completion Insight functionality shown below:
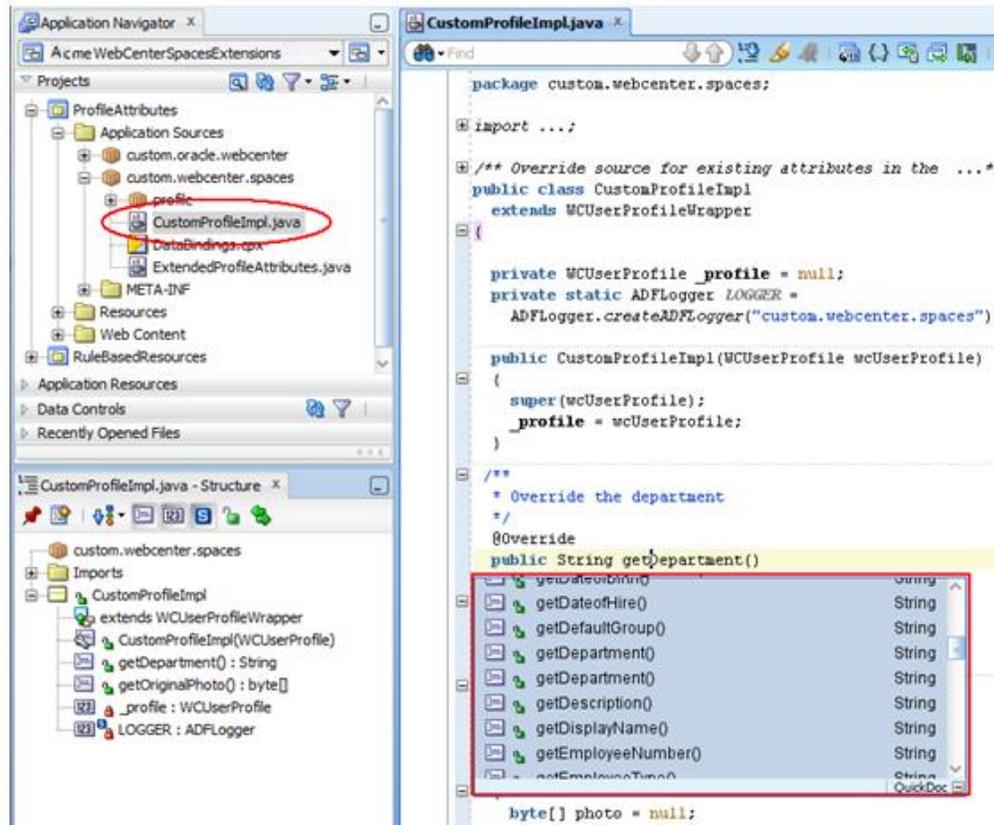
b) Open **ProfileAttributes\Application Sources\custom.oracle.webcenter\peopleconnections\profile\metadata\people-profile-config.xml**:
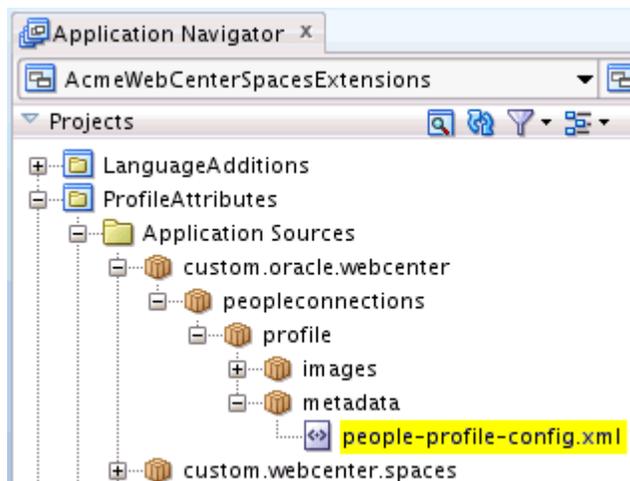
c) Use the `user-obj-impl` setting to point to the custom class, as shown in the image.

The sample specifies CustomProfileImp.java. Any valid class name and package name is acceptable here.

Notes:

- o Do not change anything else in this file.

- o Only reference one implementation class from the people-profile-config.xml file.

- o Do not rename the package containing people-profile-config.xml or modify the structure in any way.

3. To add profile attributes:

   a) Review the sample class by navigating to **ProfileAttributes > Application Sources > custom.webcenter.spaces > ExtendedProfileAttributes.java**

   You can add your own profile attributes to the file or deploy the sample attributes (Project and SIP Address).
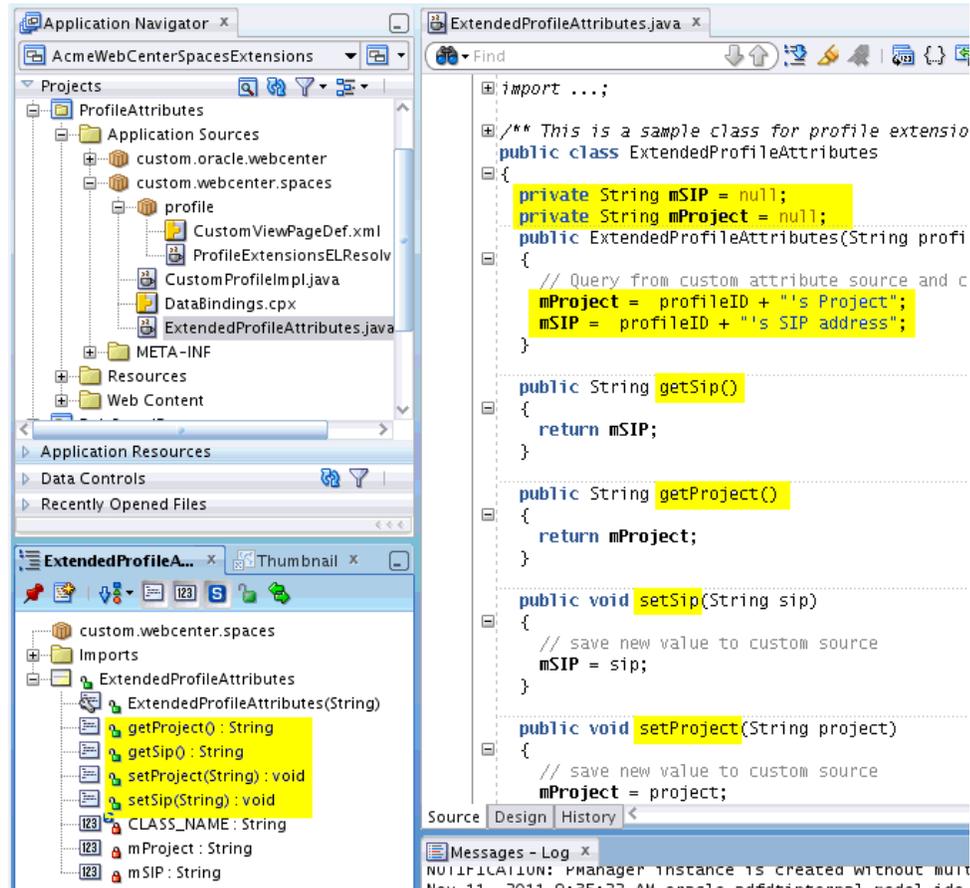
b) Add a getter (and setter if you need to save) for additional profile attributes, as required.

4. If you want to deploy the sample or your own extensions, ensure that the **ProfileAttributes** project JAR is selected in the deployment profile:
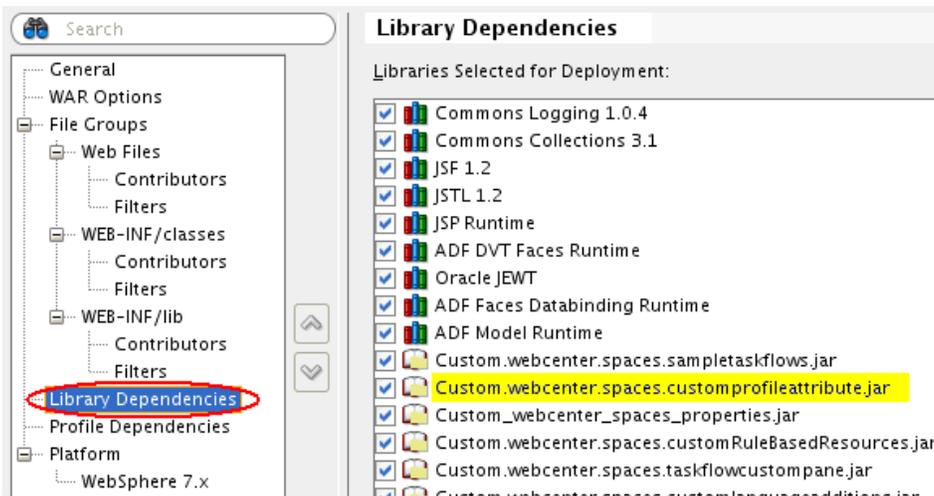
5. Deploy the custom profile attribute class.

   For details, see section [Building and Deploying a Custom Spaces Shared Library](#).

6. Log in to the Spaces application, and open your profile to verify that the customized attributes display.

   If you deployed the samples provided:

   a) Verify the following:

      o **Department** – Displays 'Sales Department'

      o **Photo** –

         ▪ monty.png displays when logged in with the user name Monty

         ▪ pat.png displays when logged in with the user name Pat

         ▪ a default shadow image displays if an alternative image is not available for the logged in user

      If new images do not display, click the Refresh link to clear cached data.

   b) Use EL to reference the new profile attributes **Project** and **SIP Address**.

      To do this, you must first expose each new attribute as a component in the Resource Catalog as follows:

      i) Navigate to the Resource Catalog design tab (**Administration > Resources > Resource Catalogs**).

ii) Create a new Resource Catalog, if required.

iii) Edit the catalog, and choose **Add >Component.**

iv) Name the component, for example **Project Attribute**.

v) Add the appropriate XML, for example, add the following snippet for the 'Project' component:

```
<af:panelLabelAndMessage
xmlns:af="http://xmlns.oracle.com/adf/faces/rich"
label="Project " labelStyle="font-weight:bold;" id="#">

   <af:inputText
   value="#{wcProfileCustomAttribute[(pageFlowScope.userId ==
   null ? securityContext.userName :
   pageFlowScope.userId)].project}" simple="true" columns="40"
   maximumLength="50" shortDesc="Enter project details" id="#"/>

</af:panelLabelAndMessage>
```
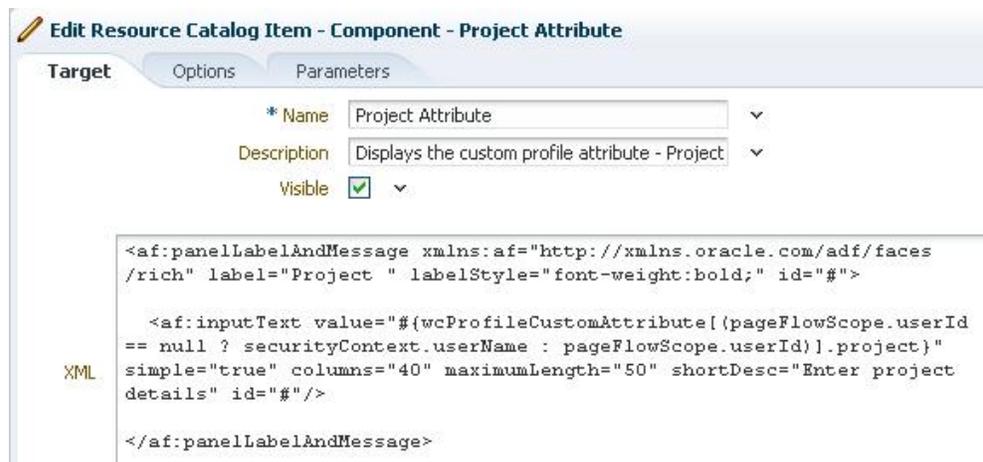

vi) Create a similar component for the **SIP Address** attribute, that is, repeat steps iii –v, changing the `wcProfileCustomAttribute` EL to reference '**sip**' and enter an appropriate description.

vii) If you have not done so already, configure pages to use the updated Resource Catalog. Do one of the following:

- If you want to add the new profile attributes to *system pages or business role pages*, navigate to Spaces Administration (**Administration > Configuration > General**), and ensure **Resource Catalog for Business**

Role Pages specifies the Resource Catalog that contains your profile attributes.

- If you want to add the new profile attributes to *pages in the Home space*, navigate to Spaces Administration (**Administration > Configuration > General**), and ensure **Resource Catalog for Home Space** specifies the Resource Catalog that contains your profile attributes.

- If you want to add the new profile attributes to *pages in a particular Space*, navigate to the space administration (Space **Settings > General**), and ensure **Resource Catalog for Pages** specifies the Resource Catalog that contains your profile attributes.

viii) Navigate to the profile page (**Administration > Pages > System Pages)** or any other page in which you want to add the new profile attributes.

ix) Click **Edit Page** to open the page in Composer.

x) Add the profile attribute to a page or task flow:

- To add the profile attribute to a page, click **Add Content**, and then select the profile attribute you require from the catalog. For example, select **Project Attribute**.

- To add the profile attribute to a task flow, select the task flow, click **View > Source**, click the task flow to Edit it, and then select the profile attribute you require from the catalog.

c) Alternatively, upload *pre-customized* user profile 'About**'** pages directly to spaces that already include EL references to the new sample attributes **Project** and **SIP Address**. Two pre-customized About pages are provided with the sample:

- `viewemployee.jsff.xml` – About page customizations for view mode; display two new attributes **Project** and **SIP Address**.

- `editemployee.jsff.xml` – About page customizations for edit mode; display the **Project** attribute so users can edit this value.

Use WLST to upload the customized About page samples to Spaces as follows:

i) Navigate to your WebCenter Portal Oracle home directory and invoke the WLST script:

- (UNIX) `WC_ORACLE_HOME/common/bin/wlst.sh`

- (Windows) `WC_ORACLE_HOME\common\bin\wlst.cmd`

ii) Connect to the Administration Server for Oracle WebCenter Portal. For example:

```
connect(username='weblogic', password='mypassword',
url='myhost.example.com:7001')
```

iii) Upload About page customizations:

```
importMetadata(application='webcenter', server='WC_Spaces',
fromLocation='<sampleziplocation>ProfileAttributes/mds/',
docs='/oracle/webcenter/peopleconnections/profile/view/jsf/reg
ions/ootbviews/mdssys/cust/site/webcenter/viewemployee.jsff.xm
l')
```

```
importMetadata(application='webcenter', server='WC_Spaces',
fromLocation='<sampleziplocation>ProfileAttributes/mds/',
docs='/oracle/webcenter/peopleconnections/profile/view/jsf/reg
ions/ootbedit/mdssys/cust/site/webcenter/editemployee.jsff.xml
')
```

iv) Log in to the Spaces application.

v) Open your user profile and click **About** to access your full profile details
(`viewemployee.jsff.xml`).

The new **Project** and **SIP Address** attributes display. For example:

vi) Click **Edit** to see the About page in edit-mode (`editemployee.jsff.xml`).

The new **Project** attribute displays. For example:

Deploying a Custom Profile Page

The following steps, describe how to deploy a custom profile page (`CustomView.jspx`) using the **SampleWebCenterSpacesExtension** workspace:

1. Open your copy of **SampleWebCenterSpacesExtensions.jws** in JDeveloper.

2. Navigate to: **ProfileAttributes > Web Content > custom > webcenter > spaces > profile > CustomView.jspx**
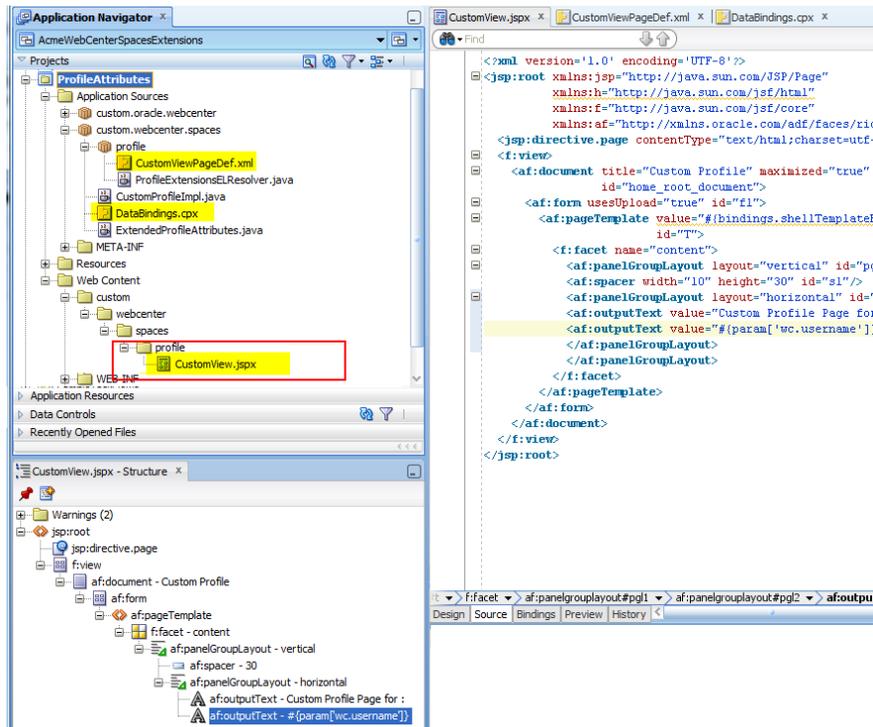
3. Review or modify the sample page `CustomView.jspx`.

4. Configure the Spaces application to use the new profile page.

    Modify the table **WC_PPL_COMMON_SETTING** as described in step b below. The SETTING_KEY column uses:

    - **spaces-settings.galleryPage.pathForSelf** to specify the profile page displayed for the currently logged in user

    - **spaces-settings.galleryPage.pathForOthers** to specify the profile page displayed for any other user

    a) Log in to SQLPlus as DBA or another administrative user to the WebCenter database.

b) Execute the following commands:

```
INSERT INTO WC_PPL_COMMON_SETTING (ID, APPLICATION_ID, SCOPE_ID,
SERVICE_ID, USER_ID, TASKFLOW_INST_ID, SETTING_KEY, SETTING_VALUE)
VALUES ('98457a7d-72df-4e92-9396-9a63b7f414e7', 'webcenter',
'defaultScope', 'oracle.webcenter.peopleconnections.profile',
'SYSTEM', 'SITE', 'spaces-settings.galleryPage.pathForSelf',
'/custom/webcenter/spaces/profile/CustomView.jspx');

INSERT INTO WC_PPL_COMMON_SETTING (ID, APPLICATION_ID, SCOPE_ID,
SERVICE_ID, USER_ID, TASKFLOW_INST_ID, SETTING_KEY, SETTING_VALUE)
VALUES ('98457a7d-72df-4e95-9396-9a63b7f414e7', 'webcenter',
'defaultScope', 'oracle.webcenter.peopleconnections.profile',
'SYSTEM', 'SITE', 'spaces-settings.galleryPage.pathForOthers',
'/custom/webcenter/spaces/profile/CustomView.jspx');
```

Note: If these rows already exist, update each SETTING_VALUE column to:
/custom/webcenter/spaces/profile/CustomView.jspx

c) Commit the updates.

5. Hide the out-of-the-box "My Profile" business role page from everyone:

a) Use the WLST command **exportMetadata** to export pages.xml from MDS.
Export pages.xml configured for public users (anonymous-role) and
authenticated users (authenticated-role) as follows:

```
exportMetadata(application='webcenter', server='WC_Spaces',
toLocation='/scratch/myMDSfiles',
docs='/oracle/webcenter/page/scopedMD/s8bba98ff_4cbb_40b8_beee_296c
916a23ed/role/anonymous-role/pages.xml')

exportMetadata(application='webcenter', server='WC_Spaces',
toLocation='/scratch/myMDSfiles',
docs='/oracle/webcenter/page/scopedMD/s8bba98ff_4cbb_40b8_beee_296c
916a23ed/role/authenticated-role/pages.xml')
```

b) Open each pages.xml file in a text editor, locate the business role page
MyProfileMainView.jspx, and set the property hidden="true":

```
<!-- Business Role Pages -->
<pageDef
id="Page_2eb852ac_10f5902cb2f__7f10"
contentMRef="/oracle/webcenter/page/scopedMD/s8bba98ff_4cbb_40b8_be
ee_296c916a23ed/businessRolePages/MyProfileMainView.jspx"
shared="true"
hidden="true"/>
```

> **Note:** If an entry does not exist for `MyProfileMainView.jspx` in `anonymous-role/pages.xml` then no changes are required to this file.

c) Upload `pages.xml` updates back to the MDS as follows:

```
importMetadata(application='webcenter', server='WC_Spaces',
fromLocation='/scratch/myMDSfiles',
docs='/oracle/webcenter/page/scopedMD/s8bba98ff_4cbb_40b8_beee_296c
916a23ed/role/anonymous-role/pages.xml')
```

```
importMetadata(application='webcenter', server='WC_Spaces',
fromLocation='/scratch/myMDSfiles',
docs='/oracle/webcenter/page/scopedMD/s8bba98ff_4cbb_40b8_beee_296c
916a23ed/role/authenticated-role/pages.xml')
```

6. To deploy the sample page or your own profile page, ensure that the **ProfileAttributes** project JAR is selected in the deployment profile:
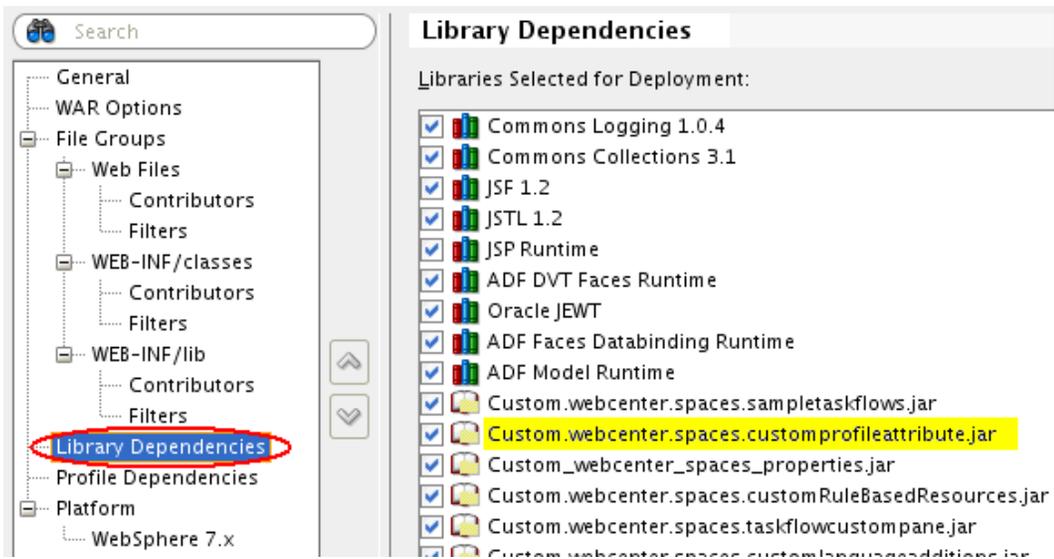
7. Deploy the custom profile page.

   For details, see section [Building and Deploying a Custom Spaces Shared Library](#).

8. Log in to the Spaces application, and then open someone's profile to verify that the custom page displays.

If you deployed the sample, the text "**Custom Profile Page for: <user name>**" displays:
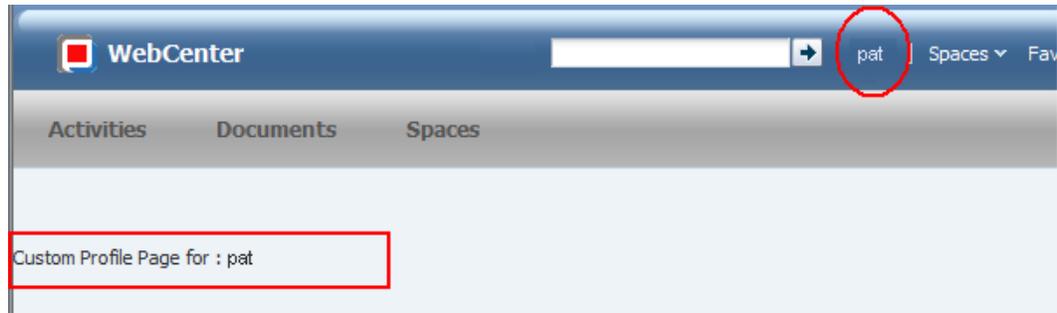
To revert to the original profile page

1. Log in to SQLPlus as DBA or another administrative user to the WebCenter database.

2. Execute the following commands:

   ```
   delete from WC_PPL_COMMON_SETTING where SETTING_KEY = 'spaces-
   settings.galleryPage.pathForSelf';
   ```

   ```
   delete from WC_PPL_COMMON_SETTING where SETTING_KEY = 'spaces-
   settings.galleryPage.pathForOthers';
   ```

3. If you updated the row (by deleting and then inserting updated values), restart the Spaces managed server (WC_Spaces).

Sample – Adding a Crawler for Custom Profile Attributes

User profile attributes that you add through the ProfileAttributes project are not automatically searchable in the Spaces application. If you are using Oracle Secure Enterprise Search (Oracle SES) to search the Spaces application, you can deploy a custom crawler implementation that searches your additional profile attributes.

The **ProfileCrawler** project provides a sample profile attribute crawler that you can use:

- **ExtendedProfileAttributesCrawler.java** and
  ExtendedProfileServiceCrawlerQueryResult.java – contain custom code to crawl custom profile attributes and populate Oracle SES with the data

- **service-definition.xml** – defines the name of the sample crawler class:

  ```
  <search-crawlable-definition id="oracle.webcenter.peopleconnections.profile.custom"

        xmlns="http://xmlns.oracle.com/webcenter/search"

        version="11.1.1.0.0"

        scoped="false">

    <crawlable-class>

        custom.webcenter.spaces.ExtendedProfileAttributesCrawler

    </crawlable-class>

   </search-crawlable-definition>
  ```

  If you write you own crawler class, you can change the name here.

- **CustomProfileDataSource.java** – enables access to custom profile attribute data. The sample hard codes data for sample custom profile attributes SIP Address/Project, for a single user named 'weblogic' as follows:

  ```
  public static Map<String, Map<String, String>> getCustomAttributesMap()

    { return new HashMap<String, Map<String, String>>()

      { { put("weblogic", new HashMap<String, String>()

        { { put("SIPAddress", "weblogic's SIP address");

          put("Project", "weblogic's  project");}        }); }}; }
  ```

  When you define additional profile attributes of your own, you must replace this with code that fetches your custom profile attribute data from your data source.

The ProfileCrawler project does not display in JDeveloper by default. Perform the following steps to display this project:

1. If you have not done so already, configure the Spaces application to use Oracle SES.

   This includes setting up crawl sources for the Spaces application in Oracle SES, and then configuring the Spaces application to use Oracle SES. For detailed steps, refer to

"Configuring Oracle SES to Search Spaces Applications" in *Administrator's Guide for Oracle WebCenter Portal*.

📖 *Administrator's Guide for Oracle WebCenter Portal* is available from Oracle Technology Network:
http://download.oracle.com/docs/cd/E23943_01/webcenter.1111/e12405/wcadm_search.htm#WCADM9801

2. Enable shared library EAR file deployment for the custom profile crawler project.

Unlike other sample projects, custom code generated from the **ProfileCrawler** project must be packaged in an *application*-level shared library (rather than a *webapp*-level shared library) and therefore, you require a project that generates an application-level shared library (EAR file). Such a project (**WCAppEarSharedLibExtension**) is provided with the sample but it is hidden by default. Follow these steps to expose **WCAppEarSharedLibExtension** and the sample **ProfileCrawler** project:

a) Ensure that your copy of SampleWebCenterSpacesExtensions is not open in JDeveloper.

b) In a text editor, such as Notepad, open your copy of `SampleWebCenterSpacesExtensions.jws`.

c) Uncomment the **ProfileCrawler** project entry.

Change:

```
<!--hash><url n="URL"
path="ProfileCrawler/ProfileCrawler.jpr"/></hash-->
```

To:

```
<hash><url n="URL"
path="ProfileCrawler/ProfileCrawler.jpr"/></hash>
```

d) Uncomment the **WCAppEarSharedLibExtension** project entry.

Change:

```
<!--hash><url n="URL"
path="WCAppEarSharedLibExtension/WCAppEarSharedLibExtension.j
pr"/></hash-->
```

To:

```
<hash><url n="URL"
path="WCAppEarSharedLibExtension/WCAppEarSharedLibExtension.j
pr"/></hash>
```

e) Open your copy of SampleWebCenterSpacesExtensions in JDeveloper to verify that the **ProfileCrawler** project displays, as well as the project that generates the EAR-level shared library (**WCAppEarSharedLibExtension**):
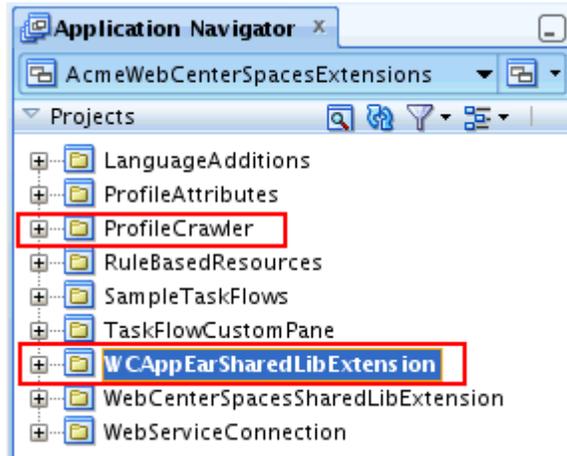
3. Set deployment options for the EAR-level shared library that will contain the custom Oracle SES crawler code.

   a) Open your copy of **SampleWebCenterSpacesExtensions.jws**.

   b) Expand **WebCenterSpacesSharedLibExtension** and open the **Resources** folder.

   c) Specify the following EAR deployment options in **config.properties** (located under the **WebCenterSpacesSharedLibExtension\Resources** folder):

PROPERTY NAME	DESCRIPTION
customer.ear.library.name	Name for your custom shared library EAR file.
	For example: com.acme.custom.webcenter.spaces.fwk
restart.implementation. version.suffix.ear	Controls incremental implementation version numbers for the custom shared library EAR file.
	(Only required if you are using the profile crawler).

   d) Save your updates to config.properties.

4. Build and deploy the custom Oracle SES crawler code:

   a) In the Application Navigator, select **WCAppEarSharedLibExtension**.

   b) To build the Spaces shared library EAR, open the **Run Ant** 🐜 dropdown menu, and then select **clean-stage**:
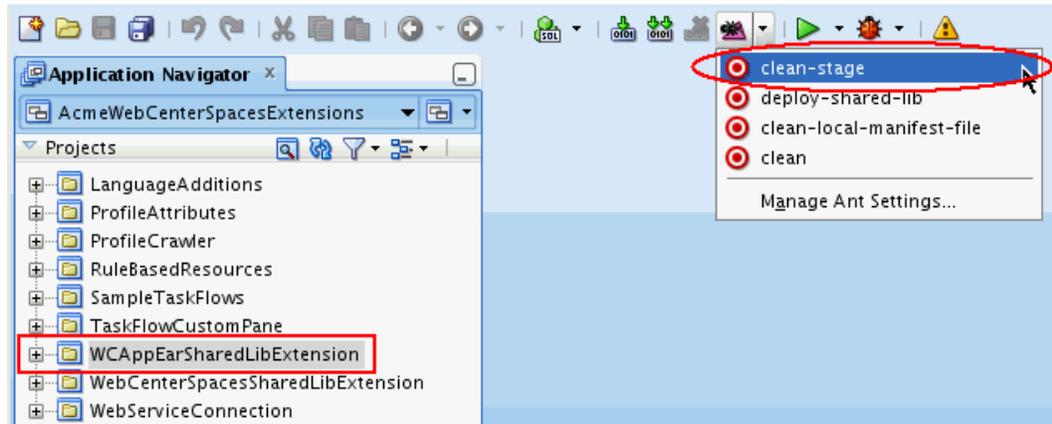
This generates a new version of the shared library EAR file. The implementation version number associated with the new .EAR is saved to `<UnzipDir>/<Your>WebCenterSpacesExtensions/WebCenterSpacesSharedLibExtension/META-INF/EAR/MANIFEST.MF`.

c) To deploy the custom Spaces shared library EAR, open the **Run Ant** dropdown menu, and then select **deploy-shared-lib**:
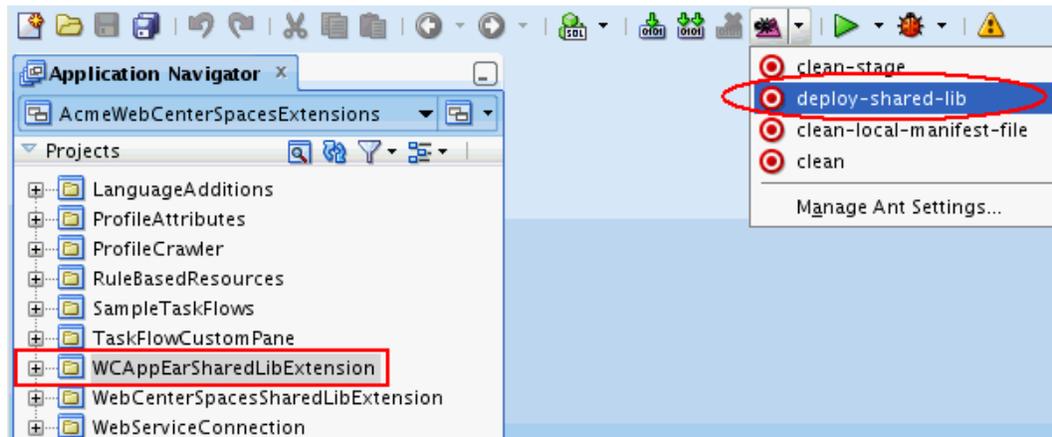
A new version of the shared library EAR is deployed.

d) Open the WLS Administration Console, navigate to the Deployment Overview page, and check that new versions of `custom.webcenter.spaces.fwk` and `<your company>.webcenter.spaces.fwk` are deployed and active.

5. Include the name of your *application-level* shared library EAR file in the Spaces application's shared library list.

a) Down load the required Spaces development workspace from Oracle Technology Network (DesignWebCenterSpaces.jws) and include some required files as described in **steps 4 through 7** in "Downloading a Workspace for Spaces

Development" in *Developer's Guide for Oracle WebCenter Portal (11.1.1.6.0)* for details, see:

http://download.oracle.com/docs/cd/E23943_01/webcenter.1111/e10148/jpsdg_wcsres.htm#JPSDG10089

b) Enable *application-level* shared library list deployment:

i) Ensure that your copy of `DesignWebCenterSpaces.jws` is not open in JDeveloper.

ii) In a text editor, such as Notepad, open `DesignWebCenterSpaces.jws`.

iii) Uncomment the **WCAppEarExtensionLibrary** project entry.

Change:

```
<!--hash><url n="URL"
path="WCAppEarExtensionLibrary/WCAppEarExtensionLibrary.jpr"/></
hash-->
```

To:

```
<hash><url n="URL"
path="WCAppEarExtensionLibrary/WCAppEarExtensionLibrary.jpr"/></
hash>
```

iv) Save the changes.

c) Edit the application-level EAR shared library list:

i) Open `DesignWebCenterSpaces.jws` in JDeveloper.

ii) Expand the **WCAppEarExtensionLibrary** project.

iii) Open **weblogic-application.xml**, which is under the **Resources** folder.

iv) Add a <library-ref> entry for the application-level EAR shared library that contains the ProfileCrawler code.
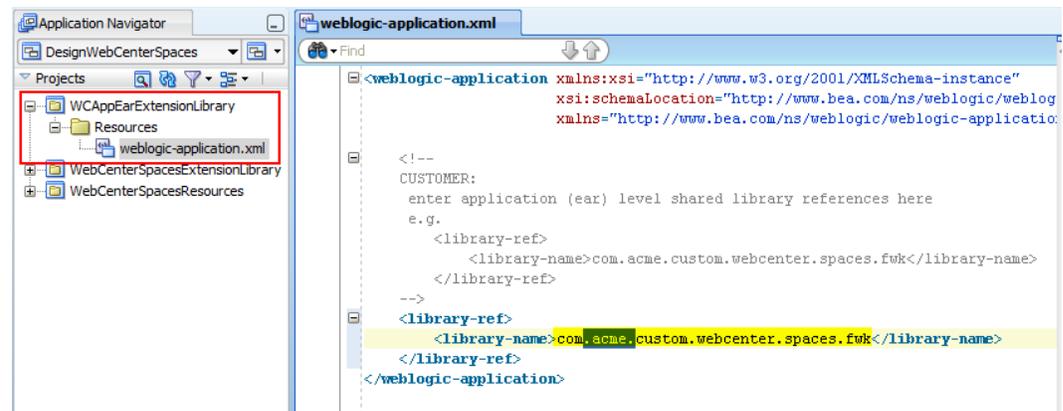
For example, include the library entry:

```
<library-ref>
 <library-name>
    com.mycompanyname.custom.webcenter.spaces.fwk
 </library-name>
</library-ref>
```

Note: Ensure that the shared library EAR file that you reference is deployed on the Spaces managed server.

d) Set build and deployment options for custom.webcenter.spaces.fwk.ear which will include your updates to the application-level EAR shared library list:

   i) Open **config.properties** (located under the **WebCenterSpacesExtensionLibrary** project).

   ii) Enter information about your JDeveloper environment and Spaces installation.

   For more information on each property, see " Configuring config.properties" in *Developer's Guide for Oracle WebCenter* Portal *(11.1.1.6.0)* available at:

   http://download.oracle.com/docs/cd/E23943_01/webcenter.1111/e10148/jpsdg_wcsres.htm#JPSDG10101

   iii) Save your updates to **config.properties**.

e) Build and deploy your updates to the application-level shared library list in `custom.webcenter.spaces.fwk.ear.`

   i) In the Application Navigator, select **WCAppEarExtensionLibrary**.

   ii) To build a new version of `custom.webcenter.spaces.fwk.ear,` open the **Run Ant** dropdown menu, and then select **clean-stage**:
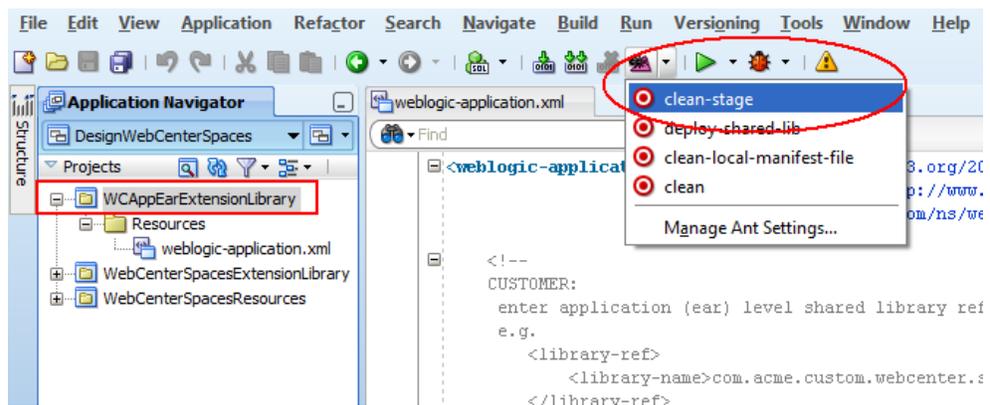
   iii) To deploy the customized Spaces shared library list, open the **Run Ant** dropdown menu, and then select **deploy-shared-lib**.

A new version of the shared library list is deployed to the 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 > custom.webcenter.spaces.fwk >Overview

6. Initiate or schedule a full crawl through Oracle SES administration (that includes the custom profile attributes).

   Refer to "Additional Oracle SES Configuration" in the *Administrator's Guide for Oracle WebCenter Portal* for information. Available from Oracle Technology Network at:

   http://download.oracle.com/docs/cd/E23943_01/webcenter.1111/e12405/wcadm_search.htm#WCADM6569

After a full crawl, custom profile attribute data is returned in Spaces searches.

Sample - Creating Rule-Based Resources

The SampleWebCenterSpacesExtensions workspace includes sample Java code that changes the Spaces skin based on custom logic in UserSkinBean.java, where getSkinForCurrentUser returns the skin to be used for the currently logged in user based on his organization. An 'olive' skin is returned if the organization is 'ADF', 'monochrome' if the organization is 'WebCenter', and 'bluesky' otherwise.

You can write you own rules for returning an appropriate skin based on custom logic of your own or deploy the sample as-is to review the custom code deployment process.

The following steps describe how to deploy the out-of-the-box sample. You can follow the same steps to deploy your own custom code.

1. Open your copy of **SampleWebCenterSpacesExtensions.jws** in JDeveloper.

2. Open the sample custom Java code, located at **RuleBasedResources\Application Sources\custom.webcenter.spaces\view\UserSkinBean.java**.

   Edit the sample skin rules if required.
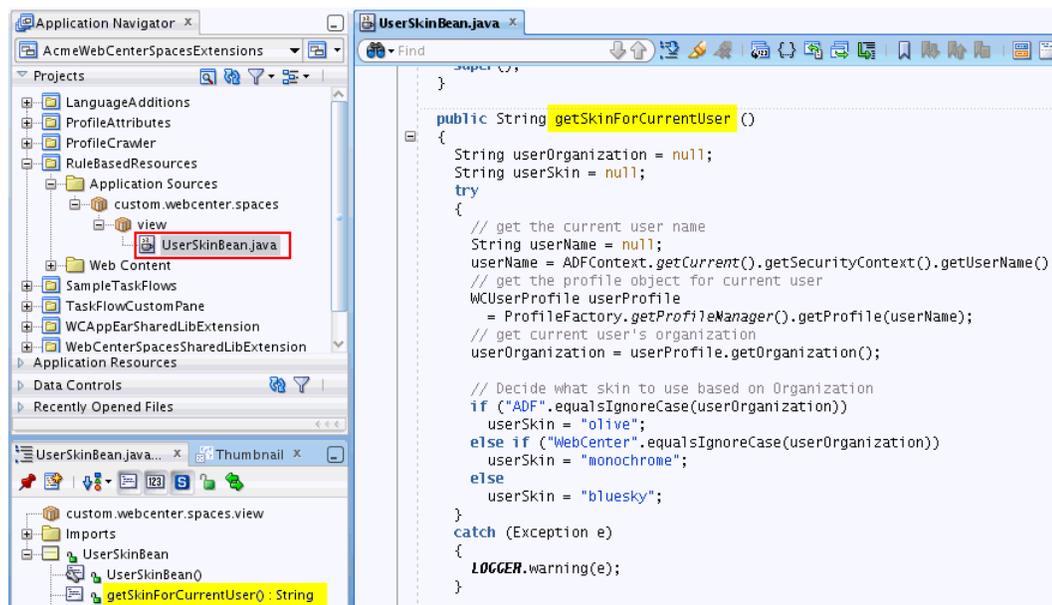
3. If you want to deploy the sample custom code, ensure that the **RuleBasedResources** project is selected in the deployment profile.
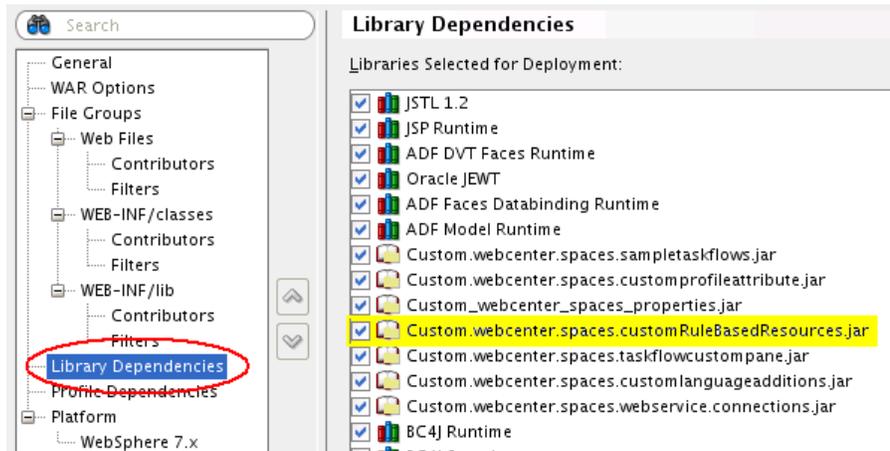
4. Deploy the custom code.

   For details, see section Building and Deploying a Custom Shared Library for the Spaces Application..

5. Log in to the Spaces application as an Administrator to test the custom code.

   For example:

   a) Open Spaces Administration, and display the **Configuration> General** tab.

   b) For **Application Skin**, click the Expression Builder icon and enter the EL expression: #{userSkinBean.skinForCurrentUser}

   c) Open your user profile and change the **Organization** property to 'ADF'.

   d) Save the changes.

   e) Navigate to a different page to verify that the 'olive' skin is applied.

Sample - Deploying Additional Task Flows

Out-of-the box, Spaces provides a range of collaborative, social networking, and personal productivity task flows. Spaces users can access built-in task flows through the Resource Catalog.

If you create your own task flows in JDeveloper, you can deploy them to Spaces and access them through the Resource Catalog alongside the default task flows.

A task flow is a self-contained, reusable ADF artifact. Task flows typically include a task flow definition file and a JSP fragment. Task flows that offer a UI built using data controls also include additional data control files, a page definition file, and a data binding file. Refer to the *Fusion Developer's Guide for Oracle Application Development Framework* for more information about packaging and deploying task flows.

> 📖 *Fusion Developer's Guide for Oracle Application Development Framework* is available from Oracle Technology Network at:
> http://download.oracle.com/docs/cd/E23943_01/web.1111/b31974/partpage3.htm

Out-of-the- Box Task Flows

Many built-in task flows are available with Spaces. To learn how to customize any of the out-of-the-box task flows, read "Customizing Task Flows" in *Developer's Guide for Oracle WebCenter Portal*.

> 📖 *Developer's Guide for Oracle WebCenter* Portal *(11.1.1.6.0)* is available from Oracle Technology Network at:
> http://download.oracle.com/docs/cd/E23943_01/webcenter.1111/e10148/jpsdg_wcsres.htm

Sample Task Flow

A sample custom task flow is provided in the **SampleWebCenterSpacesExtensions** workspace:

TASK FLOW	DESCRIPTION	SAMPLE
List of Pages	Displays a list of pages with context menus	SampleWebCenterSpacesExtensions\SampleTaskFlows\Web Content\ListOfPages.jsff

In addition, the workspace contains the `PageTitle.java` bean (located in the SampleWebCenterSpacesExtensions\SampleTaskFlows\src\bean folder), which displays a page title on the ListOfPages task flow. The following sample implementation of `PageTitle.java` shows how a bean can be accessed as a `pageFlowScope` bean in a task flow:

```
<?xml version="1.0" encoding="US-ASCII" ?>
<adfc-config xmlns="http://xmlns.oracle.com/adf/controller"
version="1.2">
  <task-flow-definition id="ListOfPages">
    <default-activity>ListOfPages</default-activity>
    <managed-bean id="__4">
      <managed-bean-name id="__3">pageTitle</managed-bean-name>
      <managed-bean-class id="__1">bean.PageTitle</managed-bean-
class>
      <managed-bean-scope id="__2">pageFlow</managed-bean-scope>
    </managed-bean>
    <view id="ListOfPages">
      <page>/ListOfPages.jsff</page>
    </view>
    <use-page-fragments/>
  </task-flow-definition>
</adfc-config>
```

The `ListOfPages.jsff` fragment has the following code, showing how the value is retrieved from the bean:

```
...
<af:panelGroupLayout id="stevepgl1" layout="vertical"
inlineStyle="background-color:transparent">
  <style>
    .xgd {background-image:none;}
  </style>
  <af:outputText value="#{pageFlowScope.pageTitle.pageTitle}"
id="ot1"/>
….
```

Packaging and Deploying the Sample Task Flow

Additional task flows must be packaged in an *ADF Library JAR* and deployed in a Spaces shared library. The following steps describe how to deploy a sample task flow using the SampleWebCenterSpacesExtensions workspace.

Note: You must create a new project if you want to deploy task flows of your own. For details, refer to the section <u>Adding New Projects to the Sample Workspace</u>.

To deploy the sample task flow:

1. Open your copy of **SampleWebCenterSpacesExtensions.jws** in JDeveloper.

   Source files for the List of Pages task flow are located under the **SampleTaskFlows** project.
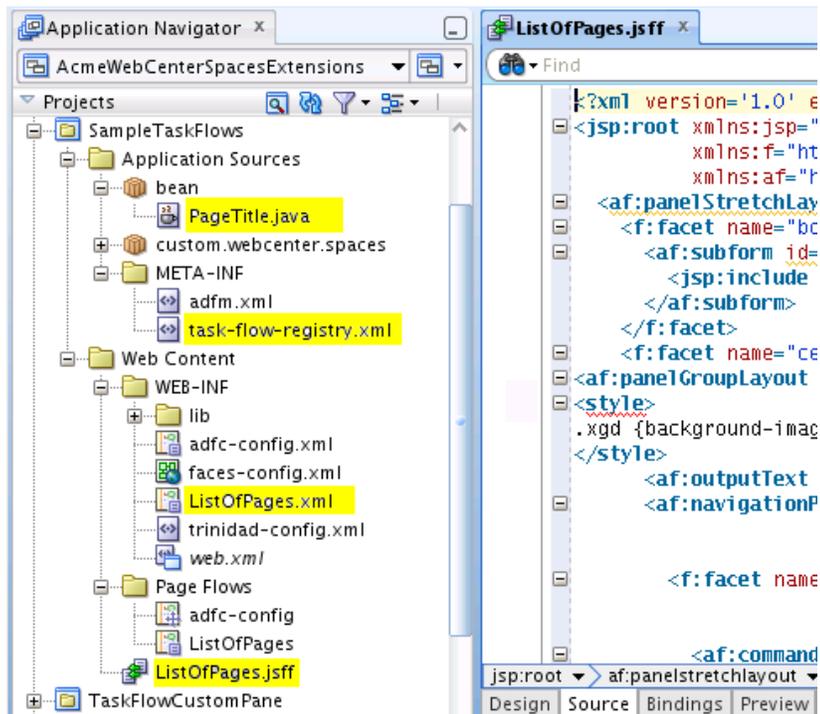
2. If you want to deploy the sample task flow, ensure that the **SampleTaskFlows** project JAR is selected in the deployment profile.
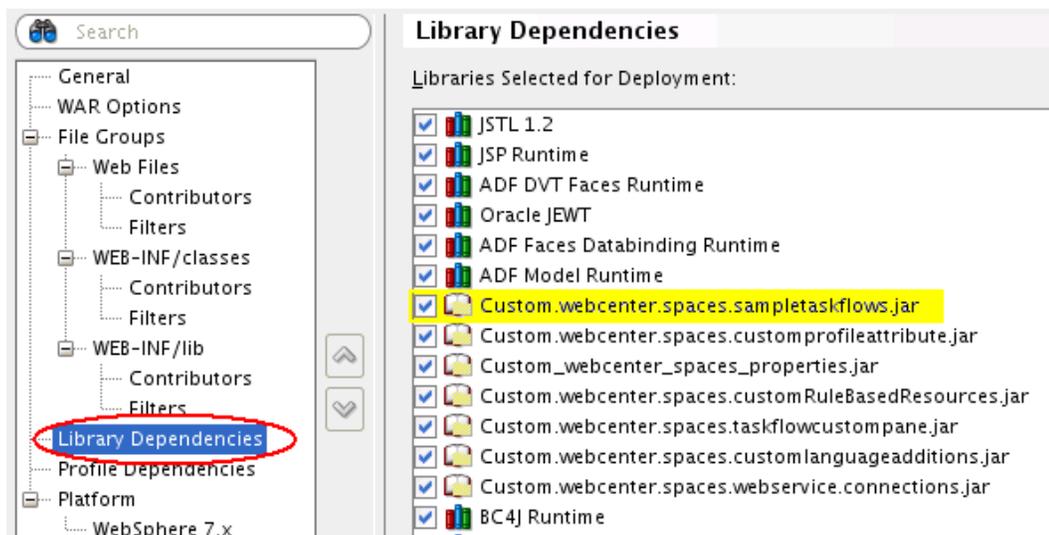
3. Deploy the sample page list task flow.

   For details, see section *Building and Deploying a Custom Spaces Shared Library*.

4. Log in to the Spaces application and verify that the new sample task flow is available in the resource library.

   Follow these steps to add the sample task flow to an existing Resource Catalog:

   a) In Resource Manager, select **Resource Catalog** in the left panel.

      - To add the task flow to an application-level Resource Catalog, navigate to the WebCenter Portal: Spaces Administration Resource tab.

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

      - To add the task flow to a space-specific Resource Catalog, navigate to the Resource tab for the space (under Space Settings).

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

   b) (Optional) Create a new Resource Catalog in which to display the new task flow, if required.

      Alternatively, add the task flow to an existing catalog. If you are not familiar with editing or creating Resource Catalogs, read "Working with Resource Catalogs" in *User's Guide for Oracle WebCenter Portal: Spaces.*

      > 📖 *User's Guide for Oracle WebCenter Portal: Spaces* is available from Oracle Technology Network at:
      > http://download.oracle.com/docs/cd/E23943_01/webcenter.1111/e10149/catalogs.htm

   c) Select the catalog in which you want to expose the new task flow, and then choose **Edit** from the Edit menu.

      You must have **Resource Catalogs-Create Edit Delete** permissions to do this.

   d) Select **Add From Library**.

   e) To find the sample task flow, enter "**ListOfPages**" in the Search field.

   f) Select **ListOfPages**, and click **Add**.

   g) Associate the Resource Catalog with pages/page templates in the Home space or a specific space. Do one of the following:

      - To associate the Resource Catalog with pages or page templates in the **Home space**, navigate to Spaces Administration (General Configuration tab):

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

Configure the Resource Catalog. Use one of the following properties:

- o Resource Catalog for Home Space
- o Resource Catalog for Business Role Pages
- o Resource Catalog for Page Templates in Home Space

- To associate the Resource Catalog with pages or page templates in a **specific space**, navigate to the General settings tab for the space (under Space Settings):

  `http://`*`host:port`*`/webcenter/spaces/`*`spaceName`*`/admin/general`

  Configure the Resource Catalog. Use one of the following properties:

  - o Resource Catalog for Pages
  - o Resource Catalog for Page Templates

Follow these steps to add the sample task flow to a page (or page template) that is configured to use the updated Resource Catalog:

h) Open the target page in edit mode.

i) Add the sample task flow to the page.
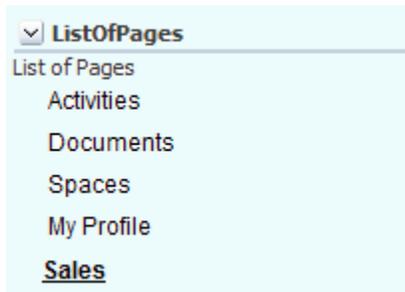
See also, "Adding Resource Catalog Component to a Page" in *User's Guide for Oracle WebCenter Portal: Spaces.*

📖 *User's Guide for Oracle WebCenter Portal: Spaces* is available from Oracle Technology Network at:
http://download.oracle.com/docs/cd/E23943_01/webcenter.1111/e10149/pages_build.htm

Sample – Creating a Custom Property Panel for Task Flows

This section describes the sample project *TaskFlowCustomPane*. This project customizes the property pane that displays for a specific task flow (in Composer's Component Properties dialog). The sample property pane contains a single input text box. When you edit the task flow in Composer, you can enter a value in the text box. The value entered in the text box is persisted so when you re-launch the Component Properties dialog, the specified value displays.

To add a custom property pane:

1. Open your copy of `SampleWebCenterSpacesExtensions.jws` in JDeveloper.

2. Examine code samples available under the **TaskFlowCustomPane** project:

   - `cust_pane.jsff`

   - `taskflow.jsff`

   - `pe_ext.xml`

   - `CustPaneMBean.java`

   - `EventListener.java`

   The samples show how the new property pane is implemented. You can implement a custom property pane using similar code:

   - The `cust_pane.jsff` task flow is the custom property pane with a simple `inputText` component:

   ```
   <?xml version='1.0' encoding='UTF-8'?>
   <jsp:root xmlns:jsp="http://java.sun.com/JSP/Page" version="2.1"
                   xmlns:af="http://xmlns.oracle.com/adf/faces/rich">

     <af:inputText label="Parameter" id="it1"
                   value="#{pageFlowScope.custPaneMBean.uiParam}"/>
   </jsp:root>
   ```

   - The `taskflow.jsff` file is the task flow to which the custom property pane is applied. This task flow displays the value entered in the custom property pane:

   ```
   <?xml version='1.0' encoding='UTF-8'?>
    <jsp:root xmlns:jsp="http://java.sun.com/JSP/Page" version="2.1"
              xmlns:af="http://xmlns.oracle.com/adf/faces/rich">

     <af:outputText value="Task Flow
   Parameter:{pageFlowScope.parameter}"
                       id="ot1"/>
     </jsp:root>
   ```

   - `pe_ext.xml` is the Composer extension file. To enable the custom property pane to display in Composer, the pane is registered in the `pe_ext.xml` file. The

custom panel is defined as `sample.cust.pane`, the custom pane is registered against the `taskflow.jsff` (through `Sample-TaskflowCustomPanel.xml`), and all the default properties set to `false`.

```xml
<?xml version="1.0" encoding="US-ASCII" ?>
  <pe-extension
xmlns="http://xmlns.oracle.com/adf/pageeditor/extension">

   <addon-config>
     <panels>
      <panel name="sample.cust.pane" title="Custom Property Panel"
         taskflow-id="/WEB-INF/cust-pane-defn.xml#cust-pane-defn">
       <event-handlers>
         <event-handler
event="save">tfcustompane.EventListener</event-handler>
       </event-handlers>
      </panel>
     </panels>

    <property-panels>
     <property-panel name="sample.prop.pane">
       <taskflow-id>/WEB-INF/Sample-
TaskflowCustomPanel.xml#Sample-TaskflowCustomPanel</taskflow-id>
        <panel name="sample.cust.pane" index="first"/>
        <panel name="oracle.adf.pageeditor.pane.events"
rendered="false"/>
        <panel name="oracle.adf.pageeditor.pane.child-navigator"
                   rendered="false"/>
        <panel name="oracle.adf.pageeditor.pane.generic-property-
inspector"
                   rendered="false"/>
      <panel name="oracle.adf.pageeditor.pane.region-param"
rendered="false"/>
        </property-panel>
     </property-panels>
  </addon-config>
</pe-extension>
```
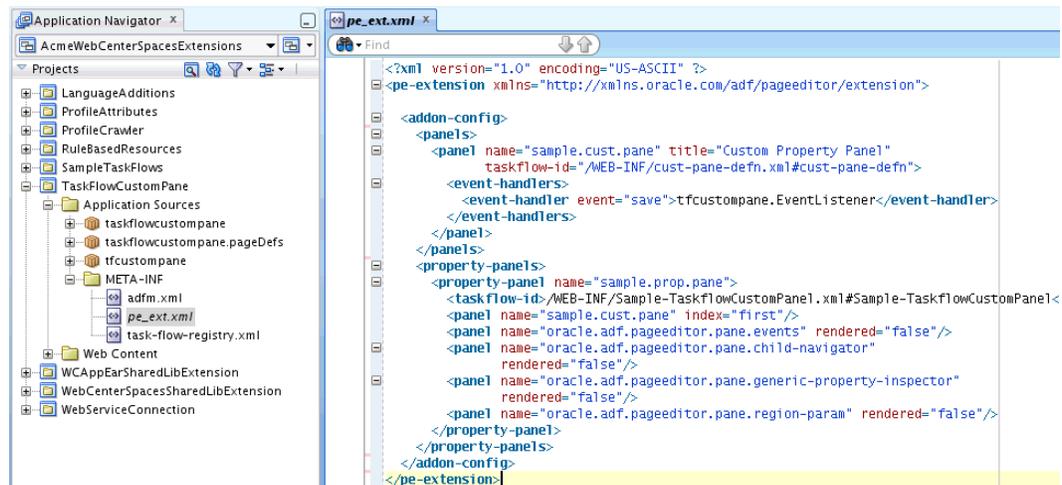
- The following Java classes are part of the custom pane implementation:

    - `CustPaneMBean.java` – Contains the logic to display the parameter value from `cust_pane.jsff` as output text in `taskflow.jsff`.

    - `EventListener.java` – Contains the logic to save the parameter value when you specify the parameter value and click OK or Apply in the Component Properties dialog.

3. If you want to deploy the custom property pane, ensure that the **TaskFlowCustomPane** project JAR is selected in the deployment profile:
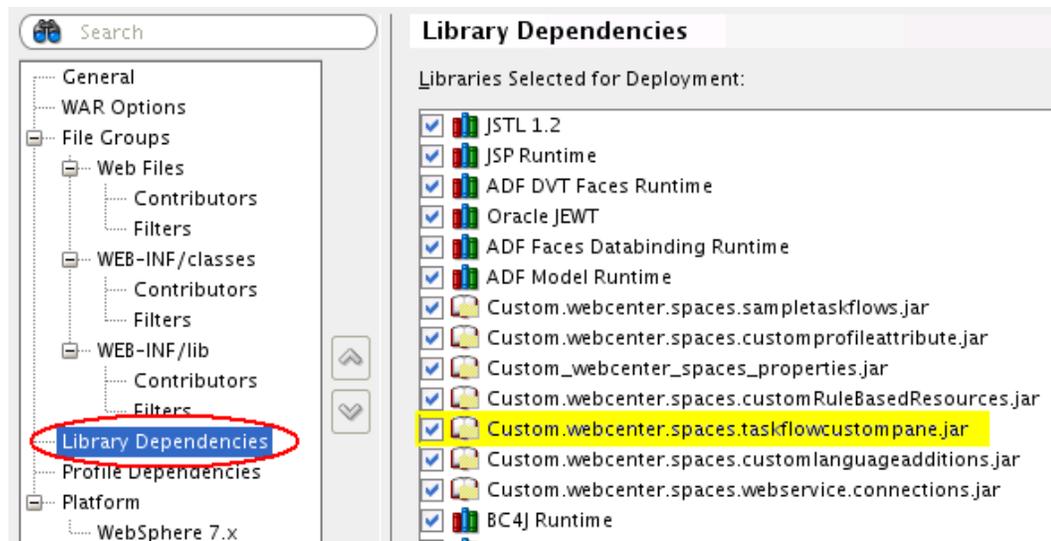
4. Deploy the project with the sample custom pane.
   For details, see section Building and Deploying a Custom Spaces Shared Library.

5. Log in to the Spaces application and verify that the new sample task flow is available in the resource library.
   Follow these steps to add the sample task flow to an existing Resource Catalog:

   a) In the Resource Manager, select **Resource Catalog** in the left panel.

      - To add the task flow to an application-level Resource Catalog, navigate to the WebCenter Portal: Spaces Administration Resource tab:
        `http://host:port/webcenter/spaces/admin/resources`

      - To add the task flow to a space-specific Resource Catalog, navigate to the Resource tab for the space (under Space Settings):
        `http://host:port/webcenter/spaces/spaceName/admin/resources`

b)   (Optional) Create a new Resource Catalog in which to display the new task flow, if required.

Alternatively, add the task flow to an existing catalog. If you are not familiar with editing or creating Resource Catalogs, read "Working with Resource Catalogs" in User's Guide for Oracle WebCenter Portal: Spaces

📖 *User's Guide for Oracle WebCenter Portal: Spaces* is available from Oracle Technology Network: http://download.oracle.com/docs/cd/E23943_01/webcenter.1111/e10149/catalogs.htm

c)   Select the Resource Catalog in which you want to expose the new task flow, and then choose **Edit** from the Edit menu.
You must have **Resource Catalogs-Create Edit Delete** permissions to do this.

d)   Select **Add From Library**.

e)   To find the sample task flow, enter "**TaskflowCustomPanel**" in the Search field.

f)   Select **Sample-TaskflowCustomPanel**, and click **Add**.

g)   Associate the Resource Catalog with pages or page templates in the Home space or a specific space. To do this, perform either of the following steps:

- To associate the catalog with pages or page templates in the **Home space**, navigate to Spaces Administration (General Configuration tab):

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

  Configure the catalog by selecting one of the following:

  - Resource Catalog for Home Space
  - Resource Catalog for Business Role Pages
  - Resource Catalog for Page Templates in Home Space

- To associate the catalog with pages or page templates in a **specific space**, navigate to the General settings tab for the space (under Space Settings):

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

  Configure the catalog by selecting one of the following:

  - Resource Catalog for Pages
  - Resource Catalog for Page Templates

Follow these steps to add the sample task flow to a page (or page template) that is configured to use the updated Resource Catalog:

h)   Open the target page in edit mode.

i)   Add the sample task flow **Sample-TaskflowCustomPanel** to the page.

See also, "Adding Resource Catalog Component to a Page" in User's Guide for Oracle WebCenter Portal: Spaces.

📖 *User's Guide for Oracle WebCenter Portal: Spaces* is available from Oracle Technology Network:

http://download.oracle.com/docs/cd/E23943_01/webcenter.1111/e10149/pages_build.htm

j)   Click the **Edit** icon to display the Component Properties dialog.
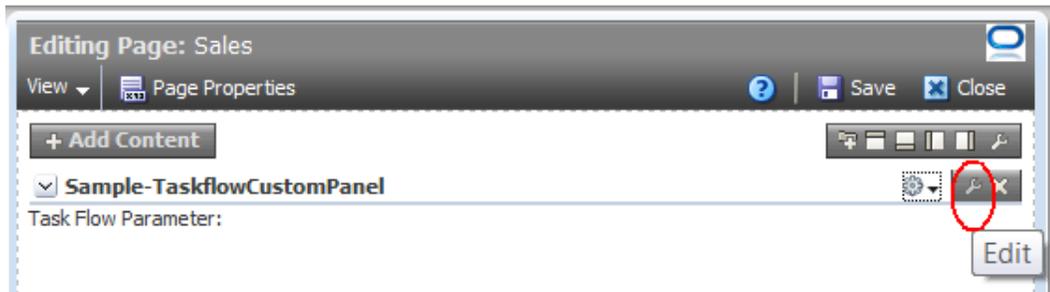

k)   Enter a value in the input text box and click **OK**.

l)   Click **Edit** again to verify that the value you entered was saved.

m) Click **Save** and close Composer.

Notice that the value entered is displayed on the page. For example, as Task Flow Parameter: 444444.

Sample – Invoking an External Public Web Service

The SampleWebCenterSpacesExtensions workspace contains a Web Service project that demonstrates how to include a Web Service-based task flow in the Spaces application. The sample describes the external public Web Service GeoIPService, `http://www.webservicex.net/geoipservice.asmx`, which accepts an IP address as an input parameter and returns the country code.
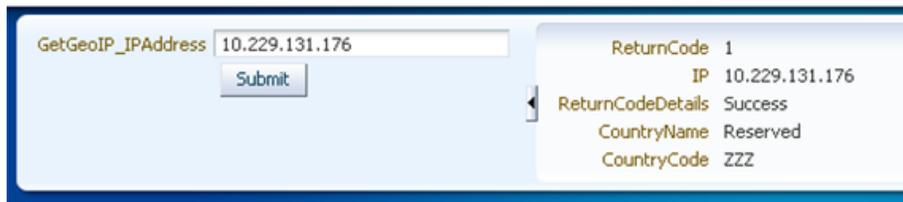
The WebServiceConnection project includes:

- `GeoIPServiceDC` - Web Service data control for GeoIPService

- `geoipfrag.jsff` – page containing the Web Service data control

- `Sample-GeoIPService` – task flow based on `geoipfrag.jsff`

- `connections.xml` – connection details for the GeoIPService Web Service. Available at `<UnzipDir>/SampleWebCenterSpacesExtensions/.adf/META-INF/connections.xml`. The Spaces application does not use this file directly because Web Service connections for the Spaces application are specified through Enterprise Manager Fusion Middleware Control. Later on, when you set up the Web Service connection, you will be asked to copy WSDL information from this file.

To deploy a sample Web Service task flow and configure the Web Service connection:

1. Open your copy of `SampleWebCenterSpacesExtensions.jws` in JDeveloper.

2. Expand the **WebServiceConnection** project and review the various files available:
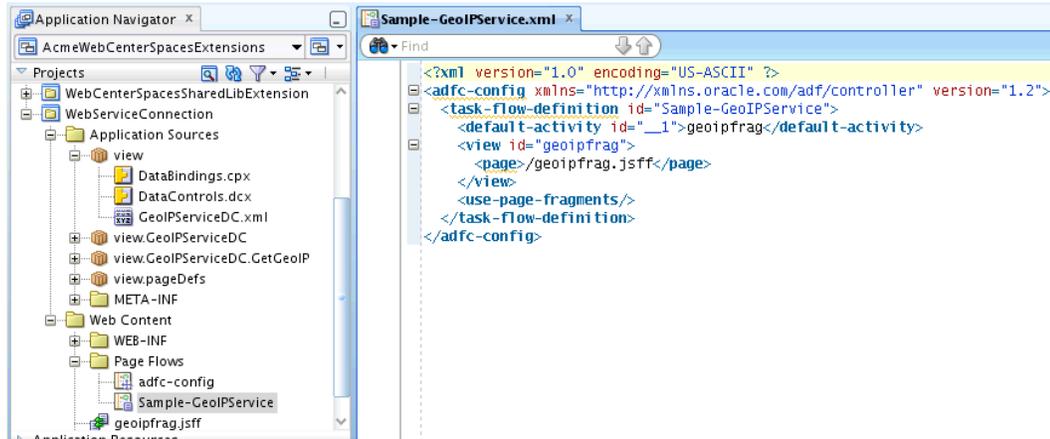
3. Define a Web Service connection for the GeoIPService using Enterprise Manager Fusion Middleware Control:

a) Log in to Fusion Middleware Control.

```
http://host:port/em
```

b) Navigate to **WebLogic Domain >wc_domain >WC_Spaces**, the managed server on which the Spaces application is deployed.

c) Select **System MBean Browser** from the **WebLogic Server** menu.
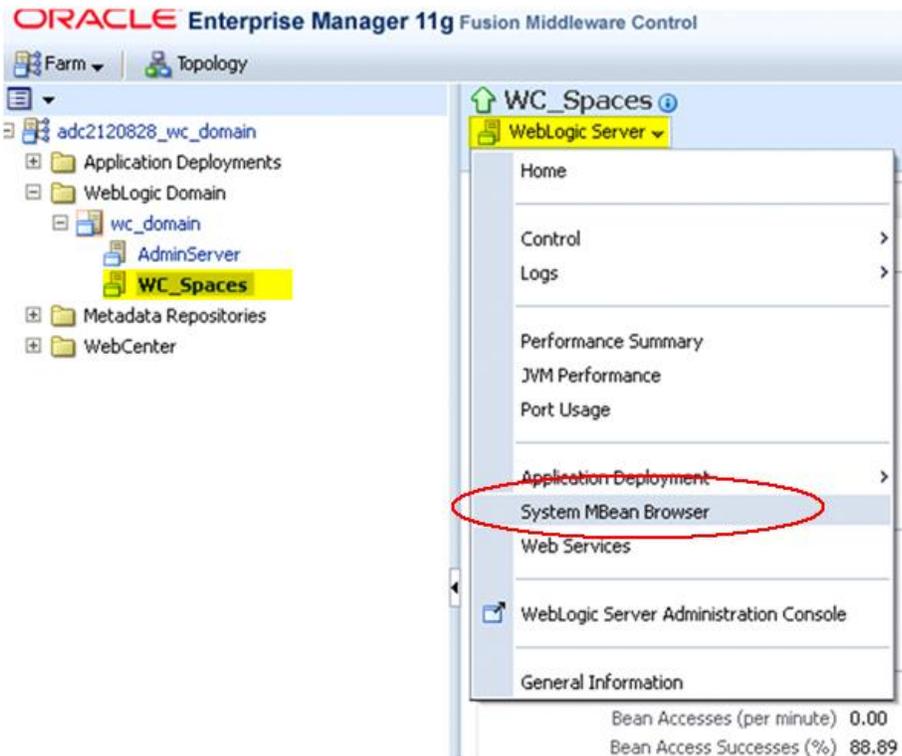
d) In the tree, drill down to locate **oracle.adf.share.connections** and
    ADFConnections.
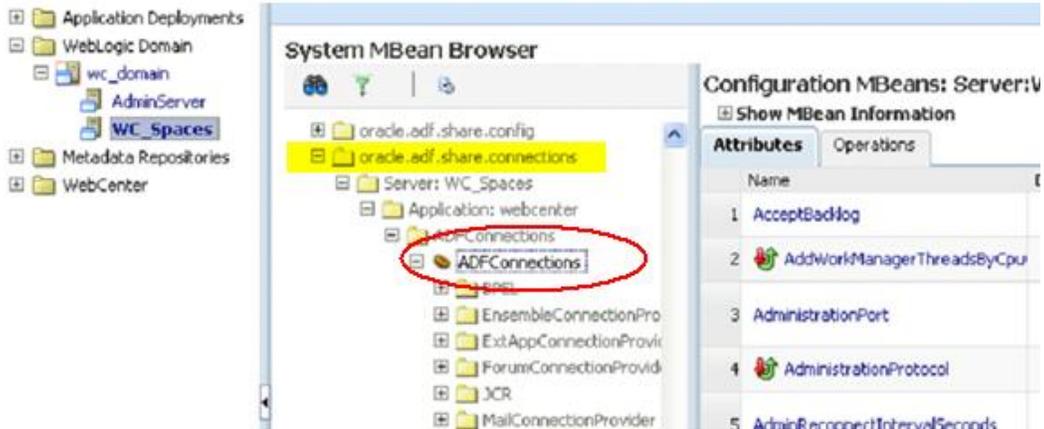
e) Select **ADFConnections**, **Operations**, and then **createConnection**.
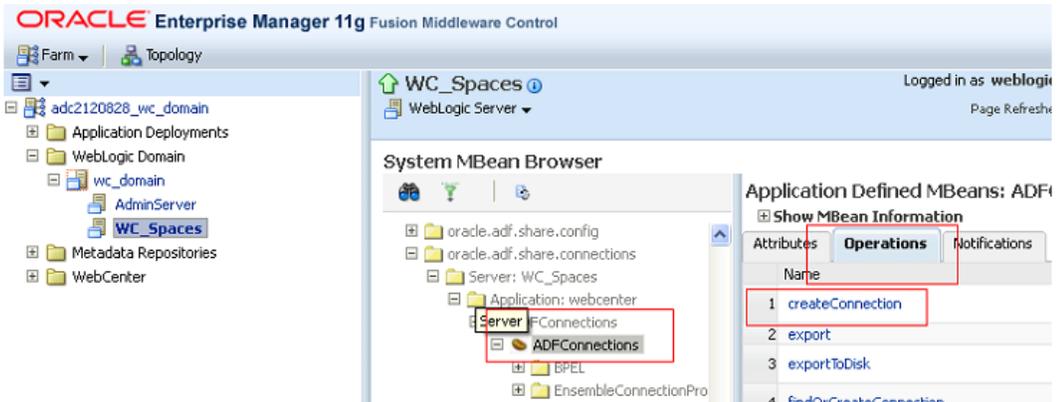
f) For Connection Type, enter **WebServiceConnection**.

g) For Connection Name, enter **GeoIPServiceDC**.

> **Note:** GeoIPServiceDC is the reference name specified in
> `SampleWebCenterSpacesExtension/.adf/META-INF/connections.xml`.

h) Click **Invoke** to save the connection.

i) Select **ADFConnections > Operations > Save**, and then click **Invoke** to save the connections.

j)   Refresh the System MBean Browser tree to display the new connection.

k)   Select **GeoIPServiceDC**.
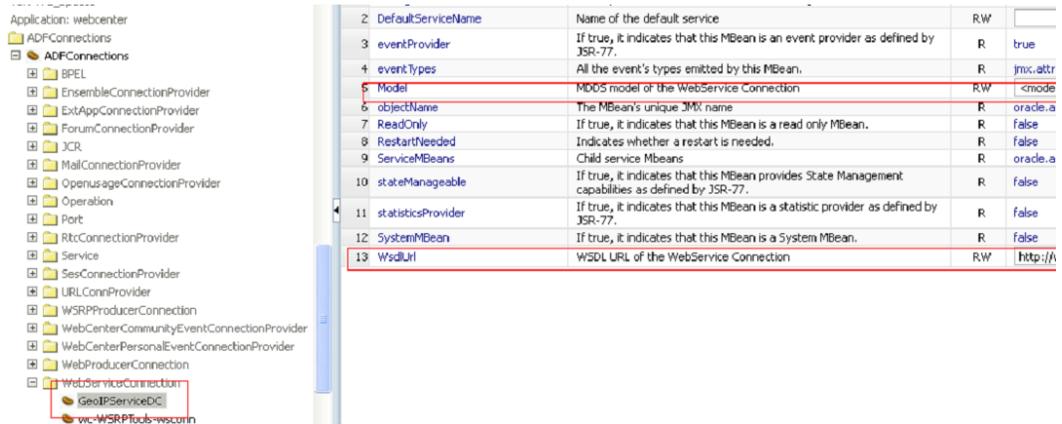
l)   Enter values for **Model** and **wsdlUrl**:
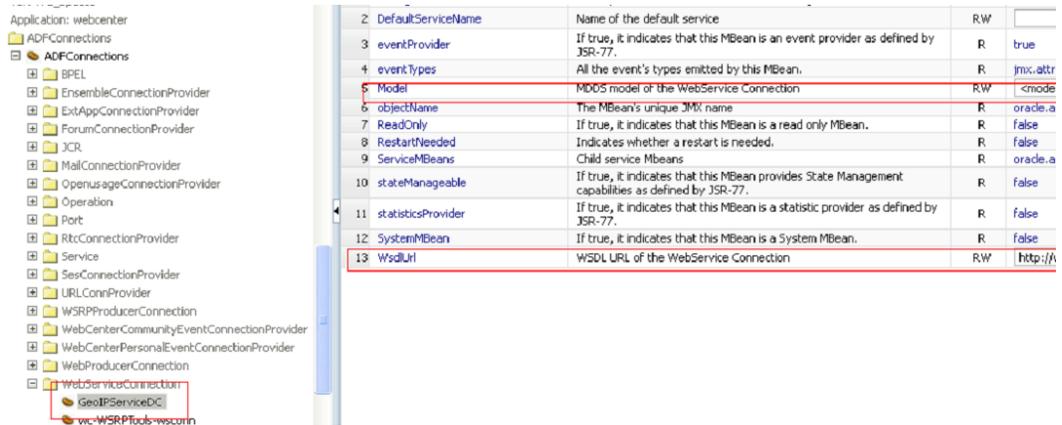
Model –

i. Open `SampleWebCenterSpacesExtension/.adf/META-INF/connections.xml` in JDeveloper.

ii. Navigate to `GeoIPServiceDC`.

iii. Copy the entire `<model>…</model>` attribute under this section.

iv. Paste the `<model>…</model>` content in the **Model** field.

```
<model name="{http://www.webservicex.net/}wsdl"
              xmlns="http://oracle.com/ws/model">
  <types xmlns="http://oracle.com/oracle/webservices/mdds">
    <complexType elementQualified="true"
               namespace="http://www.webservicex.net/"
               localPart="GeoIP"
               prefix="tns" xmlns="">
      <sequence minOccurs="1" maxOccurs="1">
      <part name="ReturnCode" required="true"
            namespace="http://www.webservicex.net/"
            elementQualified="true">
              ….....
      </model>
```

v. Click **Apply** to save.

WSDL URL -

Enter: http://www.webservicex.net/geoipservice.asmx?WSDL

m) Select **ADFConnections > Operations > Save** > **Invoke**.

n) Ensure that the appropriate proxy settings are configured in `setDomainEnv` for Spaces managed servers. The **GeoIPServiceDC** sample is an external Web Service and may require some proxies to work. For example, add the following to `setDomainEnv`:

`-Dhttp.proxyHost="www-proxy.mycompany.com" -Dhttp.proxyPort="80"`

o) Restart the WC_Spaces managed server.

4. In JDeveloper, ensure that the WebServiceConnection project JAR is selected in the deployment profile.
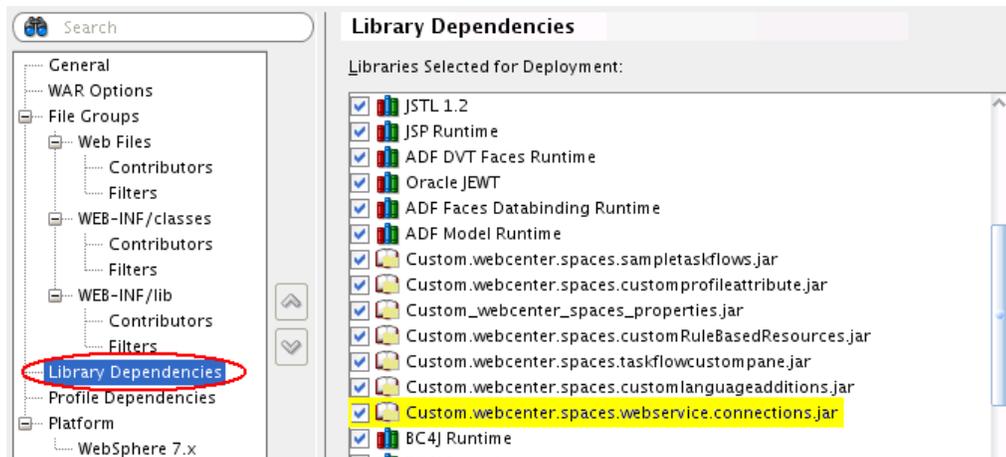
5. Deploy the sample Web Service connection project.
   For details, see section [Building and Deploying a Custom Spaces Shared Library](#).

6. Validate that the external Web Service is up and running before testing the task flow in the Spaces application:

   a) Enter the following URL in your browser:
      `http://www.webservicex.net/geoipservice.asmx`

   b) Click the **GetGeoIP** link.

   c) Enter an IP address, for example: `10.229.131.176`

   d) Click **Invoke**.

   If the test returns a country code, the Web Service is up and running and working properly.

7. Log in to the Spaces application and verify that the **Sample-GeoIPService** task flow is available in the resource library.
   Perform the following steps to add the sample task flow to edit an existing Resource Catalog:

   a) In Resource Manager, select Resource Catalog in the left panel.

   - To add the task flow to an application-level Resource Catalog, navigate to the WebCenter Portal: Spaces Administration Resource tab:
     [http://*host:port*/webcenter/spaces/admin/resources](#)

   - To add the task flow to a space-specific Resource Catalog, navigate to the Resource tab for the space (under Space Settings):
     [http://*host:port*/webcenter/spaces/spaceName/admin/resources](#)

b) (Optional) Create a new Resource Catalog in which to display the new task flow, if required.
Alternatively, add the task flow to an existing catalog. If you are not familiar with editing or creating Resource Catalogs, read "Working with Resource Catalogs" in *User's Guide for Oracle WebCenter Portal: Spaces*.

> 📖 *User's Guide for Oracle WebCenter Portal: Spaces* is available from Oracle Technology Network:
>
> http://download.oracle.com/docs/cd/E23943_01/webcenter.1111/e10149/catalogs.htm

c) Select the catalog in which you want to expose the new task flow, and then choose **Edit** from the Edit menu.
You must have **Resource Catalogs-Create Edit Delete** permissions to do this.

d) Select **Add From Library**.

e) To find the sample task flow, enter "**GeoIPService**" in the Search field.

f) Select "**Sample-GeoIPService**", and click **Add**.

g) Associate the Resource Catalog with pages/page templates in the Home space or a specific space. Do one of the following.

- To associate the catalog with pages or page templates in the **Home space**, navigate to Spaces Administration (General Configuration tab):

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

  Configure the catalog by selecting one of the following:

  - Resource Catalog for Home Space
  - Resource Catalog for Business Role Pages
  - Resource Catalog for Page Templates in Home Space

- To associate the catalog with pages or page templates in a **specific space**, navigate to the General settings tab for the space (under Space Settings):

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

  Configure the catalog by selecting one of the following:

  - Resource Catalog for Pages
  - Resource Catalog for Page Templates

Follow these steps to add the sample task flow to a page (or page template) that is configured to use the updated Resource Catalog:

h) Open the target page in edit mode.

i) Add the sample task flow **Sample-GeoIPService** to the page.
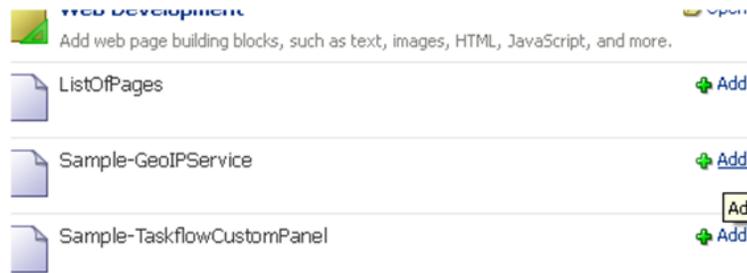
See also, "Adding Resource Catalog Component to a Page" in *User's Guide for Oracle WebCenter Portal: Spaces.*

📖 *User's Guide for Oracle WebCenter Portal: Spaces* is available from Oracle Technology Network:

http://download.oracle.com/docs/cd/E23943_01/webcenter.1111/e10149/pages_build.htm

j) In page view mode, enter a valid IP Address and click **Submit**.

Note that the task flow displays the country code for the specified IP address.
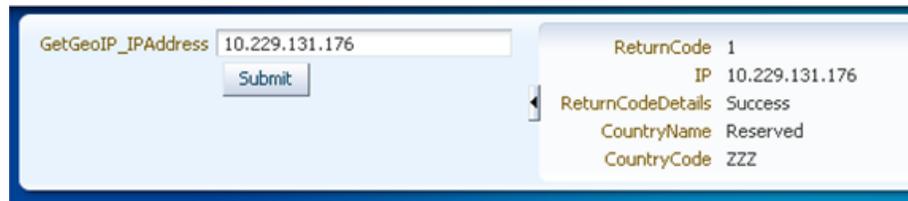
Sample – Customizing the Login Task Flow for SSO

Out-of-the-box, Spaces uses container-based authentication. Additional configuration is required to enable a Single Sign-On (SSO) solution in which a specific SSO login page displays to unauthenticated users who click " *Login*" in the Spaces Login task flow or attempt to access a protected URL.

For the authentication process to work:

- **The Login task flow must be on a JSPX page**.

  For details, see *Step 1 Ensure the Spaces Application Login Page is JPSX.*

- **Oracle Access Manager (OAM) and Web Gate must be configured** on the WebLogic Server instance running Spaces and Spaces must be configured to use single sign-on with OAM.

  Spaces supports the following OAM and WebGate versions:

  - Oracle Access Manager (OAM) 11g with WebGate 10g,   or
    (with Oracle Internet Directory 11g Release 11.1.1.6.0)

  - Oracle Access Manager (OAM) 10g with WebGate 10g
    (with Oracle Internet Directory 10g Release 10.1.4.3.0)

  For details, *Step 2 Configure Oracle Access Manager (OAM) SSO.*

- **Users must access the Spaces application using the SSO URL** (that is, Spaces must be front-ended with Oracle HTTP Server configured for SSO).

Step 1 Ensure That the Spaces Application Login Page is JSPX

The Login task flow must be on a JSPX page for the sample OAM login code to work. Out-of-the-box, the Spaces Welcome page that displays to unauthenticated users is a JSP page (`LandingGateway.jsp`).

To display a JSPX version of the Spaces Welcome page, customize the page as follows:

- On the Spaces Administration page, select **Pages > System Pages > WebCenter Portal: Spaces Welcome Page > Customize > Save**

Note: You do not have to do this step if you have disabled public access to Spaces. In this case, a Login JSPX page displays instead of the Welcome page. To disable public access to Spaces, in Spaces Administration page, select **Security > Roles > View Application > Public User > Disable**.

Step 2 Configure Oracle Access Manager (OAM) SSO

The sample OAM login code requires Oracle Access Manager 10g (with WebGate 10g) or Oracle Access Manager 11g (with WebGate 10g) and the Spaces application must be configured to use single sign-on with OAM. If you want to use the sample, ensure that OAM is installed on the WebLogic Server where Spaces is running.

This whitepaper describes how to do this for **OAM 10g** and includes the following subsections:

> Step 2A: Setting Up OAM SSO

> Step 2B: Installing OAM SDK

> Step 2C: Configuring ObAccessClient.xml for OAM SDK

> Step 2D: Setting WebLogic Server Environment Variables

> Step 2E: Accessing OAM SDK Initialization and Shutdown

Additional Information

The *Administrator's Guide for Oracle WebCenter Portal* describes in detail how to configure SSO for Spaces and related applications such as SOA, WebCenter Content, using an out-of-the-box SSO configuration file. For detailed instructions, see:

> o [Installing and Configuring OAM 11g](#)

> o [Installing and Configuring OAM 10g](#)

📖 *Administrator's Guide for Oracle WebCenter Portal* is available from Oracle Technology Network:
http://download.oracle.com/docs/cd/E23943_01/webcenter.1111/e12405/wcadm_security_sso.htm

Step 2A: Setting Up OAM SSO

Oracle Access Manager (OAM) 10g and 11g configuration is fully documented in the *[Administrator's Guide for Oracle WebCenter Portal](#)*.

In addition, you must configure your installation-specific information in the WebCenter policy domain created. The steps are summarized here (and assume OAM and Oracle Internet Directory (OID) are both installed):

1. Run **OAMConfig** tool to create Access Gate and policy domain entries.

2. Install Oracle HTTP Server (OHS) and configure **mod_weblogic** to front end the required applications. You must add a mapping for your Spaces installation. For example:

```
<Location /webcenter>
      SetHandler weblogic-handler
      WebLogicHost webcenter.example.com
      WebLogicPort 8888
</Location>
```

3. Using OAM Console, turn off **IPValidation** for the Access Gate created in step 1.

    IP address validation is specific to WebGate and is used to determine whether a client's IP address is the same as the IP address stored in the ObSSOCookie generated for single sign-on. The **IPValidation** parameter turns IP address validation on and off. If **IPValidation** is on, the IP address stored in ObSSOCookie must match the client's IP address, otherwise, the cookie is rejected and the user must reauthenticate. By default, **IPValidation** setting is on.

    You need to turn **IPValidation** off for this setup because the ObSSOCookie set by the use of Access SDK cannot get the handle to the client's IP address (`HttpServletRequest.getRemoteAddr()` returns the last proxy that sent the request, in this case OHS). WebGate also must be reconfigured.

    If you want to retain **IPValidation**, you must front-end OHS/WebGate with another proxy/load balancer and configure its IP Address in the IP Validation Exception List. For details, see *Administrator's Guide for Oracle Access Manager*.

    📖 *Administrator's Guide for Oracle Access Manager* is available from Oracle Technology Network:
    http://download.oracle.com/docs/cd/E23943_01/doc.1111/e15478/toc.htm

Configure resources for your Spaces application. For example:
    - `/webcenter/adfAuthentication`
    - `/webcenter` - include in **Default Public Policy**

4. Install WebGate on OHS.

5. Configure `OIDAuthenticator` and `OAMAsserter` on the Weblogic domain.

6. To test the OAM SSO setup, access a secure URL and verify the single sign on experience.

Step 2B: Installing OAM SDK

Install Oracle Access Manager SDK on the same machine as the WebLogic Server running Spaces.

1. Navigate to the Oracle Access Manager SDK Download page on OTN:
    http://www.oracle.com/technetwork/middleware/ias/downloads/101401-099957.html

2. **Accept** the License Agreement.

3. Download the Oracle Access Manager SDK:

   http://download.oracle.com/otn/linux/ias/101401/oam_int_linux_v7_cd3.zip

4. Follow the SDK installation documentation located on OTN:

   http://download.oracle.com/docs/html/E12491_01/sdkinst.htm

Step 2C: Configuring ObAccessClient.xml for OAM SDK

Replace:

```
$OBACCESS_INSTALL_DIR/oblix/lib/ObAccessClient.xml
```

Where `$OBACCESS_INSTALL_DIR` is the directory where you installed OAM SDK in the Step B.

With:

```
$WEBGATE_INSTALL_DIR/access/oblix/lib/ObAccessClient.xml
```

Where `WEBGATE_INSTALL_DIR` is the directory where you installed WebGate when setting up SSO.

Tip: Make a backup before you replace the file.

As the sample enables SSO through the Login task flow *and* through direct access of protected URLs, we need to share the WebGate's configuration file. This step enables OAM SDK logic in the Login task flow to pick up the WebGate's configuration file.

Step 2D: Setting WebLogic Server Environment Variables

Set the following environment variables in `setDomainEnv.sh` or `setDomainEnv.cmd` (located in your *domain*/bin directory where Spaces is installed) and restart the WebLogic Server:

- **OBACCESS_INSTALL_DIR**

```
OBACCESS_INSTALL_DIR="/scratch/myinstall/oam/sdk/AccessServe
rSDK"
export OBACCESS_INSTALL_DIR
```

There is already `POST_CLASSPATH`. Locate that and append the highlighted entries in the following:

```
if [ "${POST_CLASSPATH}" != "" ] ; then

POST_CLASSPATH="${COMMON_COMPONENTS_HOME}/modules/oracle.jrf
_11.1.1/jrf.jar${CLASSPATHSEP}${OBACCESS_INSTALL_DIR}/oblix/
lib/jobaccess.jar${CLASSPATHSEP}${POST_CLASSPATH}"
```

```
            export POST_CLASSPATH

    else

    POST_CLASSPATH="${COMMON_COMPONENTS_HOME}/modules/oracle.jrf
    _11.1.1/jrf.jar${CLASSPATHSEP}${OBACCESS_INSTALL_DIR}/oblix/
    lib/jobaccess.jar"

            export POST_CLASSPATH

    fi
```

- **LD_LIBRARY_PATH**

```
LD_LIBRARY_PATH="${OBACCESS_INSTALL_DIR}/oblix/lib:${WL_HOME
}/server/native/linux/i686${CLASSPATHSEP}${LD_LIBRARY_PATH}"
export LD_LIBRARY_PATH
```

Step 2E: Accessing OAM SDK Initialization and Shutdown

`webcenter-oam.jar` is available as part of the WebCenter Portal installation. Perform the following steps to configure the startup and shutdown classes:

1. Include a reference to `webcenter_oam.jar` in `commEnv.sh`.

    For example, include the following `WEBLOGIC_CLASSPATH` entry in *install location*/wlserver_10.3/common/bin/commEnv.sh:

    ```
    WEBLOGIC_CLASSPATH="${WEBLOGIC_CLASSPATH}${CLASSPATHSEP}${MW_HO
    ME}/Oracle_WC1/webcenter/modules/oracle.webcenter.framework_11.
    1.1/webcenter-oam.jar"
    ```

    ```
    export WEBLOGIC_CLASSPATH
    ```

    Include the new entry after the existing `WEBLOGIC_CLASSPATH` entry.

2. Restart WebLogic Server.

3. In WebLogic Server Administration Console, navigate to **Home > Startup and Shutdown Classes** and configure the following startup and shutdown classes

    - Startup Listener – `OAMConfigStartup`

      The class is packaged in `oracle.webcenter.spaces.oamconfig.OAMConfigStartup`.

    - Shutdown Listener – `OAMConfigShutdown`

      The class is packed in `oracle.webcenter.spaces.oamconfig.OAMConfigShutdown`.

    Ensure that you select **WC_Spaces** as the target server.

Step 3 Validate the Custom OAM Login Code

To verify the custom OAM login code:

1. In your browser, enter the OAM URL:
   `http://OAM_host:OAM_port/webcenter`

   Alternatively, enter the Spaces application URL. For example:

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

2. Enter a valid user name and password, and then click the **Login** button.

   Clicking the Login button authenticates users against OAM SSO.

3. Test that the OAM login code is working as expected. For example, do one of the following:

   - Navigate to any page that contains an RSS news feed:
     `http://OAM_host:OAM_port/rss/rssservlet`

     Verify that you are not prompted for login credentials.

   - Access the REST URL:
     `http://OAM_host:OAM_port/rest/api/resourceIndex`

     Verify that you are not prompted for login credentials.

Adding New Projects to the Sample Workspace

This section describes how to create a new project in the SampleWebCenterSpacesExtensions workspace and deploy your project code to a Spaces shared library:

1. Open your copy of **SampleWebCenterSpacesExtensions.jws** in JDeveloper.

2. Create a new project (.jpr) in the workspace and add your custom code, task flows, and so on.

   For example, **AcmeExtensions.jpr**.

3. Create a deployment profile for the new project

   a) Right-click your project and choose **Project Properties**.

   b) Select **Deployment > New**, and name the profile.

   For example, AcmeExtensions.

   Specify **JAR file** for the **Archive Type**, except when your project contains a task flow in which case you must choose **ADF Library JAR**.
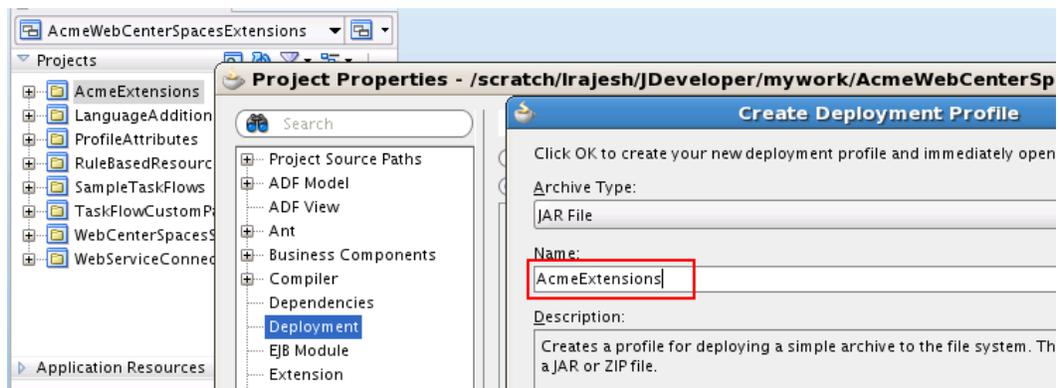

4. Select **JAR Options** to review or change the name of the JAR file generated by the deployment profile:
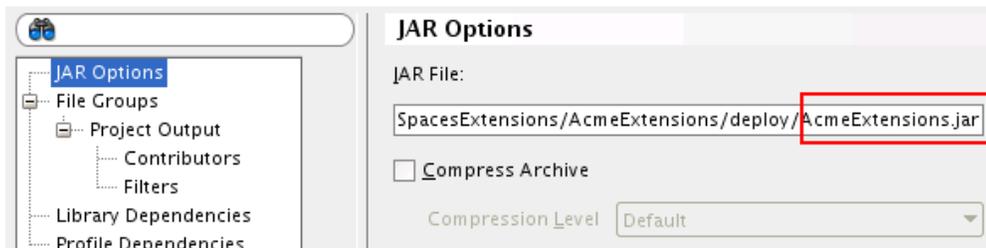
5. Right click your project folder and build the project JAR using the deployment profile.
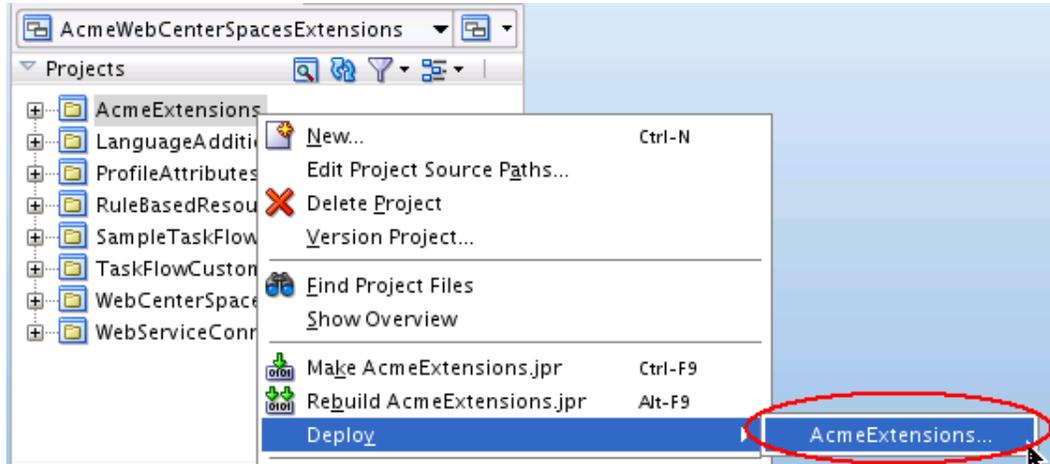

6. Include the project JAR file in the Spaces shared library:

   a) Right-click **WebCenterSharedLibraryExtensionProject > Project Properties**

   b) Select **Libraries & Classpath**.

   c) Select **Add JAR/Directory** to add your project JAR.
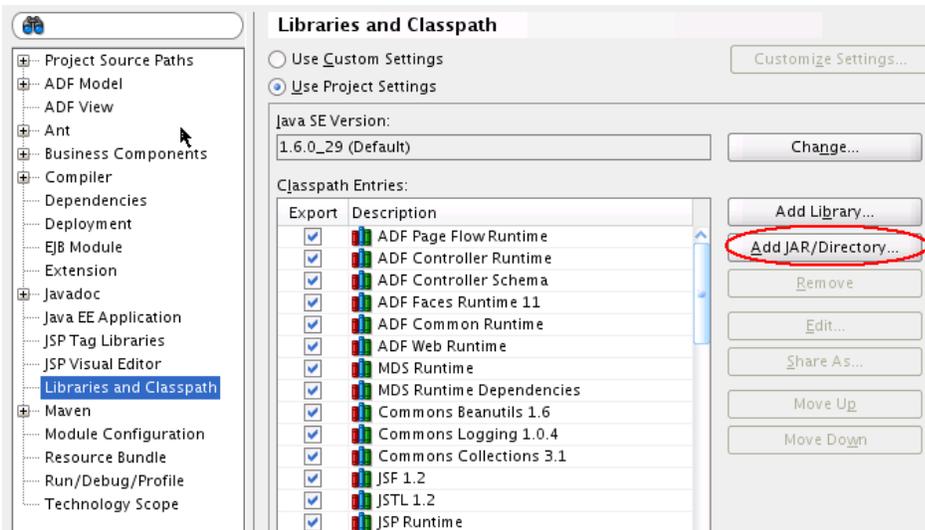
d) Select the JAR file.
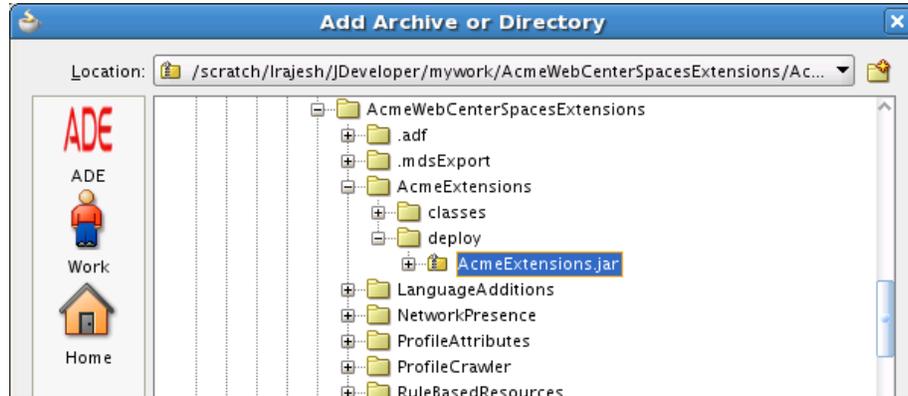
Your project's .JAR is listed:
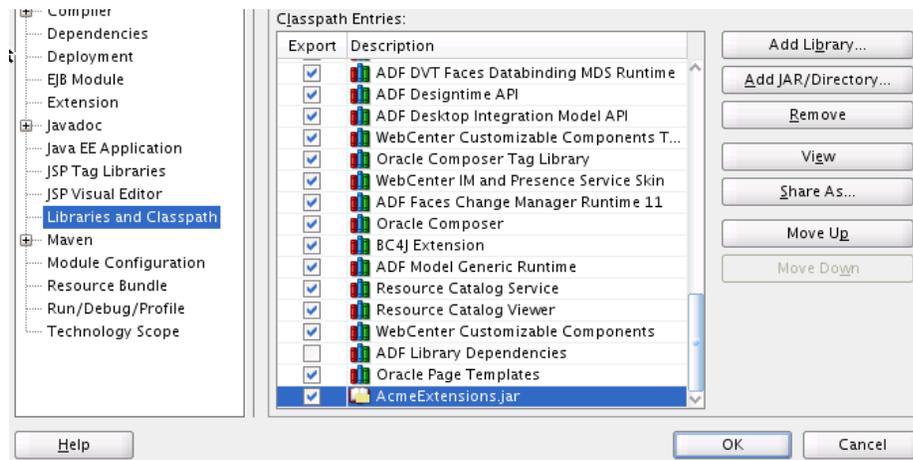
7. Edit the Spaces shared library WAR profile:

a) Select **Deployment > custom_webcenter_spaces_war**, and then click **Edit**.

b) Choose **WEB-INF/lib** > **Filters** and choose the jar generated by the new project.
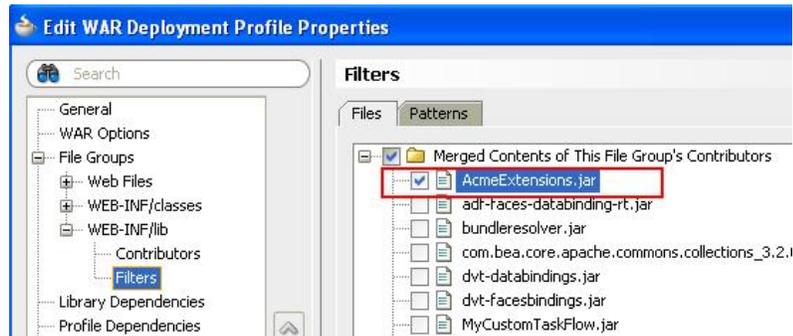
c) Select **Library Dependencies**, and ensure that your project is selected in the deployment profile.
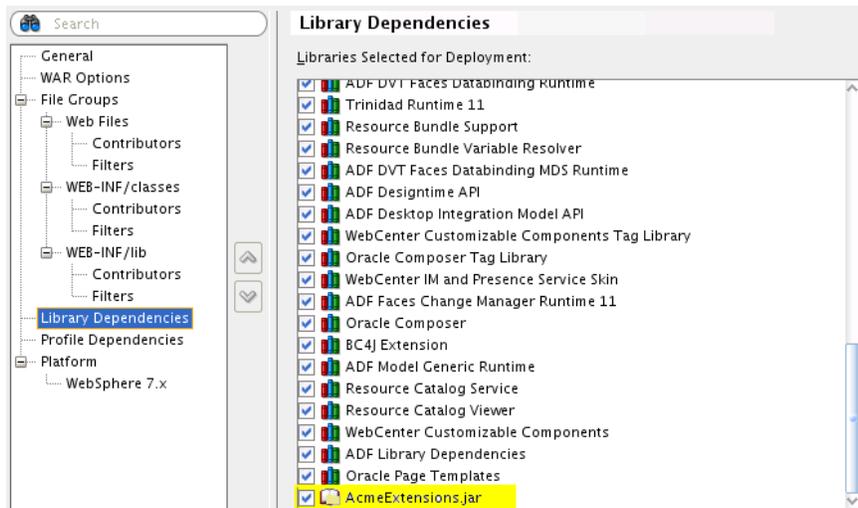
8. Build and deploy the shared library with your new project.

   For details, see section Building and Deploying a Custom Shared Library for the Spaces Application.

Using Spaces Extensions from an Earlier Release

You cannot automatically upgrade earlier versions of the Spaces extension workspace to 11.1.1.6. However, if you developed extensions for a previous release (11.1.1.2. or 11.1.1.3), you can utilize the **custom.webcenter.spaces** shared library that contains your customizations in Spaces 11.1.1.6.0 as follows:

1. Upgrade the WebCenter domain, WebCenter permissions and Spaces, as documented in *Oracle Fusion Middleware Patching Guide (11.1.1.6.0),* available on Oracle Technology Network:

   http://download.oracle.com/docs/cd/E23943_01/doc.1111/e16793/patch_webcenter.htm

   Very Important: When you upgrade a WebCenter domain from 11.1.1.2 or 11.1.1.3 to 11.1.1.6.0, you *must* set the property **upgradeCustomSpaces** to **1** (true). If you do not, resources that you developed, such as page templates and skins, will not be available in Spaces even if you redeploy the **custom.webcenter.spaces** shared library as described in steps 2 and 3.

   Immediately after the upgrade process, all your 11.1.1.2 or 11.1.1.3 customizations are available. The next step depends on what you want to do:

   - If no further customizations are required, there is no need to complete steps 2 or 3.

   - If you want to modify the **custom.webcenter.spaces** shared library in any way, continue with step 2.

   - If you make additional customizations using the latest Spaces extension workspace, continue with step 3.

2. Use ExtendWebCenterSpaces.jws (11.1.1.2. or 11.1.1.3) to deploy additional customizations in the custom.webcenter.spaces.war shared library to your Spaces 11.1.1.6.0 managed server.

   Note: Always use Oracle JDeveloper 11.1.1.2 or 11.1.1.3 to build and deploy extensions developed for this release.

   For detailed steps, see "*Extending WebCenter Spaces (11.1.1.2.0 and 11.1.1.3.0)*" whitepaper on Oracle Technology Network: http://www.oracle.com/technetwork/middleware/webcenter/index.html

3. Include **custom.webcenter.spaces.war** in the Spaces 11.1.1.6.0 shared library dependency list.

   a) If you have not done so already, download the Spaces (11.1.1.6.0) development ZIP from Oracle Technology Network:

http://download.oracle.com/otndocs/tech/webcenter/files/DesignWebCenterSpaces_ps5.zip

b) Open the DesignWebCenterSpaces.jws (11.1.1.6) workspace.

c) Navigate to: **WebCenterSpacesExtensionLibrary\Application Sources\WebCenterSpacesExtensionLibrary\public_html\WEB-INF\weblogic.xml**

d) Include **custom.webcenter.spaces** WAR file in the Spaces 11.1.1.6.0 shared library dependency list. For example, include the following <library-ref> tag:

```
<library-ref>
  <library-name>custom.webcenter.spaces</library-name>
</library-ref>
```

e) Redeploy extend.spaces.webapp.war.

For more detailed steps, refer to "Rebuilding the Spaces Shared Library List" in *Developer's Guide for Oracle WebCenter Portal*.

📖 *Developer's Guide for Oracle WebCenter Portal (11.1.1.6.0)* is available from Oracle Technology Network:
http://download.oracle.com/docs/cd/E23943_01/webcenter.1111/e10148/jpsdg_wcsres.htm

Moving Previous Extensions to SampleWebCenterSpacesExtensions.jws

If you want to move extensions developed for Spaces 11.1.1.2/11.1.1.3/11.1.1.4/11.1.1.5 into SampleWebCenterSpacesExtensions.jws (11.1.1.6), you must move your files manually. See also, the section Adding New Projects to the Sample Workspace.

The latest release of Spaces includes new tools and features that enable you to customize all aspects of your application. Oracle recommends that where possible, you re-build earlier customizations (such as, customized page templates, and so on) using Spaces built-in tools instead of JDeveloper.

Conclusion

This document describes how to extend Spaces 11.1.1.6.0 to meet the needs of your user base. For more information on Spaces, see the *User's Guide for Oracle WebCenter Portal: Spaces*, available on Oracle Technology Network at:

http://download.oracle.com/docs/cd/E23943_01/webcenter.1111/e10149/toc.htm

ORACLE®

Using Spaces Extension Samples (11.1.1.6.0)
December 2012 (Minor Revision)
Authors: Rosie Harvey
Contributing Authors:, Sunil Franklin, Vaibhav
Gupta, Rohit Kulkarni, Deepthi Umakanth

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com

Oracle is committed to developing practices and products that help protect the environment

0109