

An Oracle White Paper  
August 2011

# Using WebCenter Spaces Extension Samples (11.1.1.4.0)

Executive Overview .....	2
Introduction .....	2
Deploying and Undeploying Sample WebCenter Spaces Extensions	3
Overview.....	3
Downloading, Copying, and Exploring the WebCenter Spaces	
Sample Workspace .....	4
Setting Build and Deployment Options (config.properties) .....	8
Building and Deploying a Custom Shared Library for WebCenter	
Spaces.....	10
Removing WebCenter Spaces Extensions .....	11
Sample - Customizing User Profile Attributes .....	17
Sample – Adding a Crawler for Custom Profile Attributes.....	27
Sample - Creating Rule-Based Resources .....	34
Sample - Deploying Additional Task Flows.....	36
Sample – Customizing the Login Task Flow .....	40
Adding New Projects to the Sample Workspace.....	49
Using WebCenter Spaces Extensions from an Earlier Release .....	53
Conclusion .....	54

## Executive Overview

Oracle WebCenter Spaces is a pre-built, pre-integrated, and highly customizable WebCenter application. As you begin using WebCenter Spaces you may find there are some aspects that you want to modify or extend beyond the customization capabilities provided in WebCenter Spaces. For example, you may want to add custom functionality, such as custom Java code, additional 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 WebCenter Spaces shared libraries that contain your custom code.

## Introduction

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

Oracle recommends that you copy the sample workspace and use this as a starting point for building WebCenter 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:

- **Deploy custom Java code** – Develop custom session invalidation code and make it available in WebCenter Spaces.
- **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 WebCenter Spaces users.
- **Customize the Login task flow** – Configure the default Login task flow to authenticate users against Oracle Access Manager SSO.

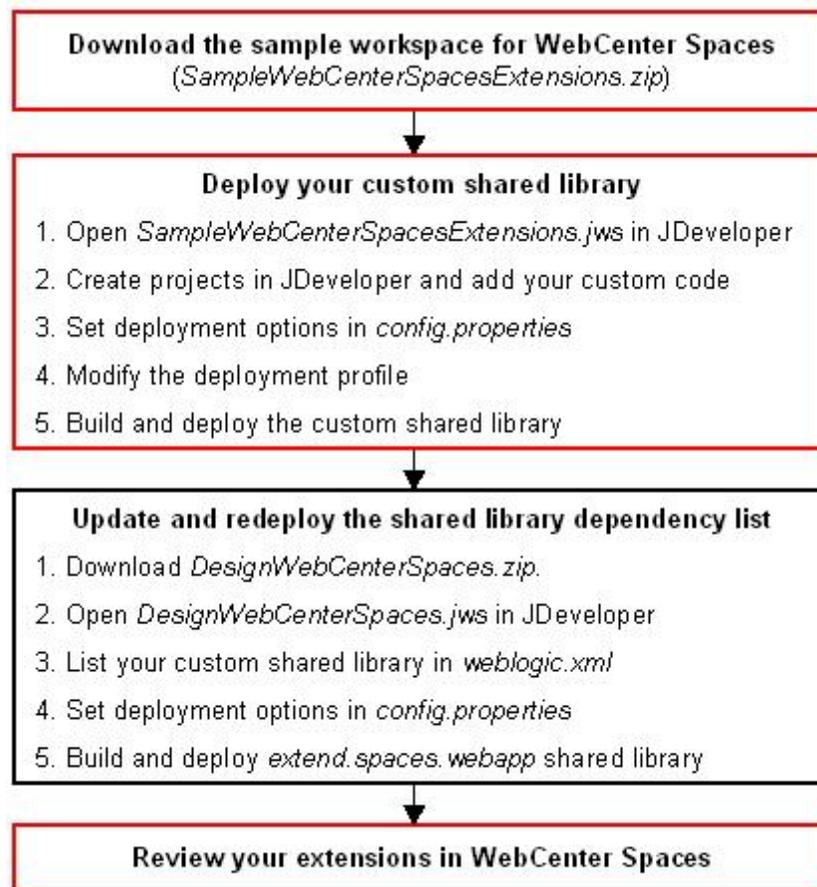
## Deploying and Undeploying Sample WebCenter Spaces Extensions

This section describes:

- Downloading and exploring the WebCenter Spaces extension samples
- Building and deploying a custom WebCenter Spaces shared library (.WAR)
- Reverting to a previous custom WebCenter Spaces shared library version

### Overview

To extend or customize WebCenter Spaces 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 WebCenter Spaces development workspace so you can register the name of your custom shared library in the *WebCenter Spaces shared library dependency list* and redeploy the updated list (in `extend.spaces.webapp`). The following diagram illustrates the process.



This whitepaper describes how to build and deploy custom WebCenter Spaces shared libraries that contain custom code (highlighted in red boxes). For instruction how to update and redeploy the WebCenter Spaces shared library dependency list, refer to "Extending WebCenter Spaces Using JDeveloper" in *Developer's Guide for Oracle WebCenter* available at [http://download.oracle.com/docs/cd/E17904\\_01/webcenter.1111/e10148/jpsdg\\_wcsres.htm](http://download.oracle.com/docs/cd/E17904_01/webcenter.1111/e10148/jpsdg_wcsres.htm).

## Downloading, Copying, and Exploring the WebCenter Spaces Sample Workspace

Sample WebCenter Spaces extensions are available from Oracle Technology Network.

1. Before you start:
  - a) Ensure WebCenter Spaces is installed, configured, and working as expected.
  - b) Download and install **Oracle JDeveloper 11g (11.1.1.4.0)**. 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 11.1.1.4.0** (oracle.webcenter.framework\_bundle.zip).
 

**Tip:** You can install extensions from JDeveloper's Help menu. Select **Help > Check for Updates**.
2. Download **SampleWebCenterSpacesExtensions.zip** from Oracle Technology Network: <http://download.oracle.com/otndocs/tech/webcenter/files/SampleWebCenterSpacesExtensions.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.

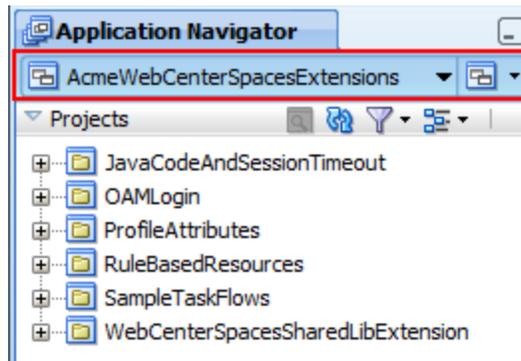
The .ZIP file contains the following files and folders:

### \SampleWebCenterSpacesExtensions

SampleWebCenterSpacesExtensions.jws	Use this workspace to review samples, develop extensions, as well as build and deploy shared libraries that contain your custom code.
\JavaCodeAndSessionTimeout	Project containing Java code that customizes the default session timeout.
\OAMLogin	Project that redirects the default Login task flow to Oracle Access Manager single sign-on.
\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 ( <code>UserSkinBean.java</code> ).
\SampleTaskFlows	Project containing a custom task flow that displays a list of pages in a bulleted list.
\WebCenterSpacesSharedLibExtension	A customizable WebCenter Spaces project that generates an application-level shared library (WAR file) containing your custom code.
\WCAppEARSharedLibExtension	Project that generates an application-level shared library (EAR file) containing custom code. (Only required for the profile crawler project).
\SourceFiles	Out-of-the-box content for you to edit/customize.
\langs	Default resource bundles for all languages
\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 folder **SampleWebCenterSpacesExtensions**.  
For example, `AcmeWebCenterSpacesExtensions`.
  - b) Rename the workspace **SampleWebCenterSpacesExtensions.jws**.  
For example, `AcmeWebCenterSpacesExtensions.jws`.
  - c) Open the following files and replace “**SampleWebCenterSpacesExtensions**” with your new name “**<new\_name>WebCenterSpacesExtensions.jws**”
    - `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 WebCenter Spaces extensions, as well as build and deploy shared libraries that contain your custom code.



The following projects display in the Application Navigator:

- **WebCenterSpacesSharedLibExtension** - generates a customized shared library WAR file for WebCenter 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)*.

- **Five sample projects:**
  - **JavaCodeAndSessionTimeout** - Project containing Java code that customizes the default session timeout.
  - **OAMLogin** - Project that redirects the default Login task flow to Oracle Access Manager single sign-on.
  - **ProfileAttributes** - Project containing code that overrides default profile attributes 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.

**Note:** The **ProfileCrawler** project is not displayed by default. If you want to deploy a custom profile attribute crawler, refer to the section "*Adding a Crawler for Custom Profile Attributes*".

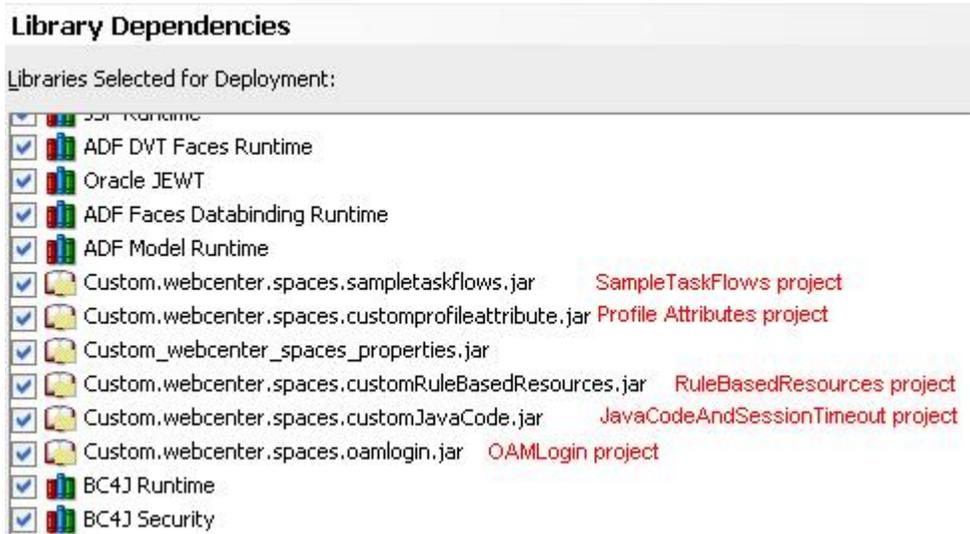
6. (Optional) Modify the WAR file deployment profile.

All five sample projects are *included* in the custom WebCenter 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**.

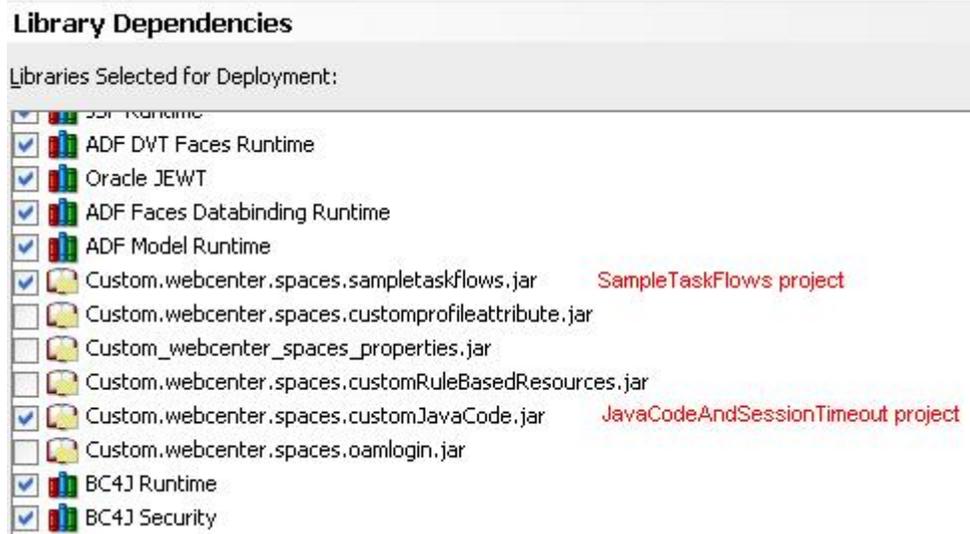
By default, all five sample projects are *included* in the custom WebCenter Spaces shared library (WAR file):



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.customJavaCode.jar** to include these projects.



Notes:

- If you choose **not** to deploy the **JavaCodeAndSessionTimeout** project, you must comment out the `<servlet>` and `<servlet-mapping>` entries for the session invalidation servlet in `web.xml` located at: `WebCenterSpacesShareLibExtension\WebContent\WEB-INF`
- **Custom\_webcenter\_spaces\_properties.jar** is provided for backward compatibility. You can ignore this JAR if you are customizing WebCenter 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 WebCenter Spaces managed server.

### Setting Build and Deployment Options (config.properties)

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

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

1. 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 WebCenter Spaces installation as shown in the table.

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

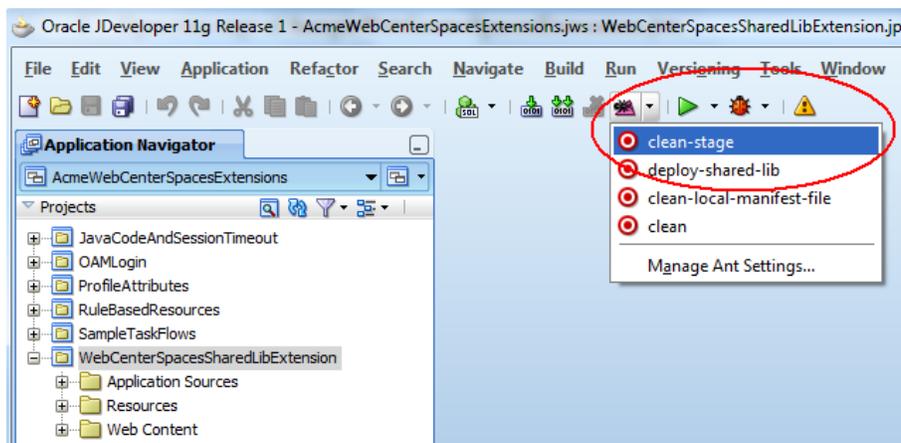
PROPERTY NAME	DESCRIPTION
<code>jdeveloper.install.home.directory</code>	Base directory where JDeveloper is installed. The directory you specify contains other folders such as <code>jdeveloper</code> , <code>wlserver_10.3</code> , <code>modules</code> , and so on.
<code>wlst.executable</code>	Name of the WLST executable file. Either <code>wlst.cmd</code> (on Windows) or <code>wlst.sh</code> (on Linux).
<code>extending.spaces.home.dir</code>	Path to your copy of the <code>SampleWebCenterSpacesExtensions</code> project. For example, <code>&lt;path&gt;AcmeWebCenterSpacesExtensions</code>
<code>oracle.jdeveloper.ojdeploy.path</code>	Path to <code>ojdeploy.exe</code> (Windows) or <code>ojdeploy</code> (Linux).
<code>wls.port</code>	Port number on which the WebLogic Administration Console is running.
<code>wls.host</code>	Host machine on which WebCenter Spaces is running and where the custom WebCenter Spaces shared library is to be deployed.
<code>wls.userkey</code>	Name and location of the file storing administrator passwords. <code>config.properties</code> describes how to generate the file using WLST. The file can be copied to any suitable, accessible location.
<code>wls.userconfig</code>	Name and location of the file storing administrator user details. <code>config.properties</code> describes how to generate the file using WLST. The file can be copied to any suitable, accessible location.
<code>wls.target</code>	Targets on which the WebCenter Spaces shared library is to be deployed.
<code>wls.webcenter.app.target</code>	Targets on which the <code>webcenter.ear</code> file is to be deployed.
<code>webcenter.app.name</code>	Name of the WebCenter Spaces application.
<code>customer.library.name</code>	Name for your custom shared library WAR file. For example: <code>com.acme.custom.webcenter.spaces</code>
<code>customer.ear.library.name</code>	Name for your custom shared library EAR file. For example: <code>com.acme.custom.webcenter.spaces.fwk</code> (Only required if you deploy the ProfileCrawler project).
<code>customer.name</code>	Company name or identifier.
<code>restart.implementation.version.suffix</code>	Controls incremental implementation version numbers for the custom shared library WAR file.
<code>restart.implementation.version.suffix.ear</code>	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 WebCenter Spaces

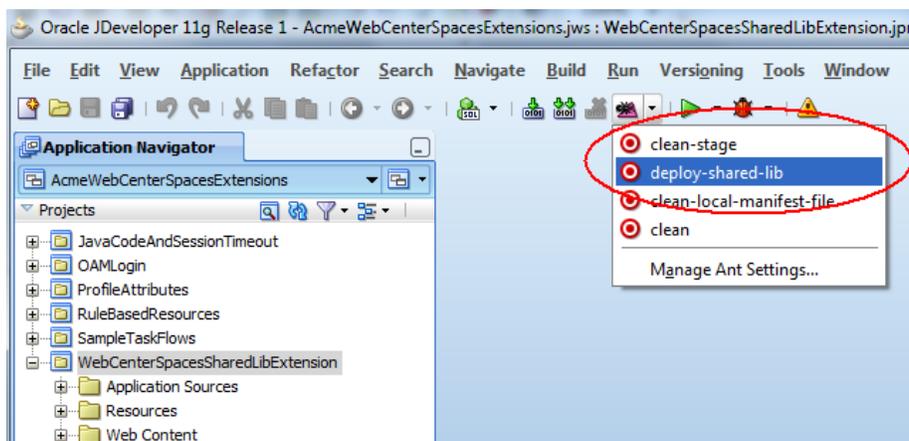
After setting up a deployment profile for the **WebCenterSpacesSharedLibExtension** project and configuring **config.properties**, you can build and deploy your custom WebCenter Spaces shared library WAR file:

1. Open your copy of **SampleWebCenterSpacesExtensions.jws**.
2. In the Application Navigator, select **WebCenterSpacesSharedLibExtension**.
3. To build a new version of the WebCenter 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 WebCenter 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**.

WebCenter Spaces 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*.

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

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

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

**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* is available from Oracle Technology Network: [http://download.oracle.com/docs/cd/E17904\\_01/webcenter.1111/e10148/jpsdg\\_wcsres.htm](http://download.oracle.com/docs/cd/E17904_01/webcenter.1111/e10148/jpsdg_wcsres.htm)

## Removing WebCenter Spaces Extensions

You can revert to the out-of-the-box WebCenter Spaces deployment at any time. For details, see "Reverting to a Previous Shared Library Version" in *Developer's Guide for Oracle WebCenter*.

## Removing OAM Login Button Customizations in MDS

If you deployed the sample OAM Login code, an additional step is required to delete the login button customization document in MDS. Use the WLST command `deleteMetadata` as follows:

1. Navigate to your WebCenter Oracle home directory and invoke the WLST script:
  - (UNIX) `WC_ORACLE_HOME/common/bin/wlst.sh`
  - (Windows) `WC_ORACLE_HOME\common\bin\wlst.cmd`

2. Connect to the Administration Server for Oracle WebCenter. For example:

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

3. Delete the MDS customization document `LoginFormView.jsff.xml`:

```
deleteMetadata(application='webcenter', server='WC_Spaces',
docs='/oracle/webcenter/webcenterapp/view/taskflows/security/mdssys/cust/site/webcenter/LoginFormView.jsff.xml')
```

#### Remove other changes relating to the OAMLogin project:

1. In `setDomainEnv.sh`, remove `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/j
rf.jar${CLASSPATHSEP}${OBACCESS_INSTALL_DIR}/oblix/lib/jobaccess.jar
${CLASSPATHSEP}${POST_CLASSPATH}"
export POST_CLASSPATH
else
POST_CLASSPATH="${COMMON_COMPONENTS_HOME}/modules/oracle.jrf_11.1.1/j
rf.jar${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}/scratch/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 WebCenter Spaces (if required).

For detail, see *Administrator's Guide for Oracle WebCenter*. Available from Oracle Technology Network:

[http://download.oracle.com/docs/cd/E21764\\_01/webcenter.1111/e12405/wcadm\\_security\\_sso.htm](http://download.oracle.com/docs/cd/E21764_01/webcenter.1111/e12405/wcadm_security_sso.htm)

8. Restart the domain.

## Sample – Customizing the Session Timeout

If you want to implement some behavior in WebCenter Spaces that requires the use of custom logic, you must deploy the custom Java code to a WebCenter Spaces shared library. The `SampleWebCenterSpacesExtensions` workspace includes a sample project to illustrate Java code deployment-- **JavaCodeAndSessionTimeout**.

The sample contains several files:

- **SessionTimeoutPhaseListener.java** - injects JavaScript into the browser causing the session to timeout 'n' seconds earlier than the default setting (45 minutes). In the sample, n is 300 seconds (or 5 minutes) reducing the session timeout to 40 minutes.
- **SessionInvalidationServlet.java** - invalidates the user session after the period specified in `SessionTimeoutPhaseListener.java` (40 minutes) and then navigates to a specified HTML page. By default, the page specified is:  
`/webcenter/custom/webcenter/spaces/SessionExpired.html`
- **SessionExpired.html** – informs users that their WebCenter Spaces session has expired and provides a link to redirect users to the login page.
- **web.xml** – includes the custom servlet and associated mappings:

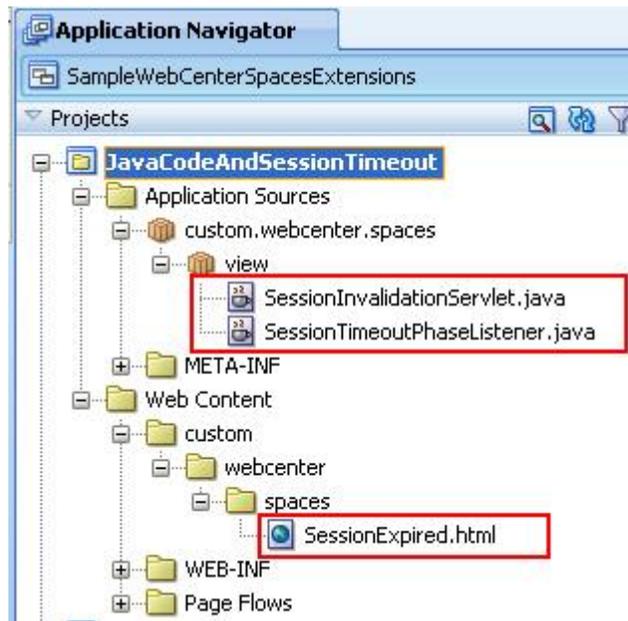
```
<servlet>
<!-- invalidates the WebCenter Spaces session based on code in the
JavaCodeAndSessionTimeout project-->
    <servlet-name>SessionInvalidationServlet</servlet-name>
    <servlet-
class>custom.webcenter.spaces.view.SessionInvalidationServlet</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>SessionInvalidationServlet</servlet-name>
    <url-pattern>/sessioninvalidation</url-pattern>
</servlet-mapping>
```

The `web.xml` file containing the servlet reference is located at:

```
WebCenterSpacesSharedLibExtension\Web Content\WEB-INF
```

The following steps describe how to deploy the custom Java code to the WebCenter Spaces shared library using the **SampleWebCenterSpacesExtension** workspace:

1. Open your copy of `SampleWebCenterSpacesExtensions.jws` in JDeveloper.
2. Navigate to `JavaCodeAndSessionTimeout\Application Sources\custom.webcenter.spaces\view`



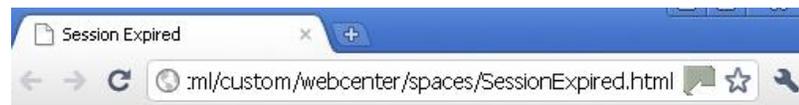
3. Customize the session timeout period, if required:
  - a) Open **SessionTimeoutPhaseListener.java**.
  - b) Specify a value for **secondsBefore**.

The sample value is 300 seconds (5 minutes) and this produces a **40-minute** session timeout, that is, the session times out *earlier* than the default session timeout configured in `web.xml` (45 minutes).

```
// time in seconds before (the value in web.xml) to expire the session
int secondsBefore = 300; // 300 seconds equals 5 minutes
if (secsTimeout > 0)
```

4. Customize the HTML page displayed to users when their session expires, if required.

The default page is shown here:

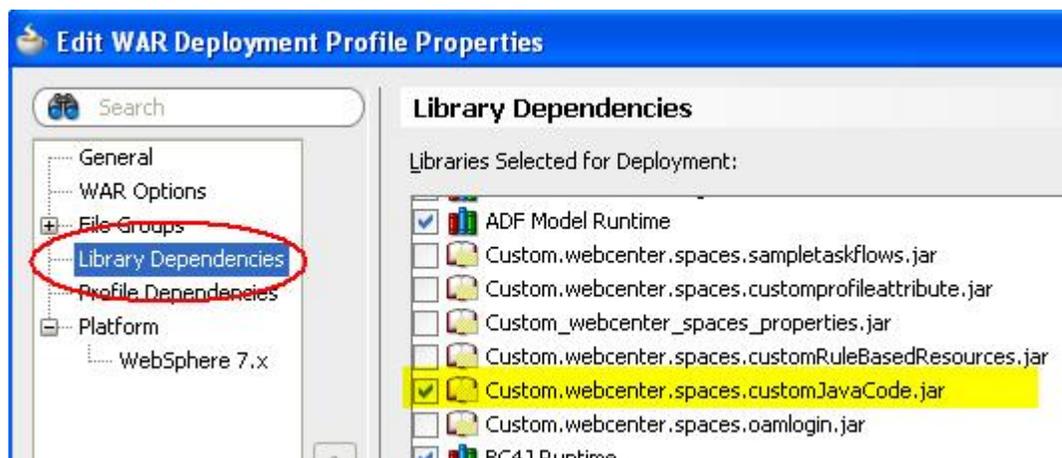


Your WebCenter Spaces session has expired.

Click [here](#) to go back.

- a) Open **SessionExpired.html**.  
 Located at: JavaCodeAndSessionTimeout\Web  
 Content\custom\webcenter\spaces
  - b) Edit the HTML as required.
5. Alternatively, redirect users to a different HTML page:
- a) Open **SessionTimeoutPhaseListener.java**.
  - b) Edit the default value for **sendRedirect**.
 

```
// if required change SessionExpired.html to your own html page.
response.sendRedirect("/webcenter/custom/webcenter/spaces/SessionExpired.html");
```
  - c) Ensure that the new HTML file is available at the location specified.
6. Ensure that the **JavaCodeAndSessionTimeout** project JAR file is selected in the deployment profile:



7. Deploy the custom Java code.
- For details, see section [Building and Deploying a Custom WebCenter Spaces Shared Library](#).
8. Login to WebCenter Spaces, and verify that the customized code is available and working as expected.

Leave the browser idle for the time specified (the sample timeout is 40 minutes), and then verify that the session expires and the correct redirect page displays.

## Sample - Customizing User Profile Attributes

In WebCenter 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 WebCenter 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

### 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:

- **Department** – replace with the hard coded string **Sales Department** as follows:
 

```
public String getDepartment()
    throws ProfileException
    {
        return "Sales Department";
    }
```
- **Photo** - replace user photos 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. WebCenter 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.

### Adding Custom Profile Attributes

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

- **Project**
- **SIP Address**

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

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

If you add your own profile attributes to `ExtendedProfileAttributes.java` you can reference them in the same way.

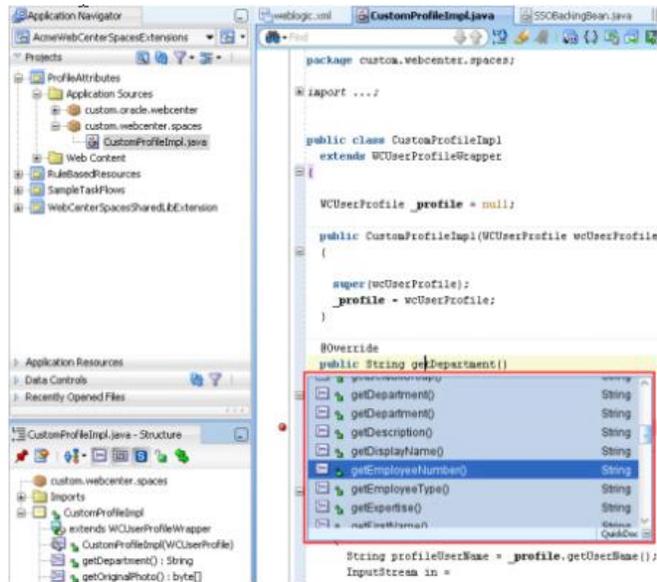
### Deploying the ProfileAttributes Project:

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

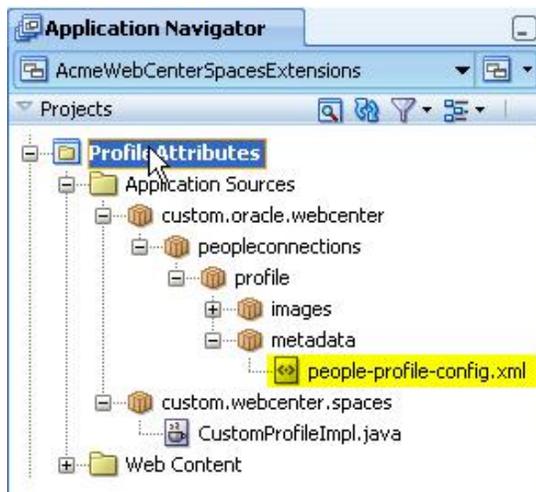
1. Open your copy of **SampleWebCenterSpacesExtensions.jws** in JDeveloper.
2. To override default profile attributes:
  - a) To review the sample class, navigate to **ProfileAttributes> Application Sources> custom.webcenter.spaces> CustomProfileImpl.java**

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.

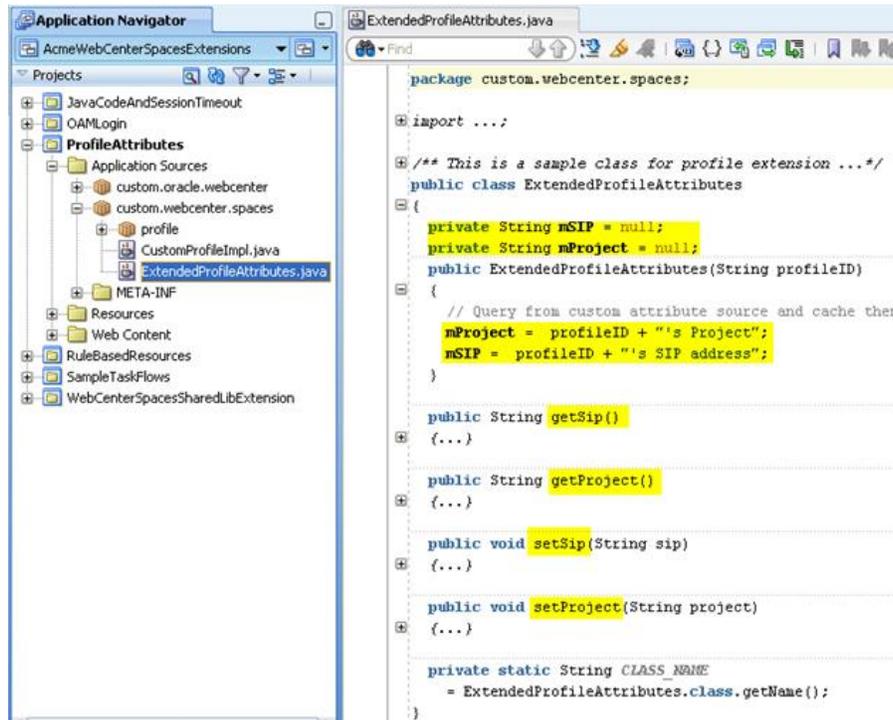
The sample specifies `CustomProfileImp.java`. Any valid class name and package name is acceptable here.



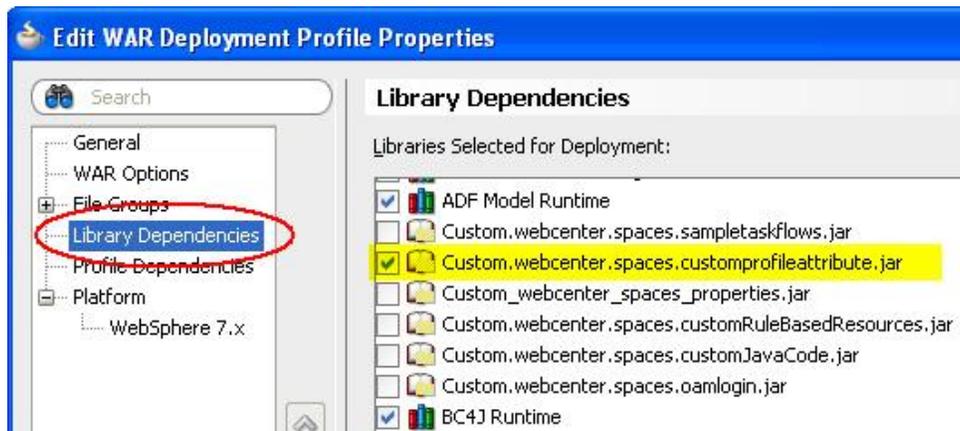
#### Notes:

- Do not change anything else in this file.
  - Only reference one implementation class from the `people-profile-config.xml` file.
  - Do not rename the package containing `people-profile-config.xml` or modify the structure in any way.
3. To add profile attributes:
- a. To review the sample class, navigate 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 WebCenter Spaces Shared Library](#).
6. Login to WebCenter Spaces, and open your profile to verify that the customized attributes display.

If you deployed the samples provided:

- a. Verify the following:
  - **Department** – Displays ‘Sales Department’
  - **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.

The screenshot shows a user profile page with the following details:

- Summary** section containing:
  - About Me**: A large empty text area.
  - Display Name**: Monty Smith
  - Email**: monty@acme.org
  - Department**: Sales Department (highlighted with a red box)
  - Designation**: Empty field
  - Manager**: Empty field
  - Direct Reports**: Empty field
  - Phone**: +86-10-82786017
  - Time Zone**: (UTC-06:00) El Salvador
- Photo**: A placeholder image of a person with a green tie, with a "Change Photo" link below it.

- 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:

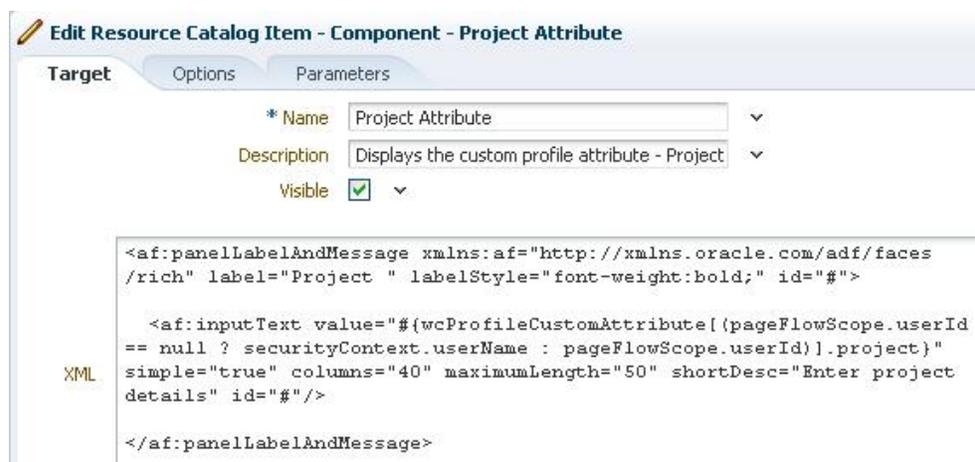
- i) Navigate to the Resource Catalog design tab (**Administration>Resources>Resource Catalogs**).
- ii) Create a new resource catalog, if required.
- iii) Edit the resource catalog, and then choose **Add>Component**.

- 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"
maxLength="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 WebCenter 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 WebCenter 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 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 (or any page) in which you want to add the new profile attributes.
- ix) Click **Edit Page** to open the page in Oracle 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 resource 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 resource catalog.
- c. Alternatively, upload *pre-customized* user profile 'About' pages directly to WebCenter 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 WebCenter Spaces as follows:

- i) Navigate to your WebCenter 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. For example:
- ```
connect(username='weblogic', password='mypassword',
url='myhost.example.com:7001')
```

iii) Upload About page customizations:

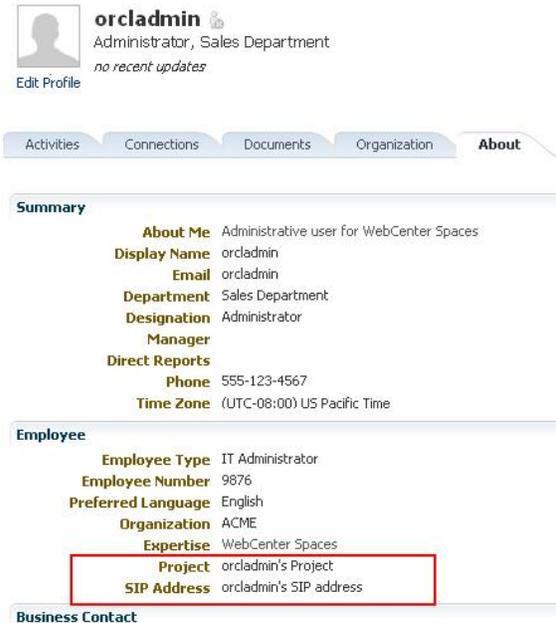
```
importMetadata (application='webcenter',
server='WC_Spaces',
fromLocation='<sampleziplocation>ProfileAttributes/m
ds/,
docs='oracle/webcenter/peopleconnections/profile/vie
w/jsf/regions/ootbviews/mdssys/cust/site/webcenter/v
iewemployee.jsff.xml')
```

```
importMetadata (application='webcenter',
server='WC_Spaces',
fromLocation='<sampleziplocation>ProfileAttributes/m
ds/,
docs='oracle/webcenter/peopleconnections/profile/vie
w/jsf/regions/ootbedit/mdssys/cust/site/webcenter/ed
itemployee.jsff.xml')
```

iv) Login to WebCenter Spaces.

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:

The screenshot shows the 'About' page for the user 'orcladmin'. The user's profile information is displayed, including their name, department, and designation. The 'Project' attribute is highlighted with a red box, showing the value 'orcladmin's Project'. Other attributes include 'Employee Type', 'Employee Number', 'Preferred Language', 'Organization', and 'Expertise'.

| Summary                                  |                             |
|------------------------------------------|-----------------------------|
| Administrative user for WebCenter Spaces |                             |
| <b>About Me</b>                          |                             |
| <b>Display Name</b>                      | orcladmin                   |
| <b>Email</b>                             | orcladmin                   |
| <b>Department</b>                        | Sales Department            |
| <b>Designation</b>                       | Administrator               |
| <b>Manager</b>                           |                             |
| <b>Direct Reports</b>                    |                             |
| <b>Phone</b>                             | 555-123-4567                |
| <b>Time Zone</b>                         | (UTC-08:00) US Pacific Time |
| Employee                                 |                             |
| <b>Employee Type</b>                     | IT Administrator            |
| <b>Employee Number</b>                   | 9876                        |
| <b>Preferred Language</b>                | English                     |
| <b>Organization</b>                      | ACME                        |
| <b>Project</b>                           | orcladmin's Project         |
| <b>Expertise</b>                         | WebCenter Spaces            |

## Sample – Adding a Crawler for Custom Profile Attributes

User profile attributes that you add through the ProfileAttributes project are not automatically searchable in WebCenter Spaces. If you are using Oracle Secure Enterprise Search (Oracle SES) to search WebCenter Spaces, 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 your 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. Follow the instructions below

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

This includes setting up crawl sources for WebCenter Spaces in Oracle SES, and then configuring WebCenter Spaces to use Oracle SES. For detailed steps, refer to “Configuring Oracle SES to Search WebCenter Spaces Applications” in *Administrator’s Guide for Oracle WebCenter*.

 *Administrator’s Guide for Oracle WebCenter* is available from Oracle Technology Network: [http://download.oracle.com/docs/cd/E21764\\_01/webcenter.1111/e12405/wcadm\\_search.htm#WCADM9801](http://download.oracle.com/docs/cd/E21764_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 (WCAppEarExtensionLibrary) is provided with the sample but it is hidden by default. Follow these steps to expose both WCAppEarExtensionLibrary and the sample ProfileCrawler project:

- i) Ensure that your copy of SampleWebCenterSpacesExtensions is not open in JDeveloper.
- ii) In a text editor, such as Notepad, open your copy of SampleWebCenterSpacesExtensions.jws.
- iii) 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>
```

- iv) Uncomment the WCAppEarExtensionLibrary project entry.

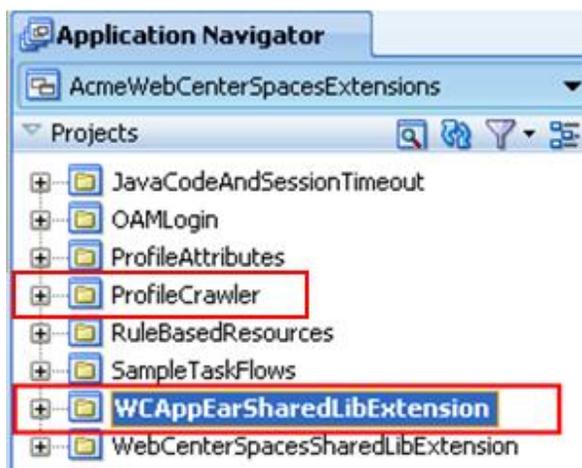
Change:

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

To:

```
<hash><url n="URL"
path="WCAppearExtensionLibrary/WCAppearExtensionLib
rary.jpr"/></hash>
```

- v) 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**):

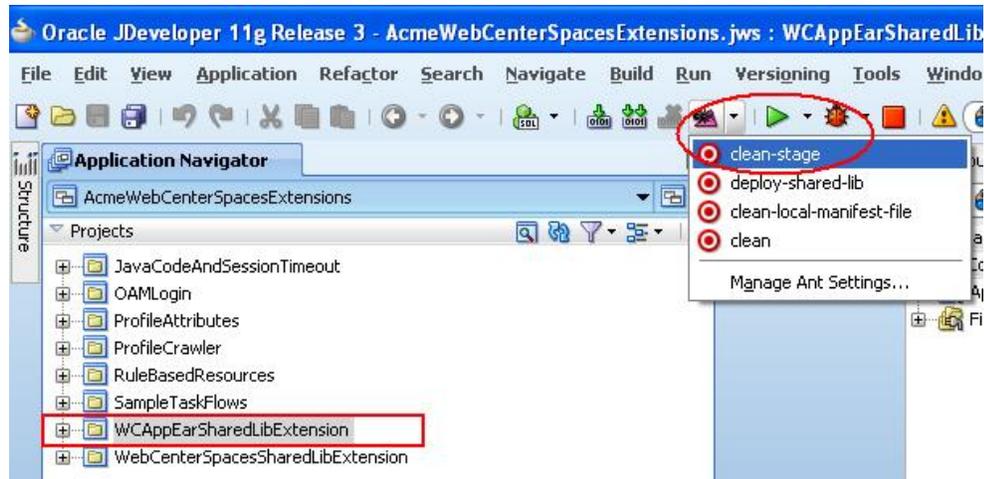


3. Set deployment options for the EAR-level shared library that will contain the custom Oracle SES crawler code.
  - i) Open your copy of `SampleWebCenterSpacesExtensions.jws`.
  - ii) Expand **WebCenterSpacesSharedLibExtension** and open the **Resources** folder.
  - iii) Specify the following EAR deployment options in `config.properties` (located under the **WebCenterSpacesSharedLibExtension\Resources** folder):

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

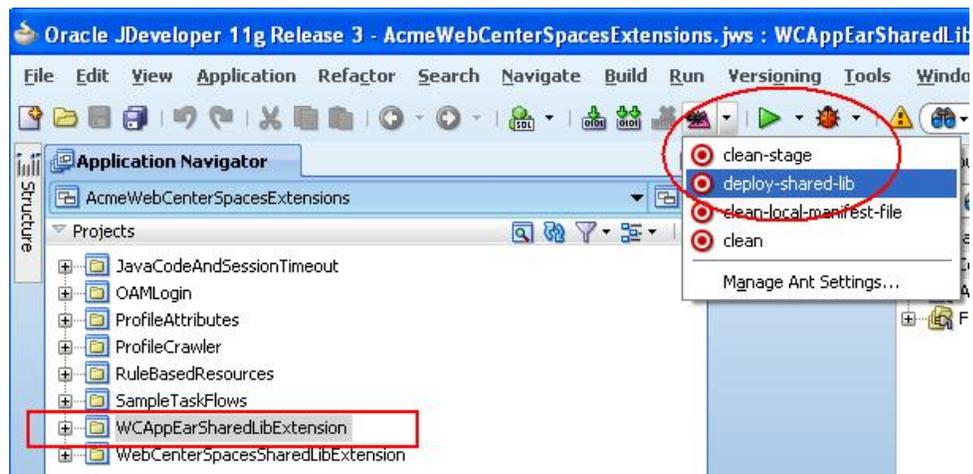
- iv) Save your updates to `config.properties`.
4. Build and deploy the custom Oracle SES crawler code:
  - a) In the Application Navigator, select **WCAppearSharedLibExtension**.

- b) To build the WebCenter 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 WebCenter 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 WebCenter Space's shared library list.
  - a) Download the required WebCenter 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 WebCenter Spaces Development" in *Developer's Guide for Oracle WebCenter*. For details, see:
 

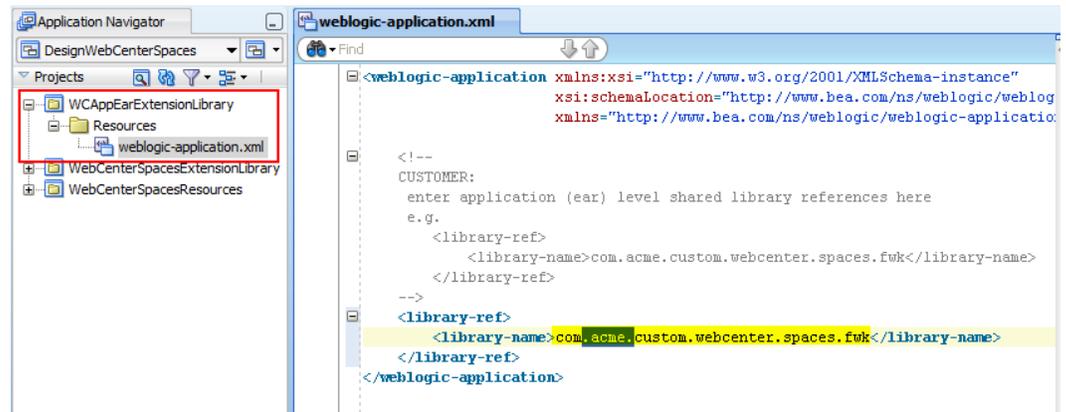
[http://download.oracle.com/docs/cd/E21764\\_01/webcenter.1111/e10148/jpsdg\\_wcs\\_res.htm#JPSDG10089](http://download.oracle.com/docs/cd/E21764_01/webcenter.1111/e10148/jpsdg_wcs_res.htm#JPSDG10089)
  - b) Enable *application-level* shared library list deployment:
    - i. Ensure that your copy of `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. Navigate to `weblogic-application.xml`, under **Application Sources\WebCenterSpacesExtensionLibrary\public\_html\WEB-INF**.
    - iv. Add a `<library-ref>` entry for the application-level EAR shared library that contains the ProfileCrawler code.



For example, include the library entry:

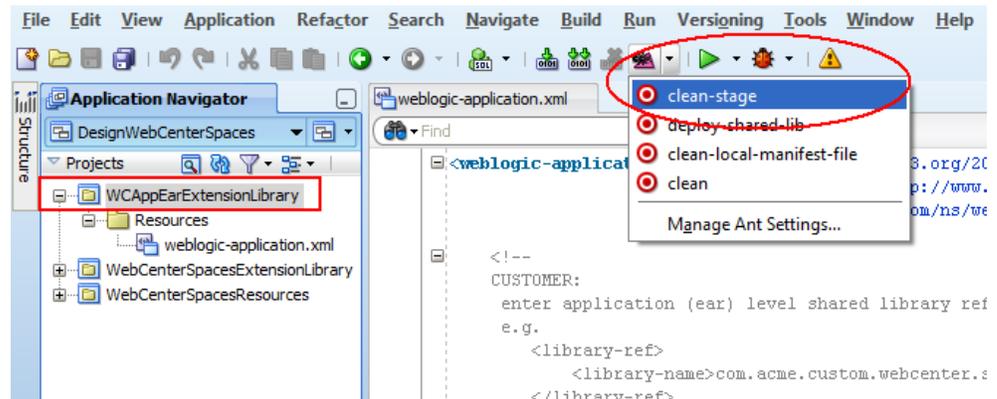
```
<library-ref>
  <library-name>
    com.mycompanyname.custom.webcenter.spaces.fwk
  </library-name>
</library-ref>
```

**Note:** Ensure that the shared library EAR file that you reference is deployed on the WebCenter 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 WebCenter Spaces installation.
 

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

[http://download.oracle.com/docs/cd/E21764\\_01/webcenter.1111/e10148/jp\\_sdg\\_wcsres.htm#JPSD10101](http://download.oracle.com/docs/cd/E21764_01/webcenter.1111/e10148/jp_sdg_wcsres.htm#JPSD10101)
  - 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 WebCenter 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 WebCenter Spaces managed server. To verify the new deployment, login to the WLS Administration Console, navigate to the deployment overview page, and check the implementation version, that is, navigate to:

**Deployments> 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* for information. Available from Oracle Technology Network at:  
[http://download.oracle.com/docs/cd/E17904\\_01/webcenter.1111/e12405/wcadm\\_search.htm#WCADM6569](http://download.oracle.com/docs/cd/E17904_01/webcenter.1111/e12405/wcadm_search.htm#WCADM6569)

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

## Sample - Creating Rule-Based Resources

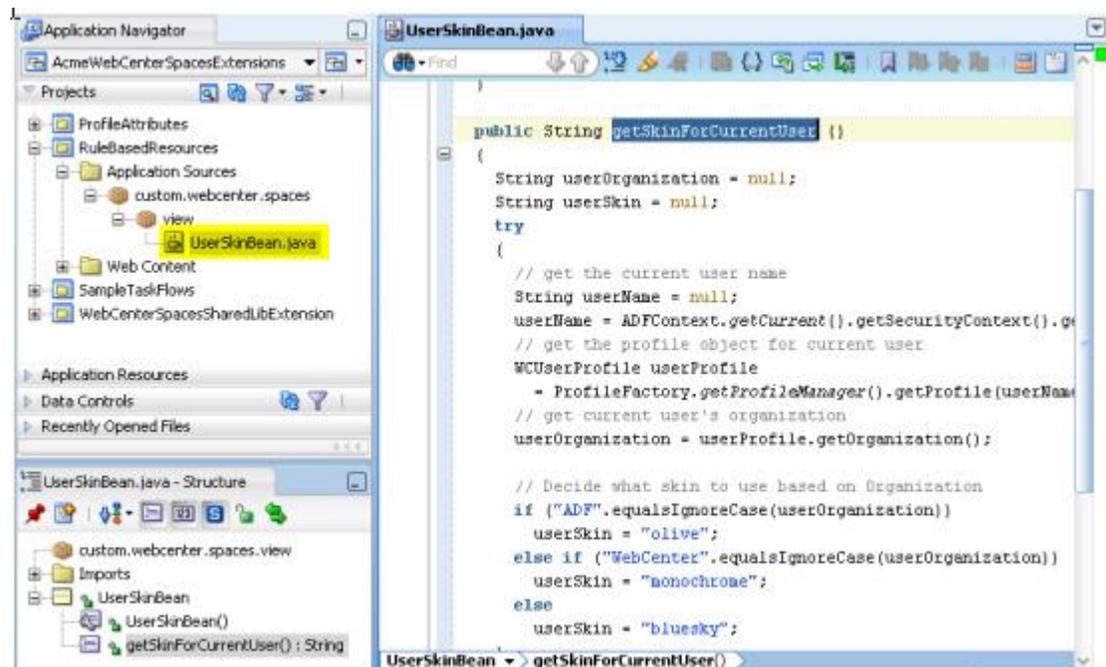
The SampleWebCenterSpacesExtensions workspace includes sample Java code that changes the WebCenter Spaces skin based on custom logic in `UserSkinBean.java` where `getSkinForCurrentUser` returns the skin to be used for the currently logged in user based on his organization. An 'olive' skin is returned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

You can write your 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.

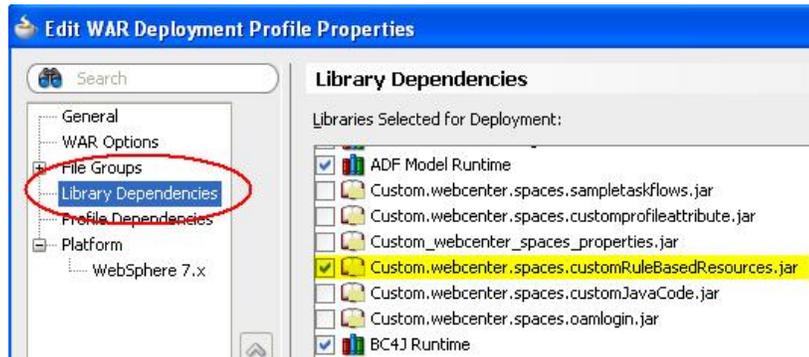
Alternatively, you can follow the same steps but deploy your own custom code.

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.



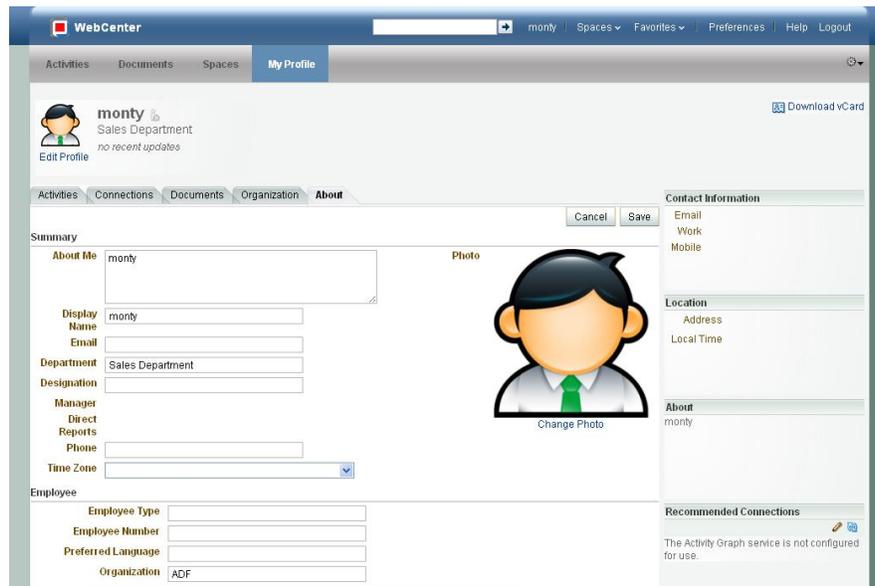
4. Deploy the custom code.

For details, see section [Building and Deploying a Customized WebCenter Spaces .WAR](#).

5. Login in to WebCenter Spaces as an Administrator to test the custom code.

For example:

- a) Open WebCenter 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, WebCenter Spaces provides a range of collaborative, social networking, and personal productivity task flows. WebCenter 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 WebCenter 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:

[http://download.oracle.com/docs/cd/E17904\\_01/web.1111/b31974/partpage3.htm](http://download.oracle.com/docs/cd/E17904_01/web.1111/b31974/partpage3.htm)

### Out-of-the-Box Task Flows

Many built-in task flows are available with WebCenter 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*.

 *Developer's Guide for Oracle WebCenter* is available from Oracle Technology Network:

[http://download.oracle.com/docs/cd/E17904\\_01/webcenter.1111/e10148/jpsdg\\_wcsres.htm](http://download.oracle.com/docs/cd/E17904_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\WebContent\ListOfPages.jsff

### Packaging and Deploying the Sample Task Flow

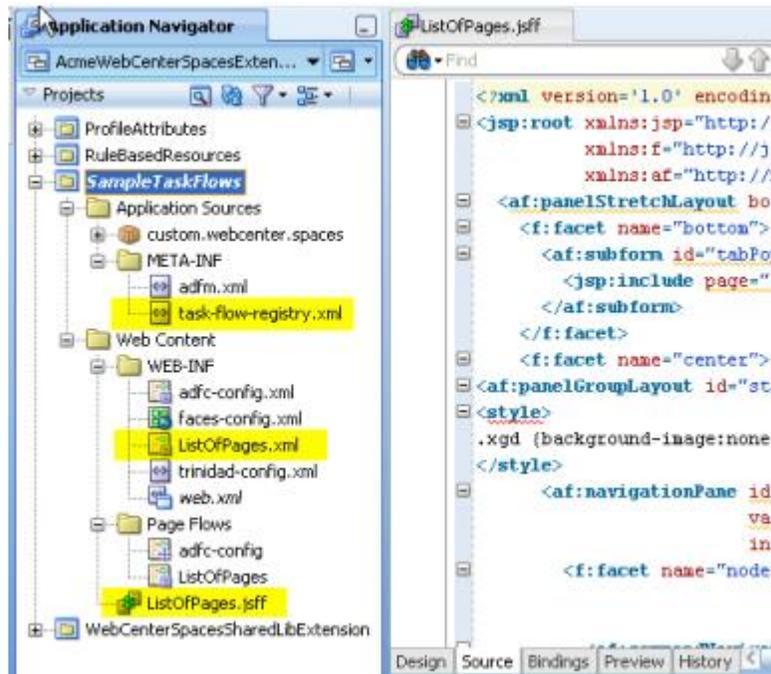
Additional task flows must be packaged in an *ADF Library JAR* and deployed in a WebCenter Spaces shared library. The following steps describe how to deploy a sample task flow using the **SampleWebCenterSpacesExtensions** workspace.

**Note:** You must create a new project if you want to deploy task flows of your own. For details, refer to the section [Adding New Projects to the Sample Workspace](#).

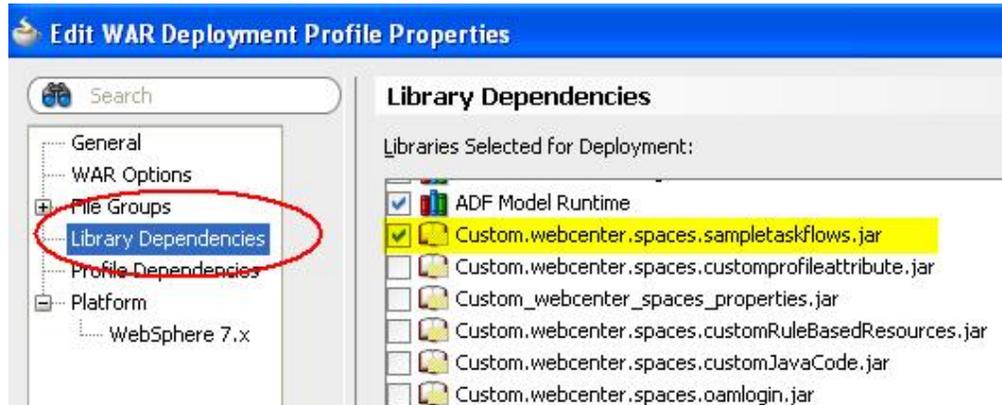
To deploy the sample task flow:

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 WebCenter Spaces Shared Library](#).

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

Follow these 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 Administration Resource tab.

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

- To add the task flow to a Spaces-specific resource catalog, navigate to the Space Settings Resource tab.

`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 Spaces*

 *User’s Guide for Oracle WebCenter Spaces* is available from Oracle Technology Network:

[http://download.oracle.com/docs/cd/E17904\\_01/webcenter.1111/e10149/catalogs.htm](http://download.oracle.com/docs/cd/E17904_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 “**ListOfPages**” in the Search field.

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

Associate the resource catalog with pages/page templates in the Home Space or a specific Space:

g) Do one of the following:

- To associate the resource catalog with pages or page templates in the **Home Space**, navigate to the WebCenter Administration General Configuration tab.

<http://host:port/webcenter/spaces/admin/general>

Configure the resource catalog. Use one of the following properties:

- Resource Catalog for Home Space
- Resource Catalog for Business Role Pages
- 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 Space Settings General tab.

<http://host:port/webcenter/spaces/spaceName/admin/general>

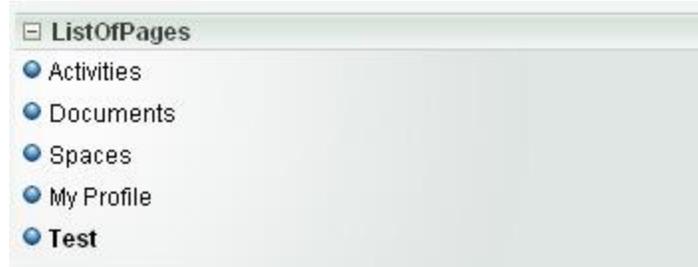
Configure the resource catalog. Use one of the following properties:

- 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 to the page.



See also, “Adding Resource Catalog Component to a Page” in *User’s Guide for Oracle WebCenter Spaces*.

 *User’s Guide for Oracle WebCenter Space* is available from Oracle Technology Network:  
[http://download.oracle.com/docs/cd/E17904\\_01/webcenter.1111/e10149/pages\\_build.htm](http://download.oracle.com/docs/cd/E17904_01/webcenter.1111/e10149/pages_build.htm)

## Sample – Customizing the Login Task Flow

Out-of-the-box, WebCenter 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 attempt to access a protected URL.

You can provide a similar SSO experience by modifying the login button code on the default Login task flow, in lieu of a SSO login page from a fully configured SSO setup. The SampleWebCenterSpacesExtensions workspace includes modified login button code for the WebCenter Spaces Login task flow and appropriate backing code for you to deploy.

**Note:** The sample requires Oracle Access Manager (OAM) 10.1.4. If you want to use the sample, ensure that OAM is installed on the WebLogic Server where WebCenter Spaces is running and that WebCenter Spaces is configured to use Single Sign-On with OAM. See also, *Configuring Oracle Access Manager (OAM) SSO*.

(The sample is validated on Oracle Internet Directory (OID) Release 11.1.1.3.0.)

This sample comprises the following key files:

SSO TASK FLOW PROJECT	DESCRIPTION
/OAMLogin	Folder containing custom code for the WebCenter Spaces Login task flow.
/Application Sources	
/custom/webcenter/spaces/view/OAMBackingBean.java	Java code that uses OAM SDK API to authenticate users against SSO using credentials provided through the Login task flow.
/META-INF/task-flow-registry.xml	Task flow registry file associated with the sample. This file lists LoginTaskflow.xml which publishes the Login task flow in the WebCenter Spaces resource library.  When a task flow is listed in the resource library, it can be exposed to users through the Resource Catalog, for inclusion on WebCenter Spaces pages.
/Resources/mds/oracle/webcenter/webcenterapp/view/taskflows/security/mdssys/cust/site/webcenter/LoginFormView.jsff.xml	Changes the Login button code to point to new OAM login code.
/lib/jobaccess.jar	OAM SDK API library required for the backing code in the sample to compile.  This library must not be packaged when the sample is deployed because the library will already exist at runtime if OAM is installed on WebLogic Server.
/lib/webcenter_oam.jar	Startup and shutdown Listener classes for OAM SDK.

To use the sample **OAMLogin** project, follow these steps:

1. Modify WebCenter Spaces Login button code.
2. Package and deploy the OAM login code.
3. Ensure the WebCenter Spaces login page is JPSX.
4. Configure OAM.
5. Validate the custom OAM login code.

Each step is described in detail below.

### Step 1 Modify the Login Task Flow Button Code

The OAMLogin project contains a customized version of `LoginFormView.jsff.xml`. This sample rewires the WebCenter Spaces login button to invoke OAM login code as follows:

```
<mds:replace node="submitLogin"/>
<mds:insert parent="logfrmpfl1" position="last">
<af:commandButton
xmlns:af="http://xmlns.oracle.com/adf/faces/rich" id="e7487585626"
text="Login" action="#{oamLoginBackingBean.doLogin}"/>
</mds:insert>
</mds:customization>
```

You can use the WLST command `importMetadata` to upload the login button customizations directly to WebCenter Spaces as follows:

4. Navigate to your WebCenter Oracle home directory and invoke the WLST script:
  - (UNIX) `WC_ORACLE_HOME/common/bin/wlst.sh`
  - (Windows) `WC_ORACLE_HOME\common\bin\wlst.cmd`

5. Connect to the Administration Server for Oracle WebCenter. For example:

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

6. Upload Login task flow customizations:

```
importMetadata(application='webcenter', server='WC_Spaces',
fromLocation='<sampleziplocation>/OAMLogin/mds',
docs='oracle/webcenter/webcenterapp/view/taskflows/security
/mdssys/cust/site/webcenter/LoginFormView.jsff.xml')
```

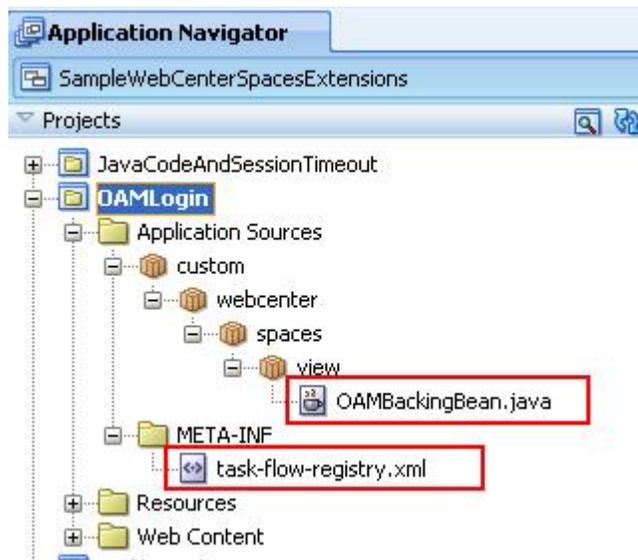
## Step 2 Package and Deploy the OAM Login Code

OAMBackingBean.java, the custom OAM SSO login code, is packaged in the ADF Library JAR - custom.webcenter.spaces.oamlogin.jar. You must add this JAR file to a WebCenter Spaces shared library and deploy your customized shared library to WebCenter Spaces.

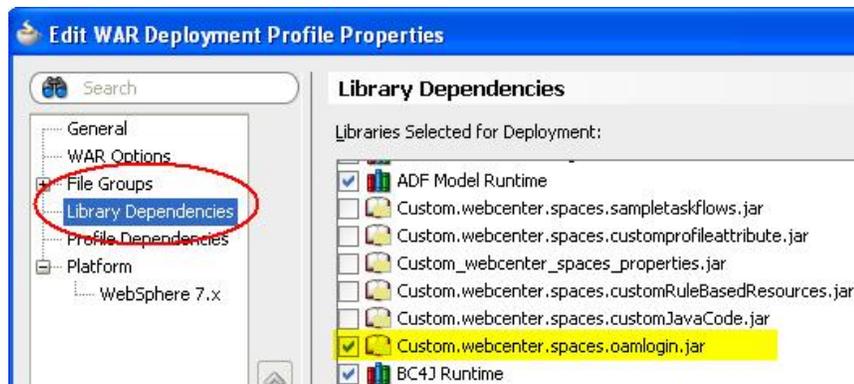
The following steps, describe how to do this using the **SampleWebCenterSpacesExtension** workspace:

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

The OAM login code is located under **OAMLogin\Application Sources\custom.webcenter.spaces\view\OAMBackingBean.java**.



2. To deploy the sample OAM login code, ensure that the **OAMLogin** project JAR file is selected in the deployment profile.



### 3. Deploy the sample OAM login code.

For details, see section [Building and Deploying a Customized WebCenter Spaces .WAR](#).

To verify the sample code, click the Login button for WebCenter Spaces and verify the single sign on experience. Clicking the Login button authenticates users against OAM SSO. For the authentication process to work:

- The Login task flow must be on a JSPX page. For details, see *Step 3 Ensure the WebCenter Spaces Login Page is JSPX*.
- OAM must be configured on the WebLogic Server instance running WebCenter Spaces and WebCenter Spaces must be configured to use Single Sign-On with OAM. For details, *Step 4 Configure Oracle Access Manager (OAM) SSO*.

### Step 3 Ensure the WebCenter Spaces 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 WebCenter Spaces Welcome page that displays to unauthenticated users is a JSP page (LandingGateway.jsp).

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

- Admin>Pages>System Pages>WebCenter Welcome Page>Customize>Save

**Note:** You do not have to do this step if you have disabled public access to WebCenter Spaces. In this case, a Login JSPX page displays in lieu of the Welcome page:

- Disable public access to WebCenter Space:  
Admin>Security>Roles>View Application>Public User>Disable

### Step 4 Configure Oracle Access Manager (OAM) SSO

The sample OAMLogin project requires Oracle Access Manager (OAM) 10.1.4.3 and WebCenter Spaces 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 WebCenter Spaces is running. This section describes how:

Step A: Setting Up OAM SSO

Step B: Installing OAM SDK

Step C: Configuring ObAccessClient.xml for OAM SDK

Step D: Setting WebLogic Server Environment Variables

Step E: Accessing OAM SDK Initialization and Shutdown

### Step A: Setting Up OAM SSO

Oracle Access Manager (OAM) 10.1.4.3 configuration is documented in the [Administrator's Guide for Oracle WebCenter](#). The “Configuring Single Sign-On“ chapter describes how to configure SSO for WebCenter Spaces and related applications such as SOA, UCM, using an out-of-the-box SSO configuration file.



*Administrator's Guide for Oracle WebCenter* is available from Oracle Technology Network:

[http://download.oracle.com/docs/cd/E17904\\_01/webcenter.1111/e12405/wcadm\\_security\\_sso.htm](http://download.oracle.com/docs/cd/E17904_01/webcenter.1111/e12405/wcadm_security_sso.htm)

In addition, you must configure your installation-specific information in the WebCenter policy domain created. The steps are summarized as follows (and assume OAM and 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 WebCenter 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 WebGates 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/E17904\\_01/doc.1111/e15478/toc.htm](http://download.oracle.com/docs/cd/E17904_01/doc.1111/e15478/toc.htm)

Configure resources for your WebCenter Spaces application. For example:

- /webcenter/adfAuthentication
- /webcenter - include in **Default Public Policy**

4. Install WebGate on OHS.
5. Configure OIDAAuthenticator 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 B: Installing OAM SDK

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

- Download Oracle Access Manager SDK from OTN:  
<http://www.oracle.com/technetwork/middleware/ias/downloads/101401-099957.html>
- Follow the SDK installation documentation located on OTN:  
[http://download.oracle.com/docs/html/E12491\\_01/sdkinst.htm](http://download.oracle.com/docs/html/E12491_01/sdkinst.htm)

### Step C: 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 D: Setting WebLogic Server Environment Variables**

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

- **OBACCESS\_INSTALL\_DIR**

```
OBACCESS_INSTALL_DIR="/scratch/<myinstall>/oam/sdk/AccessServerSDK"
```

```
export OBACCESS_INSTALL_DIR
```

```
Include jobaccess.jar in 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
```

```
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 E: Accessing OAM SDK Initialization and Shutdown**

To use the Access Manager SDK, you must invoke the following startup and shutdown methods on `ObConfig`:

- `OAMStartup.java`
- `OAMShutdown.java`

Both these startup and shutdown listener classes are packaged in `webcenter_oam.jar`, available with the sample at:

<UnzipDir>/SampleWebCenterSpacesExtensions/OAMLogin/lib

Oracle recommends that you incorporate these calls in your application server startup and shutdown listeners as follows:

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

For example, if the listeners are packaged in  
<UnzipDir>/SampleWebCenterSpacesExtensions/OAMLogin/lib,  
include the following **WEBLOGIC\_CLASSPATH** entry in <install  
location>/wlserver\_10.3/common/bin/commEnv.sh:

```
WEBLOGIC_CLASSPATH="${WEBLOGIC_CLASSPATH}${CLASSPATHSEP}/<UnzipDir/Sa
mpleWebCenterSpacesExtensions/OAMLogin/lib/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 `custom.webcenter.spaces.view.oamconfig.OAMConfigStartup`.
- Shutdown Listener- `OAMConfigShutdown`  
The class is packed in `custom.webcenter.spaces.view.oamconfig.OAMConfigShutdown`.

Ensure that you select **WC\_Spaces** as the target server.

## Step 5 Validate the Custom OAM Login Code

To verify the custom OAM login code:

1. In your browser, enter the WebCenter Spaces 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.  
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

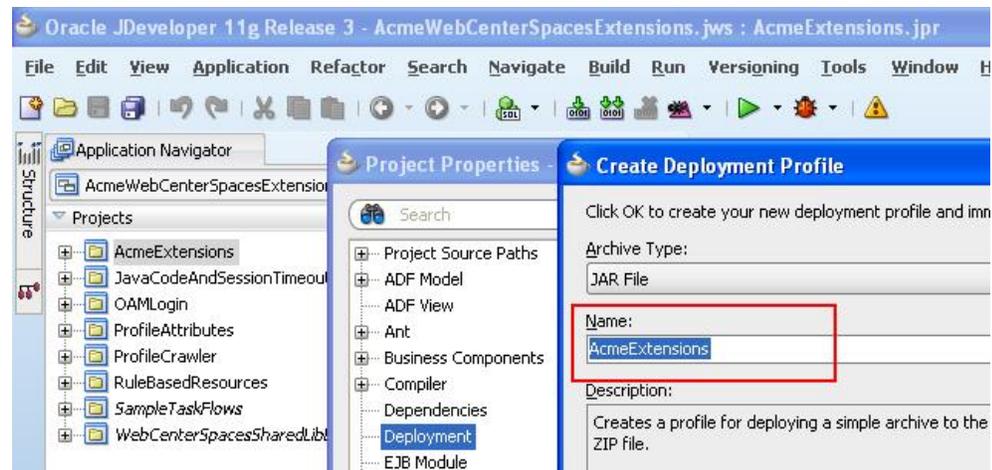
This section describes how to create a new project in the SampleWebCenterSpacesExtensions workspace and deploy your project code to a WebCenter Spaces shared library:

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.

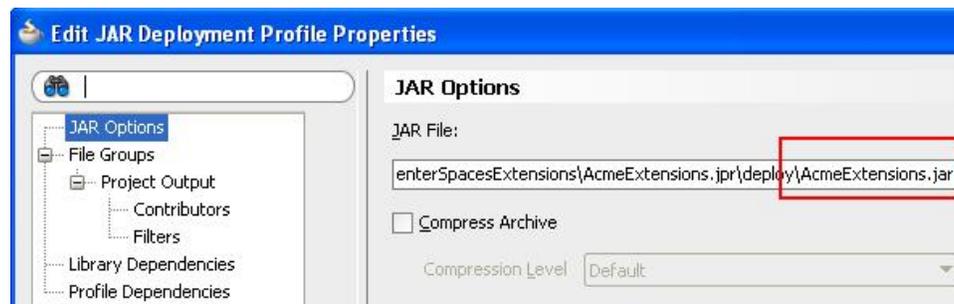
For example, **AcmeExtensions.jpr**.

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.

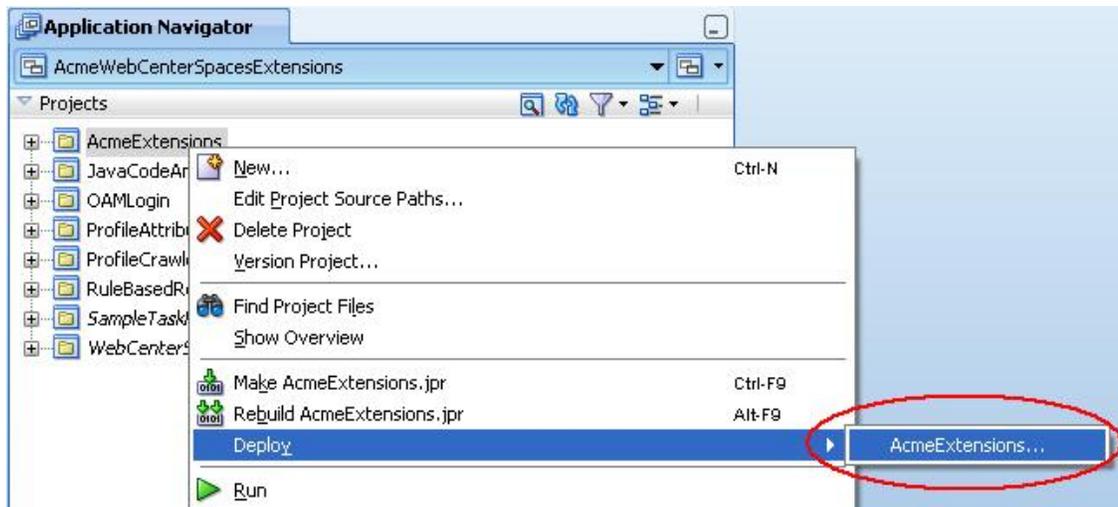
For example, AcmeExtensions.



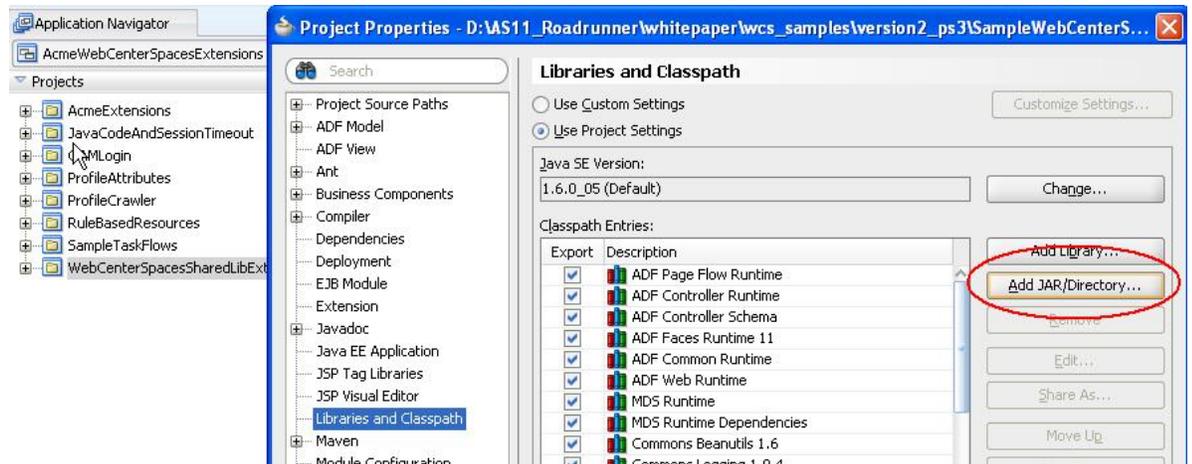
4. Select **JAR Options** to review or change the name of the JAR file generated by the deployment profile:



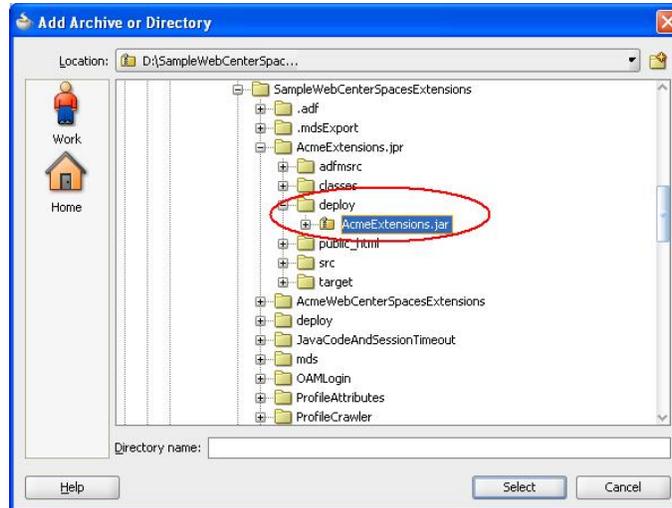
5. Right click your project folder and build the project JAR using the deployment profile.



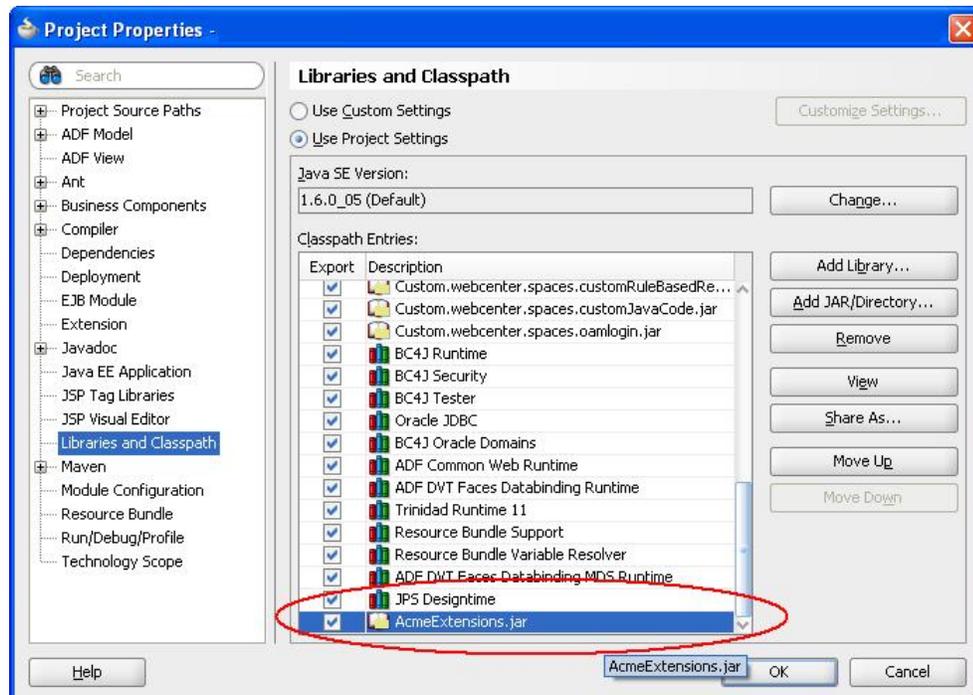
6. Include the project JAR file in the WebCenter Spaces shared library:
  - a) Right-click **WebCenterSharedLibraryExtensionProject > Project Properties**
  - b) Select **Libraries & Classpath**.
  - c) Select **Add JAR/Directory** to add your project JAR.



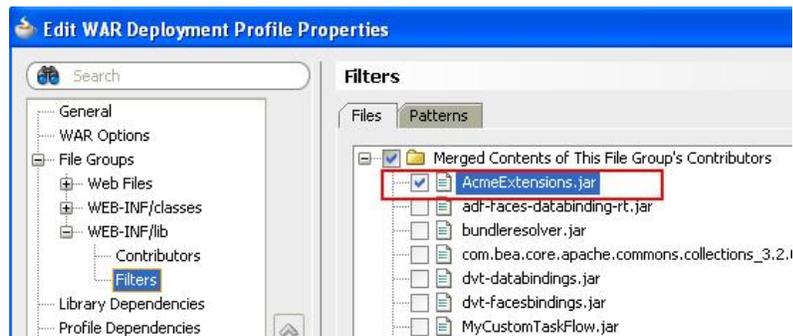
d) Select the JAR file.



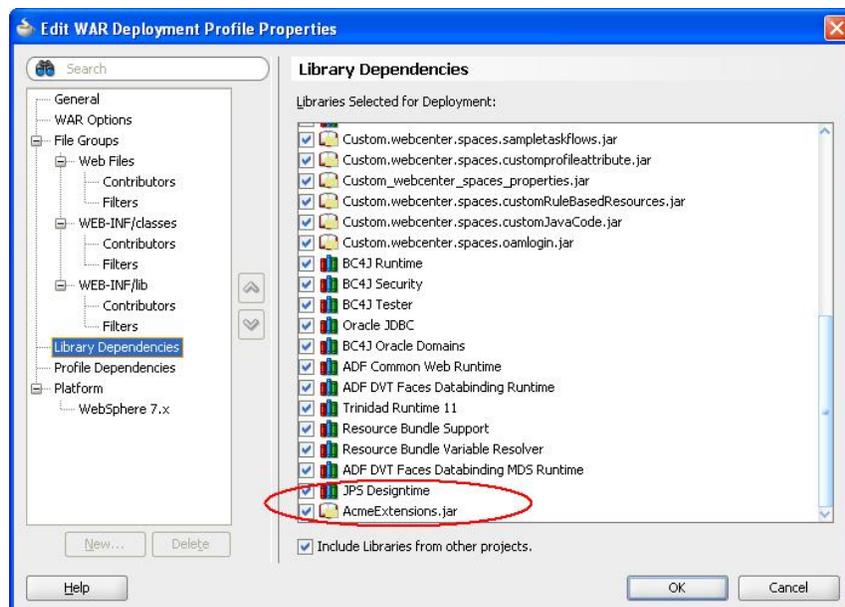
Your project's JAR is listed:



7. Edit the WebCenter 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 Customized WebCenter Spaces .WAR](#).

## Using WebCenter Spaces Extensions from an Earlier Release

You cannot automatically upgrade earlier versions of the WebCenter Spaces extension workspace to 11.1.1.4. 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 WebCenter Spaces 11.1.1.4 as follows:

1. Upgrade the WebCenter domain, WebCenter permissions and WebCenter Spaces, as documented in *Oracle Fusion Middleware Patching Guide (11.1.1.4)*, available on Oracle Technology Network:

[http://download.oracle.com/docs/cd/E17904\\_01/doc.1111/e16793/toc.htm](http://download.oracle.com/docs/cd/E17904_01/doc.1111/e16793/toc.htm)

**Very Important:** When you upgrade the WebCenter domain from 11.1.1.2 or 11.1.1.3 to 11.1.1.4, 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 WebCenter 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 WebCenter Spaces extension workspace, continue with step 3.
2. Use `ExtendWebCenterSpaces.jws` (11.1.1.2. or 11.1.1.3) to deploy customizations in the `custom.webcenter.spaces.war` shared library to your WebCenter Spaces 11.1.1.4 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 WebCenter Spaces 11.1.1.4 shared library dependency list.
  - a) If you have not done so already, download the WebCenter Spaces (11.1.1.4) development ZIP from Oracle Technology Network:

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

- b) Open the DesignWebCenterSpaces.jws (11.1.1.4) workspace.
- c) Navigate to: **WebCenterSpacesExtensionLibrary\Application Sources\WebCenterSpacesExtensionLibrary\public\_html\WEB-INF\weblogic.xml**
- d) Include **custom.webcenter.spaces** WAR file in the WebCenter Spaces 11.1.1.4 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 WebCenter Spaces Shared Library List” in *Developer’s Guide for Oracle WebCenter*.



*Developer’s Guide for Oracle WebCenter* is available from Oracle Technology Network:

[http://download.oracle.com/docs/cd/E17904\\_01/webcenter.1111/e10148/jpsdg\\_wcsres.htm](http://download.oracle.com/docs/cd/E17904_01/webcenter.1111/e10148/jpsdg_wcsres.htm)

### Moving Previous Extensions to `SampleWebCenterSpacesExtensions.jws`

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

The latest release of WebCenter 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 WebCenter Spaces built-in tools instead of JDeveloper.

## Conclusion

This document describes how to extend WebCenter Spaces 11.1.1.4.0 to meet the needs of your user base. For more information on WebCenter Spaces, see the *User’s Guide for Oracle WebCenter Spaces*, available on Oracle Technology Network at:

[http://download.oracle.com/docs/cd/E17904\\_01/webcenter.1111/e10149/toc.htm](http://download.oracle.com/docs/cd/E17904_01/webcenter.1111/e10149/toc.htm)



Using WebCenter Spaces Extension Samples  
(11.1.1.4.0)  
August 2011  
Author: Rosie Harvey, Sunil Franklin  
Contributing Authors: Manish Devgan, Rohit  
Kulkarni, Vaibhav Gupta

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

Copyright © 2011, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.