Oracle Utilities
Customer Self Service

Whitepaper: Customizing and Extending the OUCSS Custom Portal

Release 2.1.0 Service Pack 2

E36186-07

November 2015

ORACLE

Oracle Utilities Customer Self Service Whitepaper: Customizing and Extending the OUCSS Custom Portal

Release 2.1.0 Service Pack 2

E36186-07

November 2015

Copyright © 2011, 2015, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Table of Contents

Overview 5

Additional Resources 5

Abbreviations 5

Preparation and Procedures 6

Software Requirements 6

Pre-Installation Tasks 6

Assumptions 6

Portal Extension 7

About Security 7

Adding New Taskflows and Pages 8

Accessing User Account Details 11

Adding Support for a New Language 16

Deploying the *extend.oucss.portal* as a Shared Library 19

Parallel Development on the Portal Extension 24

Deploying CM ExtendOUCSSPortal Projects 31

Parallel Customization of Existing Taskflows 33

WebCenter Portal Application Local Instance 36

Portal Rebranding 55

Customizing the Page Template 55

Customizing an Existing Page Template 55

Exporting Page Templates 60

Importing Page Templates 62

Customizing the Navigation Model 63

Customizing Provided/Creating New Navigation Models 63

Accessing Pages from the Extended OUCSS Portal 68

Exporting an Existing Navigation Model 70
Importing a Navigation Model 71
Customizing the Resource Catalog 73
Customizing Skins 79
Portal Labels 85

Chapter 1

Overview

This whitepaper describes the steps required to customize and re-brand OUCSS Custom Portals.

Additional Resources

Further information is available in the *Developing Portals with Oracle WebCenter Portal (11.1.1.8.0)*, available on the OTN at http://docs.oracle.com/cd/E29542_01/webcenter.1111/e63259/toc.htm.

> **Note**: This document and related documentation is subject to revision and updating. For the most recent versions of documentation relating to this produce, as well as information on functionality and known issues for other Oracle products that may be required for installation and proper functionality of this product, check the Oracle Utilities Customer Self Service section of the Oracle Utilities Documentation area on the Oracle Technology Network (OTN) web site (http://www.oracle.com/technetwork/apps-tech/utilities/documentation/index.html).

Abbreviations

CCB - Oracle Utilities Customer Care and Billing

MDM - Oracle Utilities Meter Data Management

DDL – Data Definition Language

MDS – Metadata Services

OUCSS – Oracle Utilities Customer Self Service

Portal – Oracle WebCenter Portal Framework

Rest – Representational State Transfer

Chapter 2

Preparation and Procedures

Software Requirements

Please refer to the *Oracle Utilities Customer Self Service Installation Guide* for product-specific versions and the current patch level.

Pre-Installation Tasks

The following tasks should be completed before you can customize OUCSS Portal.

- Ensure that your Oracle JDeveloper version is 11gR1 PS6 (11.1.1.7.0).
- Ensure that the WebCenter Extension bundle (11.1.1.8.0) is enabled in JDeveloper.
- Ensure that your OUCSS (custom) Portal has been deployed and is running.

Assumptions

- The OUCSS Portal is built using the "WebCenter Portal Framework" application template.
- The Custom Portal is deployed onto an Oracle WebLogic (10.3.6) server.
- The OUCSS Schema is created and populated with seed data. For running the taskflows, a data source with jndi name **jdbc/OUCSSDS** is also expected in the target application server.
- For taskflows that require the sending of emails (e.g., Registration, Invite, etc.), a mail session with the jndi name **mail/OUCSS** is also expected on the target application server.

Note: The following information is not within the scope of this document:
- Setup of database and MDS or other schemas required by the custom portal.
- Detailed security of the custom portal to restrict access to taskflows, page templates, navigation models, and/or portal pages.

Portal Extension

The OUCSS Portal can be extended by creating custom taskflows and pages using the **ExtendOUCSSPortal** workspace included in ExtendOUCSSPortal.zip. The ExtendOUCSSPortal workspace includes three projects: **OUCSSPortalADFLibrary**, **OUCSSRestExtensionLibrary** and **OUCSSPortalSharedLibrary**.

The OUCSSPortalADFLibrary project is used to add portal artifacts (taskflows and pages) and the OUCSSRestExtensionLibrary project is used to extend OUCSS Rest Services and adding custom services. The OUCSSPortalSharedLibrary project is used for redeploying the **extend.oucss.portal** shared library.

Note: You can add pages at runtime from the **Administrator > Pages** screen. For a smooth OUCSS upgrade, Oracle recommends that you create pages using the ExtendOUCSSPortal approach.

About Security

The OUCSS Portal is secured and requires permissions to be granted to render custom taskflows and pages. The OUCSS Portal's security is implemented with required permissions to allow rendering of taskflows deployed using the **OUCSSPortalADFLibrary** project.

For this security design to work the taskflow definition should be created under

```
<<ExtensionWorkSpace>>/OUCSSPortalADFLibrary/public_html/WEB-INF/oracle/ugbu/ss/custom/secure
```

and/or

```
<<ExtensionWorkSpace>>/OUCSSPortalADFLibrary/public_html/WEB-INF/oracle/ugbu/ss/custom/public
```

Custom Pages and Taskflow fragments should also be created under

```
<<ExtensionWorkSpace>>/OUCSSPortalADFLibrary/public_html/oracle/ugbu/ss/custom/pages/secure
```

and/or

```
<<ExtensionWorkSpace>>/OUCSSPortalADFLibrary/public_html/oracle/ugbu/ss/custom/pages/public
```

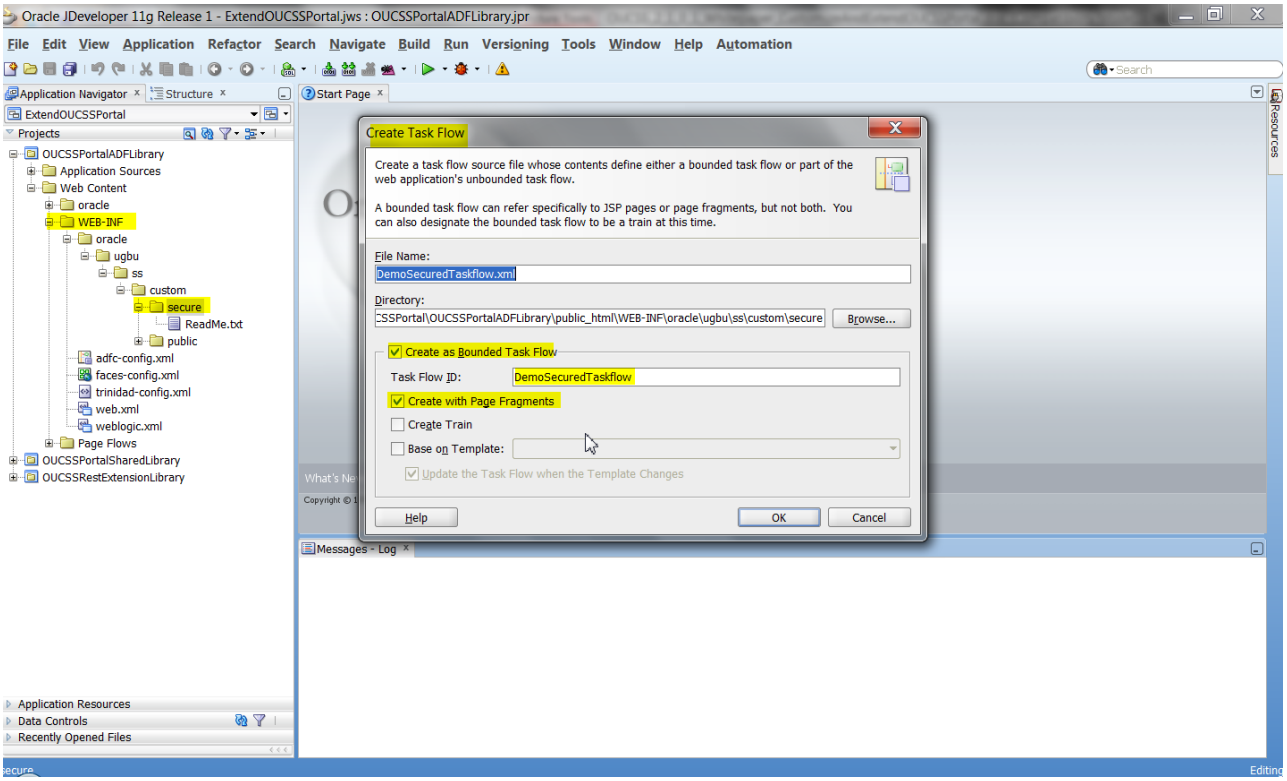
Use the following security criteria when creating taskflows:

- Create taskflows under `\WEB-INF\oracle\ugbu\ss\custom\secure` if access is to be restricted to authenticated/ logged-in users.
- Create taskflows under `\WEB-INF\oracle\ugbu\ss\custom\public` if the taskflows are to be accessible to all users (including public users).
- Create pages/fragments under `\oracle\ugbu\ss\custom\pages\secure` if the pages should be accessible only to authenticated/logged-in users.
- Create pages/fragments under `\oracle\ugbu\ss\custom\pages\public` if the pages should be accessible to all users (including public users).

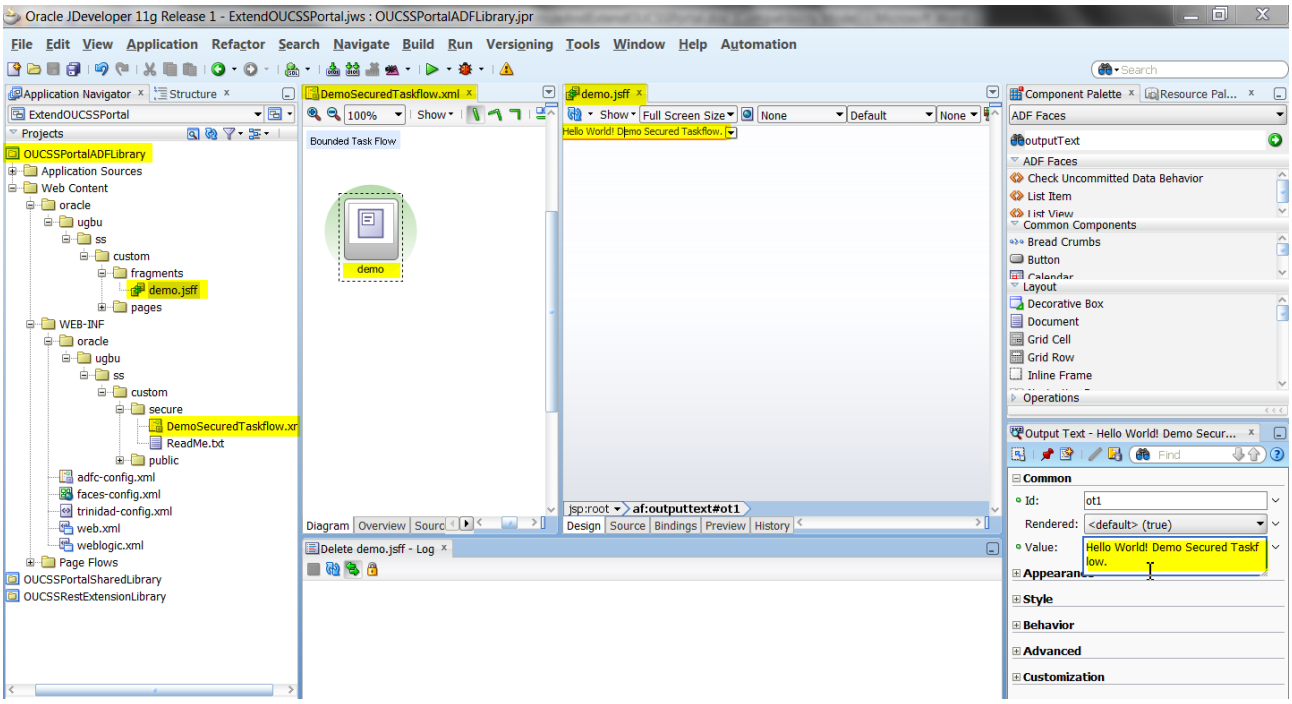
The OUCSS Portal Administrator can manage the pages and taskflows created under both secure and public folders.

Adding New Taskflows and Pages

1. Extract the contents of `ExtendOUCSSPortal.zip` and open `ExtendOUCSSPortal.jws` in JDeveloper.
2. Select the `public_html\WEB-INF\oracle\ugbu\ss\custom\secure` folder in the **OUCSSPortalADFLibrary** project and add a new ADF Taskflow.



3. For the purposes of this demo, we'll create a secure taskflow containing one **view/jsff** (`demo.jsff`) that displays, "Hello World! Demo Secured Taskflow".

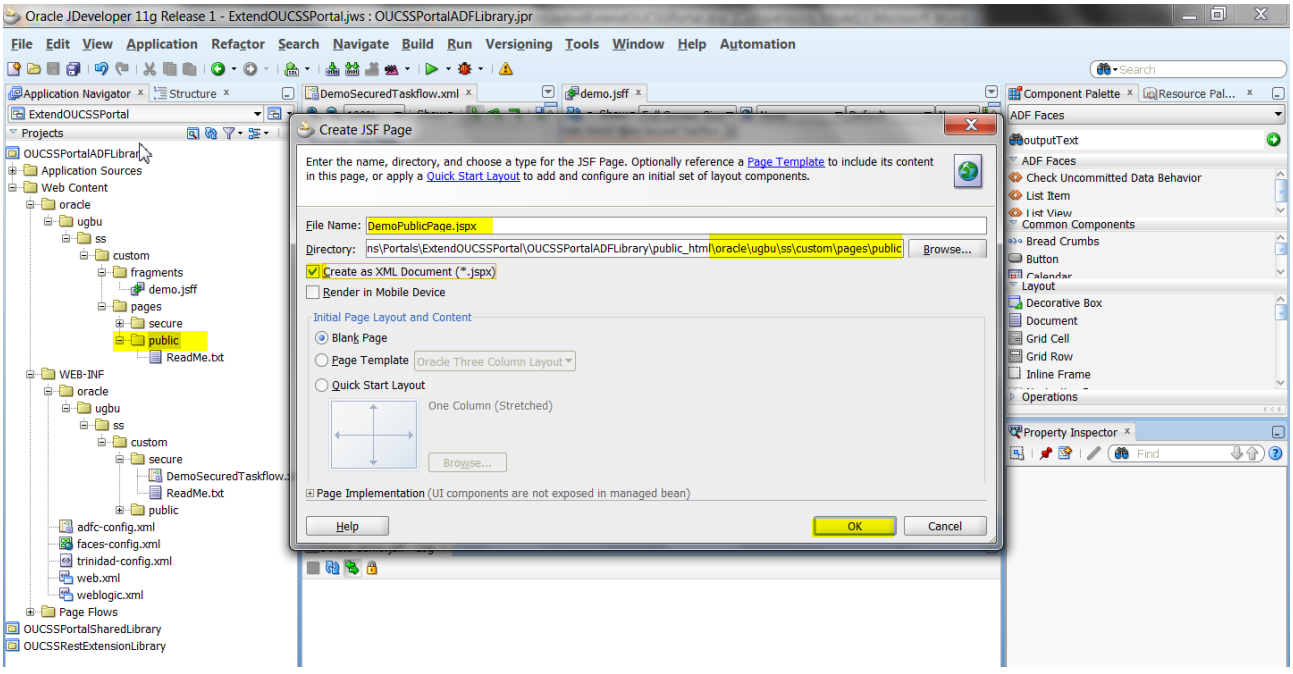


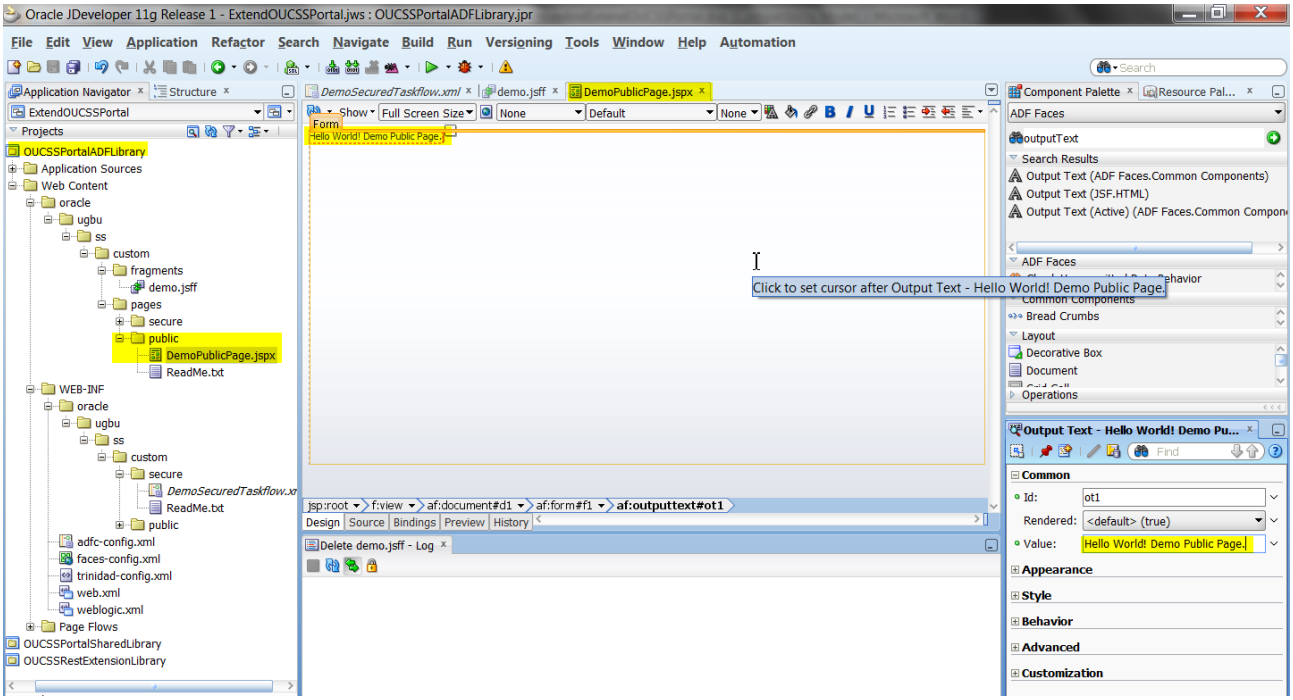
4 Repeat step 2 and 3 to create other secured and public taskflows.

5 Select the `/public_html/oracle/ugbu/ss/custom/public` folder in the **OUCSSPortalADFLibrary** project and add a new **Page** (jspx).

For the purposes of this demo, a page named `DemoPublicPage.jspx` is created that contains the text, "Hello World! Demo Public Page".

Note: For security to work, always add a **Page Definition** for all pages added in an **ExtendOUCSSPortal** project. If the page definition is not added, the page will not render in the portal.





6 The newly-created page should reference a page template and page editor to enable runtime customization. To create the references:

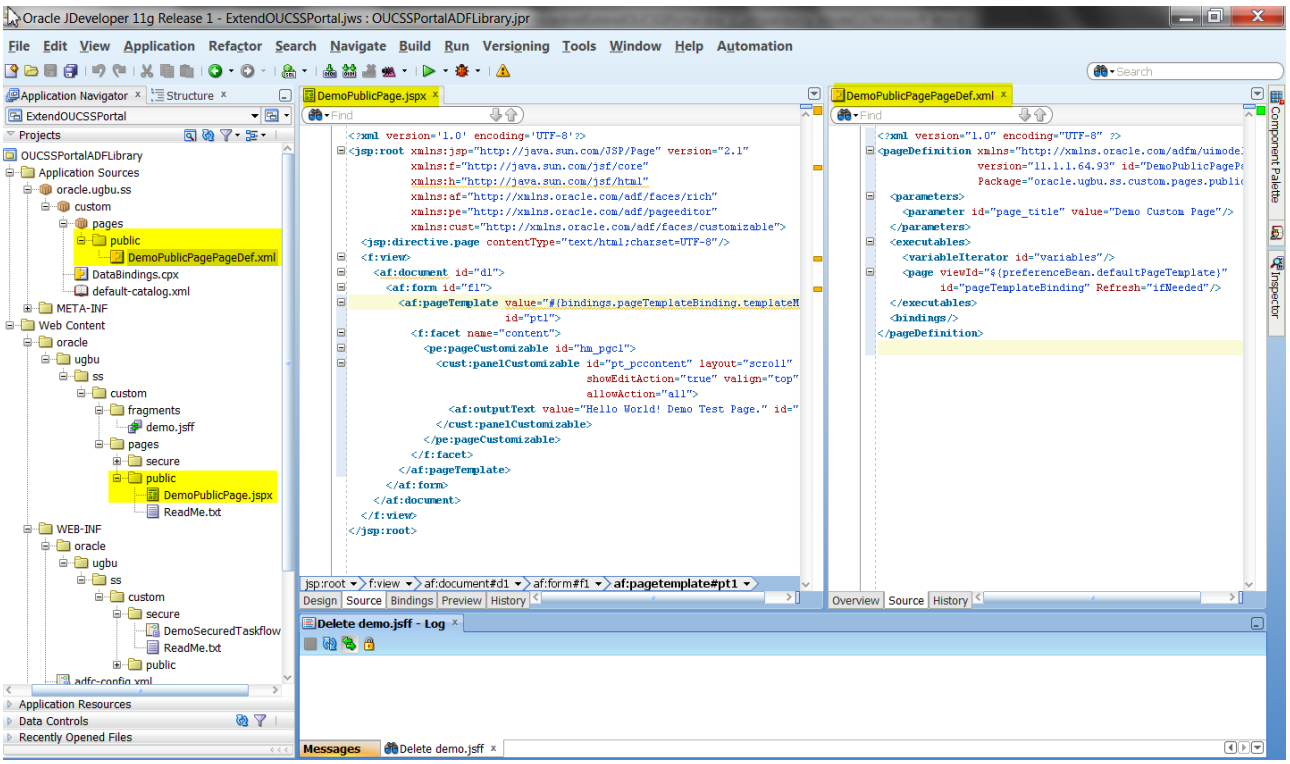
- Modify the JSPX as follows:

```
<?xml version='1.0' encoding='UTF-8'?>
<jsp:root xmlns:jsp="http://java.sun.com/JSP/Page" version="2.1"
          xmlns:f="http://java.sun.com/jsf/core"
          xmlns:h="http://java.sun.com/jsf/html"
          xmlns:af="http://xmlns.oracle.com/adf/faces/rich"
          xmlns:pe="http://xmlns.oracle.com/adf/pageeditor"
          xmlns:cust="http://xmlns.oracle.com/adf/faces/customizable">
  <jsp:directive.page contentType="text/html;charset=UTF-8"/>
  <f:view>
    <af:document id="d1">
      <af:form id="f1">
        <af:pageTemplate value="#{bindings.pageTemplateBinding.templateModel}"
                         id="pt1">
          <f:facet name="content">
            <pe:pageCustomizable id="hm_pgc1">
              <cust:panelCustomizable id="pt_pccontent" layout="scroll"
                                      showEditAction="true" valign="top"
                                      allowAction="all">
                <af:outputText value="Hello World! Demo Test Page." id="ot1"/>
              </cust:panelCustomizable>
            </pe:pageCustomizable>
          </f:facet>
        </af:pageTemplate>
      </af:form>
    </af:document>
  </f:view>
</jsp:root>
```

- Modify the Page Def as follows:

```
<?xml version="1.0" encoding="UTF-8" ?>
<pageDefinition xmlns="http://xmlns.oracle.com/adfm/uimodel"
                version="11.1.1.64.93" id="DemoPublicPagePageDef"
```

```
                Package="oracle.ugbu.ss.custom.pages.public">
  <parameters>
    <parameter id="page_title" value="Demo Custom Page"/>
  </parameters>
  <executables>
    <variableIterator id="variables"/>
    <page viewId="${preferenceBean.defaultPageTemplate}"
          id="pageTemplateBinding" Refresh="ifNeeded"/>
  </executables>
  <bindings/>
</pageDefinition>
```



7 Repeat step 4 and 5 to create other secured and public pages.

8 After deployment of exted.oucss.portal shared library,

9 Review the Configure the Resource Catalog section of this document to consume the taskflow in OUCSS Portal.

10 Review the Access Pages from Extend OUCSS Portal section of this document to access page in OUCSS Portal.

Accessing User Account Details

As part of OUCSS Portal Install, libraries are installed that contain the taskflows and java classes. These libraries are packaged as .war files under <<webcenter_WSS_zip>>\Install\application\. The following table summarizes the libraries

WAR File Name	Deployed As	Description
OUCSS_Common_Extension.war	com.oracle.ugbu.ss.lib (11.1.1, 11.1.1.9.0)	The common library contains OUCSS APIs and common code.
OUCSS_Residential_Extension.war	com.oracle.ugbu.ss.residential.lib (11.1.1, 11.1.1.9.0)	The residential library contains Core OUCSS taskflows and modules.
OUCSS_Commercial_Extension.war	com.oracle.ugbu.ss.commercial.lib (11.1.1,	The commercial library contains taskflows

11.1.1.9.0) related to Sets and Multi-Account operations.

The com.oracle.ugbu.ss.lib deployed contains jars of OUCSS API and other shared java classes. By adding jars from OUCSS_Common_Extension.war to OUCSSPortalADFLibrary projects will allow access to the OUCSS API and other artifacts in custom pages and taskflow views.

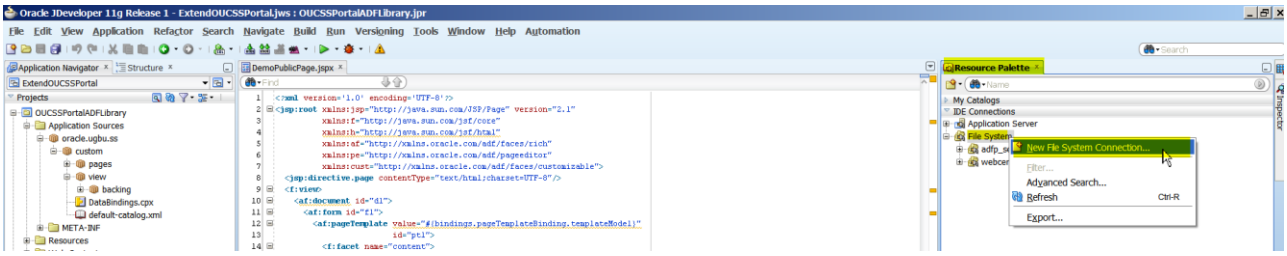
In the following example, a managed bean is created and accessed using the Secured Taskflow. The managed bean will retrieve accounts of a logged-in user and display the list in the custom taskflow.

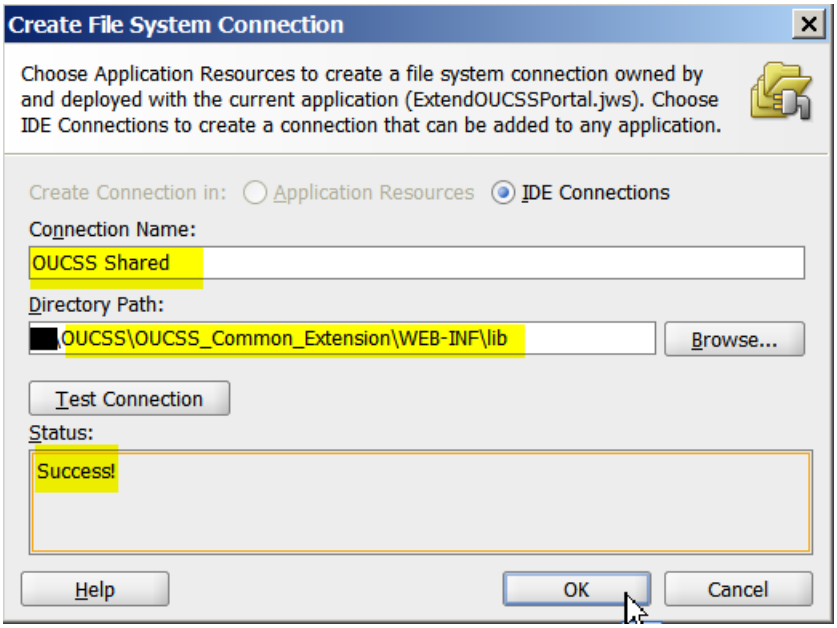
1. Unjar/Unzip the `OUCSS_Common_Extension.war` file from `<<webcenter_WSS_zip>>\Install\application\` into a folder that is accessible to JDeveloper.

Name	Date modified	Type	Size
oracle.ugbu.ss.shared.utilities.jar	3/25/2015 6:52 PM	Executable Jar File	262 KB
oracle.ugbu.ss.shared.model.jar	3/25/2015 6:52 PM	Executable Jar File	1,043 KB
oracle.ugbu.ss.shared.commercial.view.jar	3/25/2015 6:52 PM	Executable Jar File	24 KB
oracle.ugbu.ss.shared.commercial.model.jar	3/25/2015 6:52 PM	Executable Jar File	11 KB
oracle.ugbu.ss.shared.commercial.api.jar	3/25/2015 6:52 PM	Executable Jar File	5 KB
oracle.ugbu.ss.shared.api.jar	3/25/2015 6:52 PM	Executable Jar File	41 KB
oracle.ugbu.ss.base.acctcontextinfo.model.jar	3/25/2015 7:05 PM	Executable Jar File	79 KB

2. Unjar/unzip the `OUCSS_Residential_Extension.war` and/or **OUCSS_Commercial_Extension** wars from `<<webcenter_WSS_zip>>\Install\application\` into a folder that is accessible to JDeveloper.

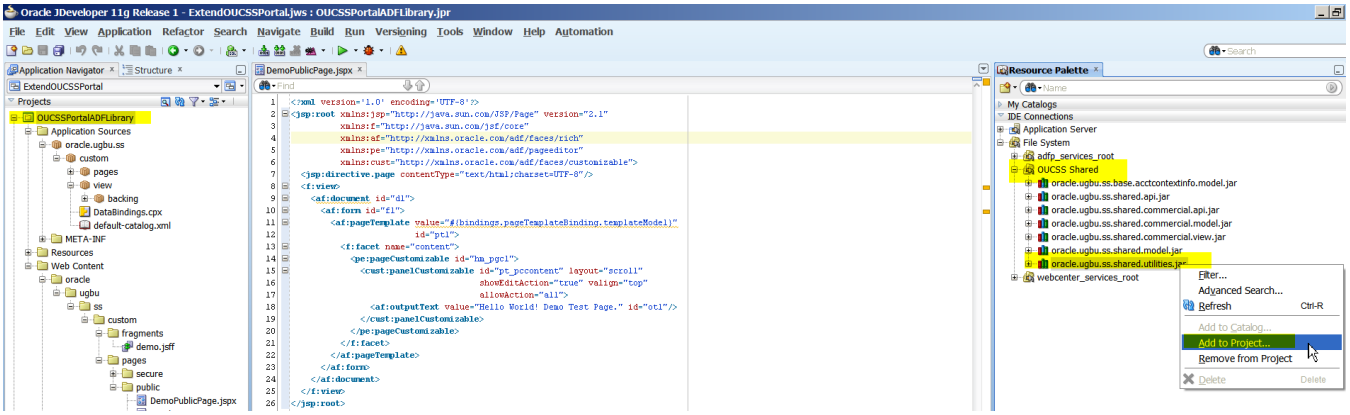
3. From JDeveloper's **Resource Palette**, create a **File System Connection** to the folder containing the jar files. For example, create a File System Connection named "OUCSS Shared" using the `<<Unzip_Folder>>\OUCSS_Extension\WEB-INF\lib` directory created in step 1.





4 Click **OK** and explore the OUCSS Shared connection to see the jars in the **Resource Palette**.

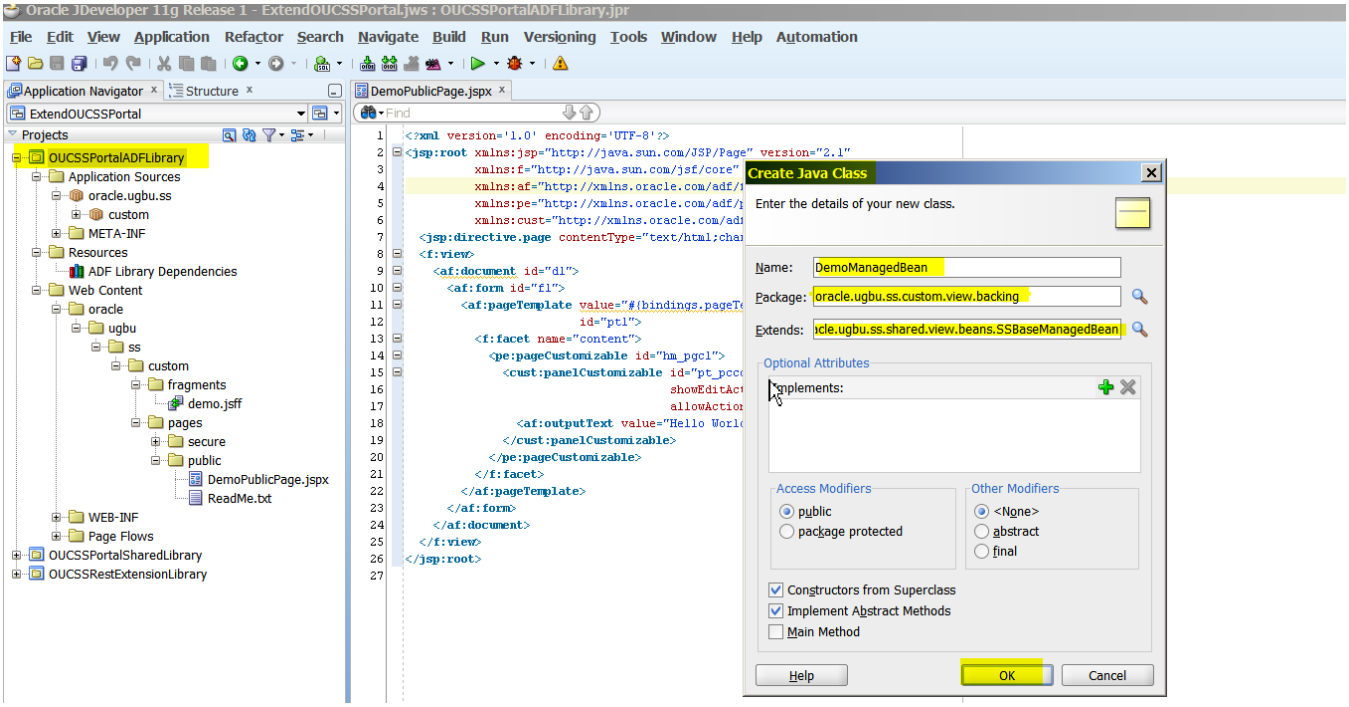
5 Make sure the **OUCSSPortalADFLibrary** project is selected in JDeveloper. Right-click `oracle.ugbu.ss.shared.utilities` under the **OUCSS Core Taskflows File System** connection, then select **Add to Project** from the context menu. If a confirmation dialog appears, click **Add Library**.



6 Repeat step 4 and add the `oracle.ugbu.ss.shared.api` library to the project.

7 **Save** all.

8 Create a Java class (`DemoManagedBean.java`, sample code below) and extend the `SSBaseManagedBean` class.



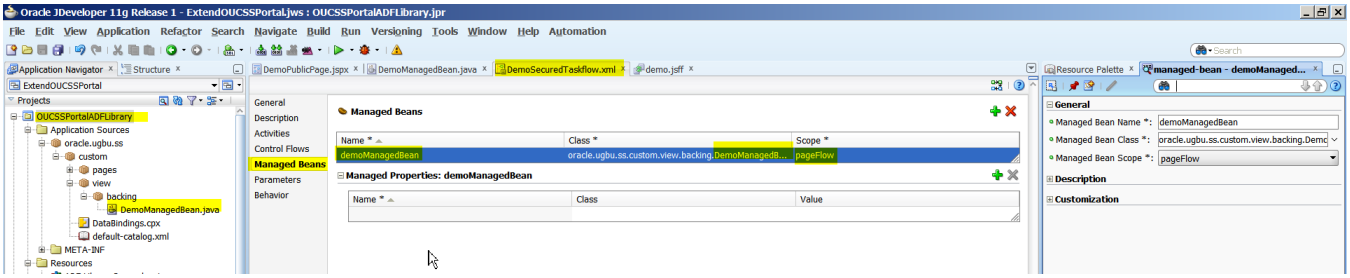
9 Use the following sample code to access the account details in the taskflow.

Note: The following code sample assumes that the custom taskflow or page will default the `userId` to the logged-in User. If you want to test the code without logging in, then override the user to a valid OUCSS User by calling `super.setWebUserId("<<valid_UserId>>);` in the constructor of the DemoManagedBean.

```
package oracle.ugbu.ss.custom.view.backing;

import com.oracle.ugbu.ss.api.model.account.OUCSSAccount;
import com.oracle.ugbu.ss.api.model.account.OUCSSAccountList;
import com.oracle.ugbu.ss.api.model.security.OUCSSUser;
import com.oracle.ugbu.ss.shared.view.beans.SSBaseManagedBean;

public class DemoManagedBean
  extends SSBaseManagedBean
{
  @SuppressWarnings("compatibility")
  private static final long serialVersionUID = 1L;

  String mLOB = null; //Set the LOB for account. Null will fetch accounts for all LOBS.

  public DemoManagedBean()
  {
    super();
    // Uncomment to Override the user for testing in JDev.
    //super.setWebUserId("<<valid_UserId");
  }

  /**
   * Method to get details about the current logged in user.
   * @return
   */
  public OUCSSUser getUser()
  {
    return super.getContext().getCurrentUser();
  }
```

```
  /**
   * Method to get list of accounts enrolled to logged in User.
   * @return
   */
  public OUCSSAccountList getUserAccountList()
  {
    OUCSSAccountList accountList =
      getContext().getCurrentUser().getUserAccounts(mLOB);
    return accountList;
  }

  /**
   * Method to get the currently selected account.
   * @return
   */
  public OUCSSAccount getCurrentAccount()
  {
    OUCSSAccount currentAccount =
      getContext().getCurrentUser().getCurrentUserAccount();
    return currentAccount;
  }
}
```

10 Set the DemoManagedBean class as a managed bean in the taskflow definition. For the purposes of this example, the scope used is **pageFlowScope**.



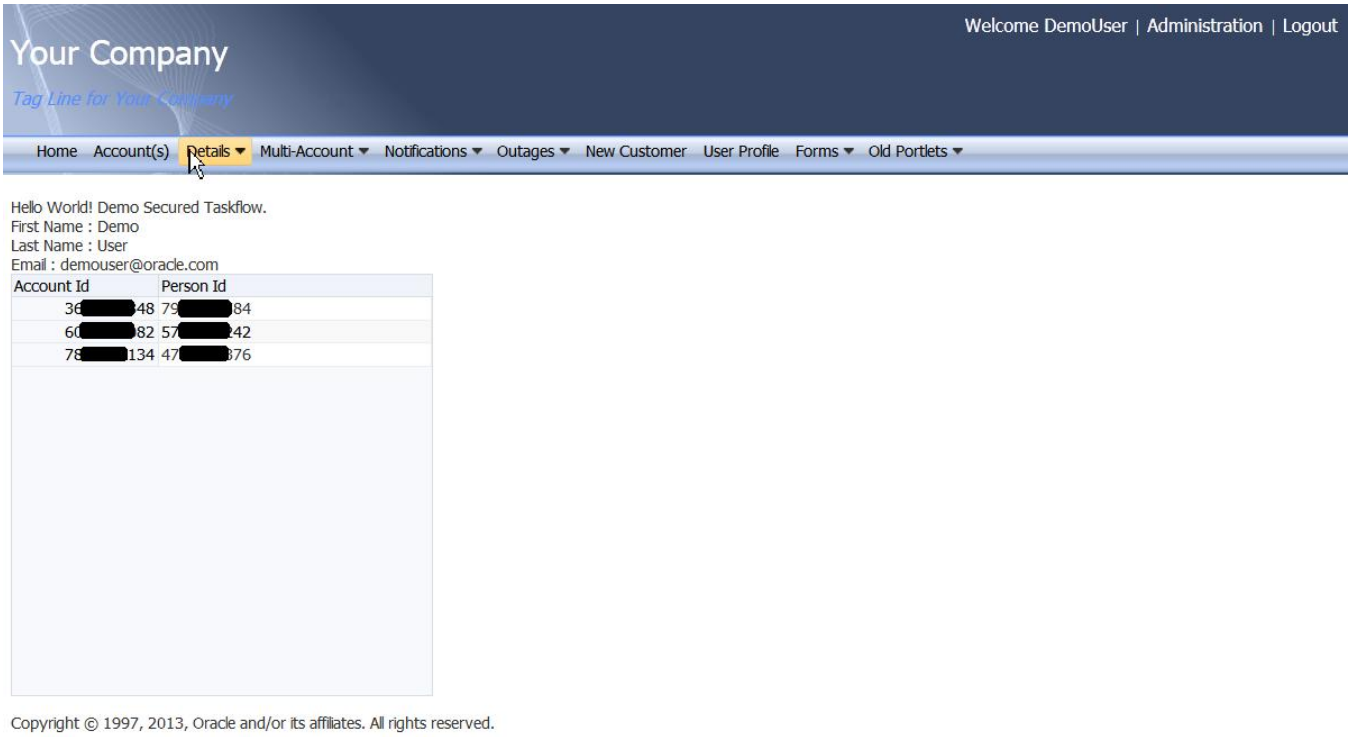
11 Update the previously-created `demo.jsff` to include user and account details by referencing the managed bean methods.

Sample JSPX code:

```
<?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"
          xmlns:f="http://java.sun.com/jsf/core">
  <af:panelBorderLayout id="pbl1">
    <f:facet name="top">
      <af:outputText value="Hello World! Demo Secured Taskflow." id="ot1"/>
    </f:facet>
    <af:panelGroupLayout layout="vertical" halign="start" id="pgl1">
      <af:outputText value="First Name : #{pageFlowScope.demoManagedBean.user.firstName}"
                     id="ot4"/>
      <af:outputText value="Last Name : #{pageFlowScope.demoManagedBean.user.lastName}"
                     id="ot5"/>
      <af:outputText value="Email : #{pageFlowScope.demoManagedBean.user.emailAddress}"
                     id="ot6"/>
      <af:table value="#{pageFlowScope.demoManagedBean.userAccountList}"
                var="row" id="t1" contentDelivery="immediate"
                columnStretching="last" rowSelection="none"
                summary="Account List" columnSelection="none"
                rowBandingInterval="1" shortDesc="Account List">
        <af:column headerText="Account Id" id="c1" rowHeader="true">
          <af:outputText value="#{row.key1}" id="ot2"/>
```

```
        </af:column>
        <af:column headerText="Person Id" id="c2">
          <af:outputText value="#{row.key2}" id="ot3"/>
        </af:column>
      </af:table>
    </af:panelGroupLayout>
  </af:panelBorderLayout>
</jsp:root>
```

12 Deploy the extension again and consume the taskflow in any page. The sample code produces the following result:



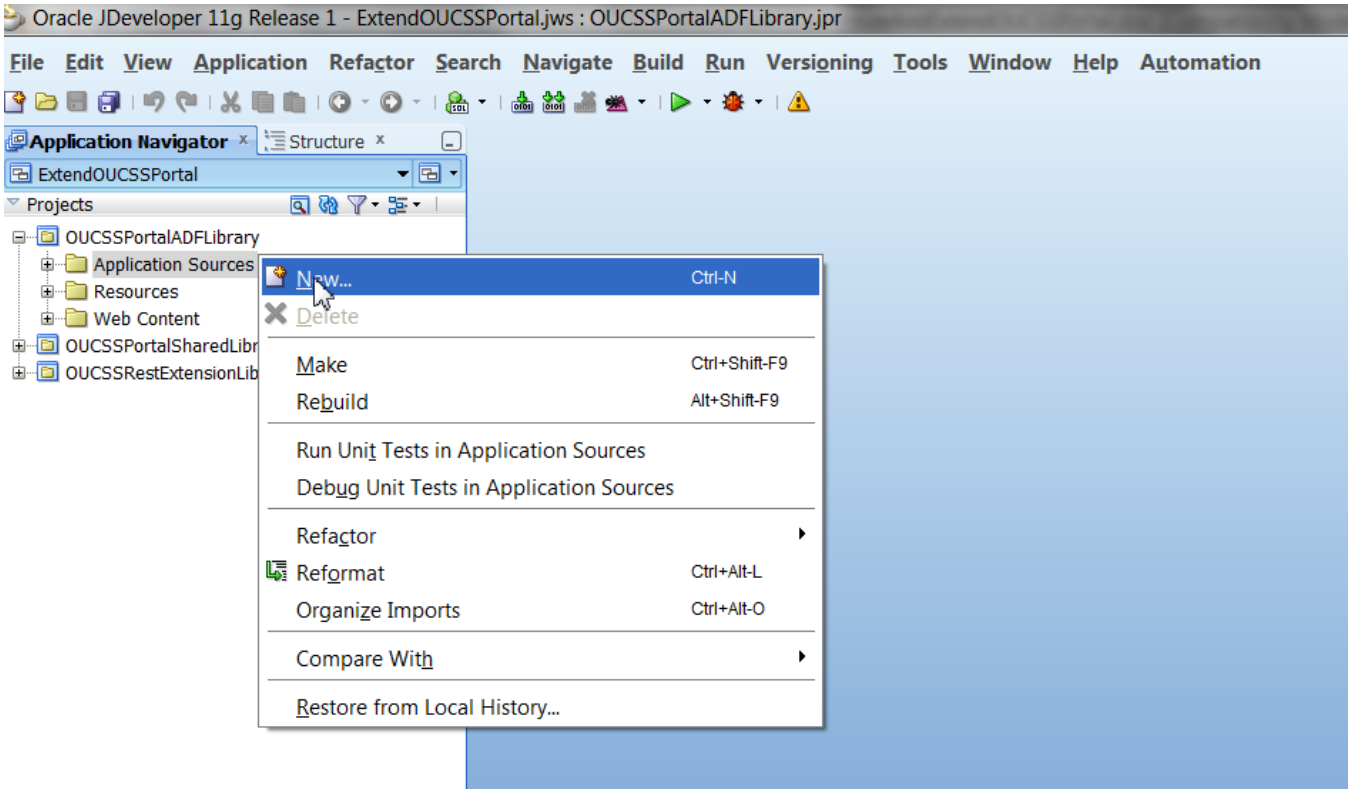
13 Review the following:

- Customizing the Resource Catalog section of this document to consume the taskflow in the OUCSS Portal.
- Accessing Pages from the Extended OUCSS Portal section of this document to access the page in the OUCSS Portal.

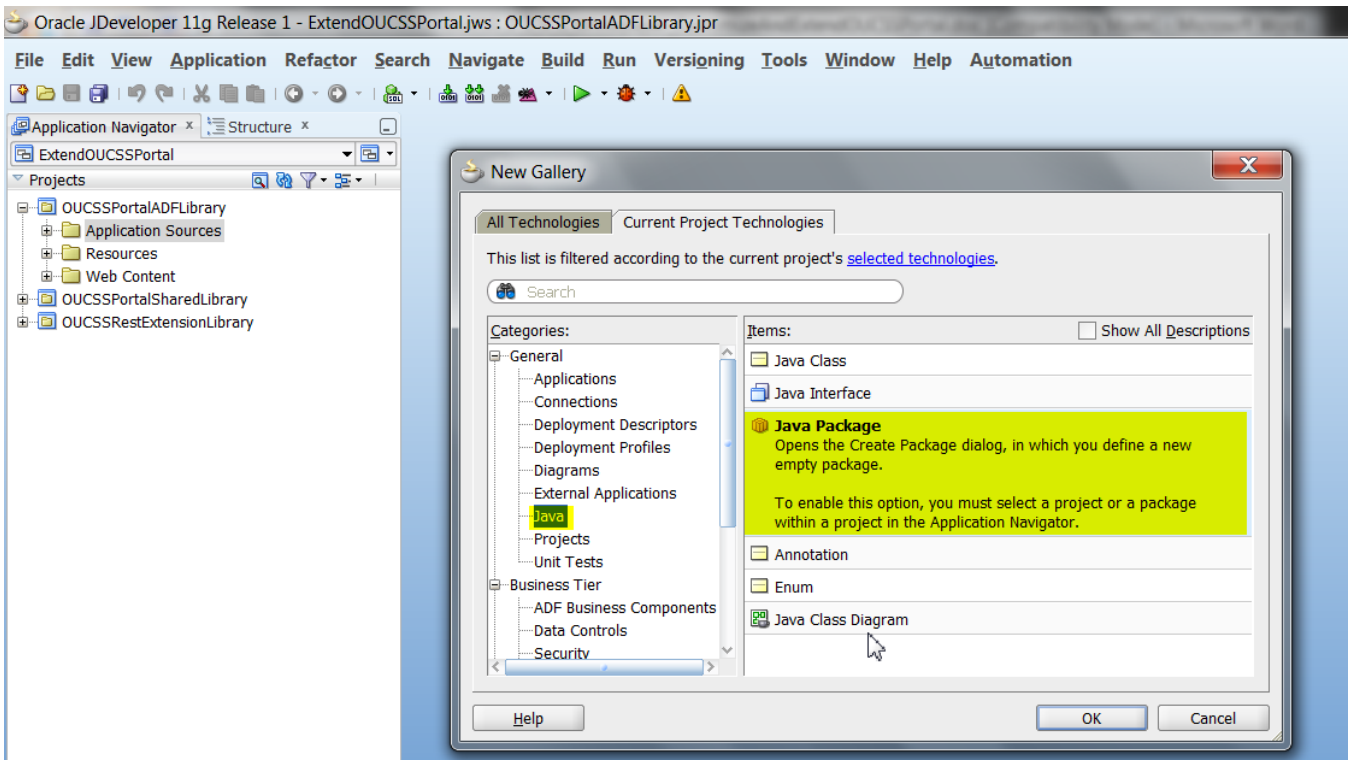
Adding Support for a New Language

OUCSS Portal by default is configured only to support English. In order to translate navigation links and page titles in OUCSS Portal, Portal bundles for the additional bundles are required to be created. In order to add a Portal Bundle for your supported language(s),

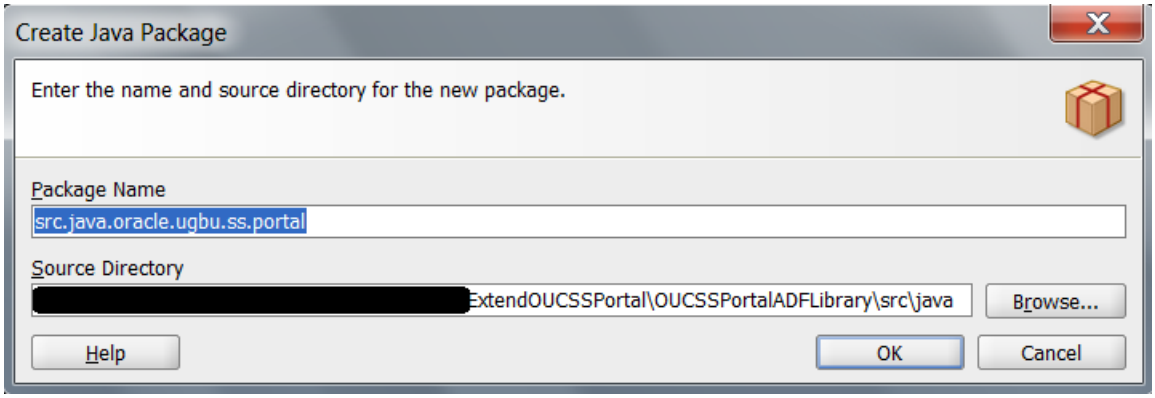
1 Create a new package in the Application Sources under the project OUCSSPortalADFLibrary. Right click on the 'Application Sources' directory and select 'New' from the pop-up menu.



2 Select Java under 'General' categories in the left panel. Select 'Java Package' in the right panel. Click 'Ok'.

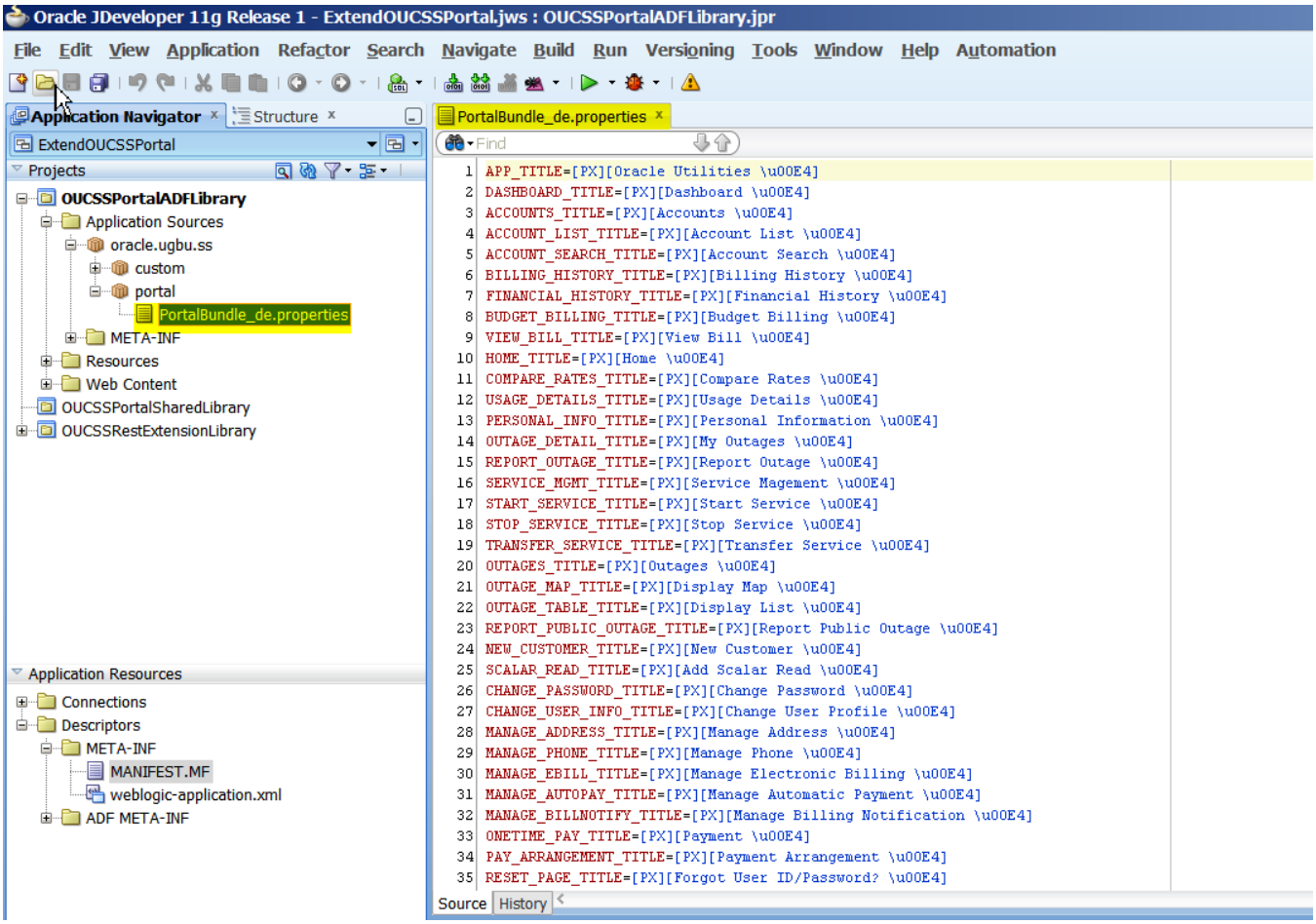


3 Enter `src.java.oracle.ugbu.ss.portal` as the **Package Name** and click **OK**.



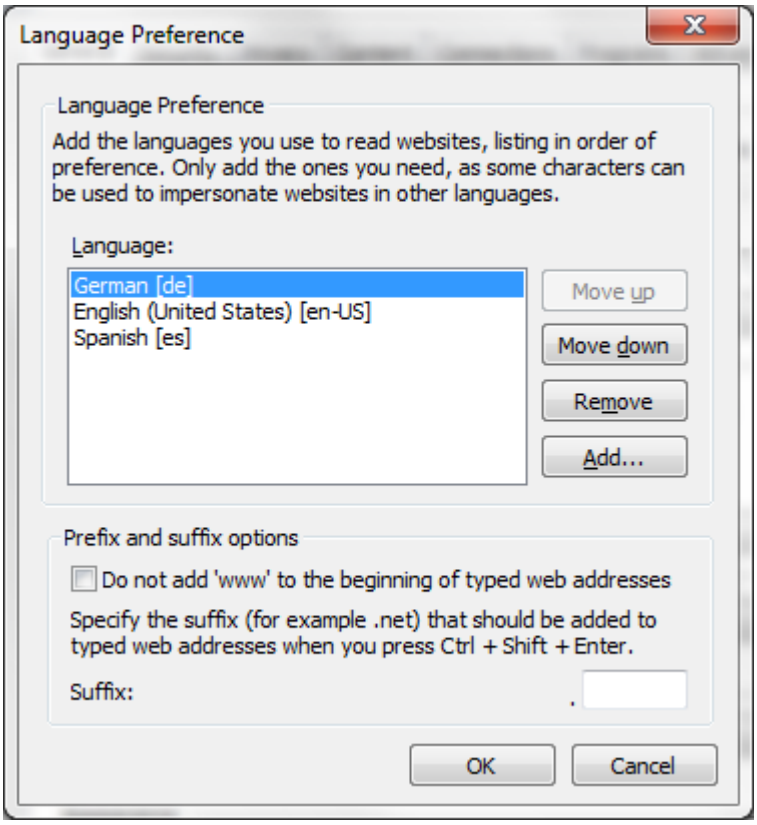
4 Create a new resource bundle with the appropriate locale and the name format **PortalBundle_<<locale>>.properties** under the package created in step 3. A separate bundle must be created for each supported locale.

For example, the following **PortalBundle_de.properties** bundle is created to support the German language/locale. See Appendix-A for the list of properties to be added to the Portal Bundle.



5 Deploy the extension library and restart the managed server to load the bundle.

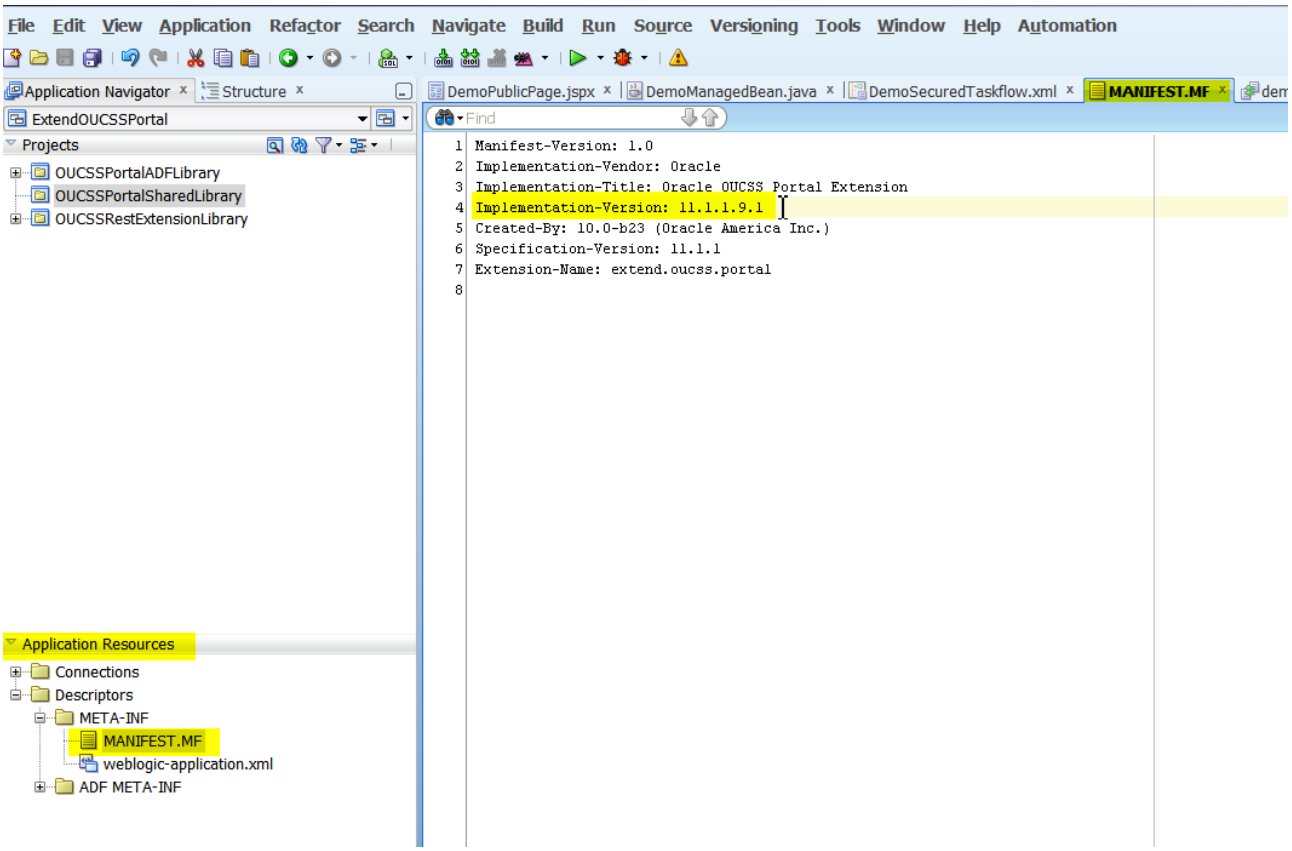
6 Set the browser language to the appropriate locale and access the portal to see the translated labels. In the following example, the browser locale is set to German.



Deploying the *extend.oucss.portal* as a Shared Library

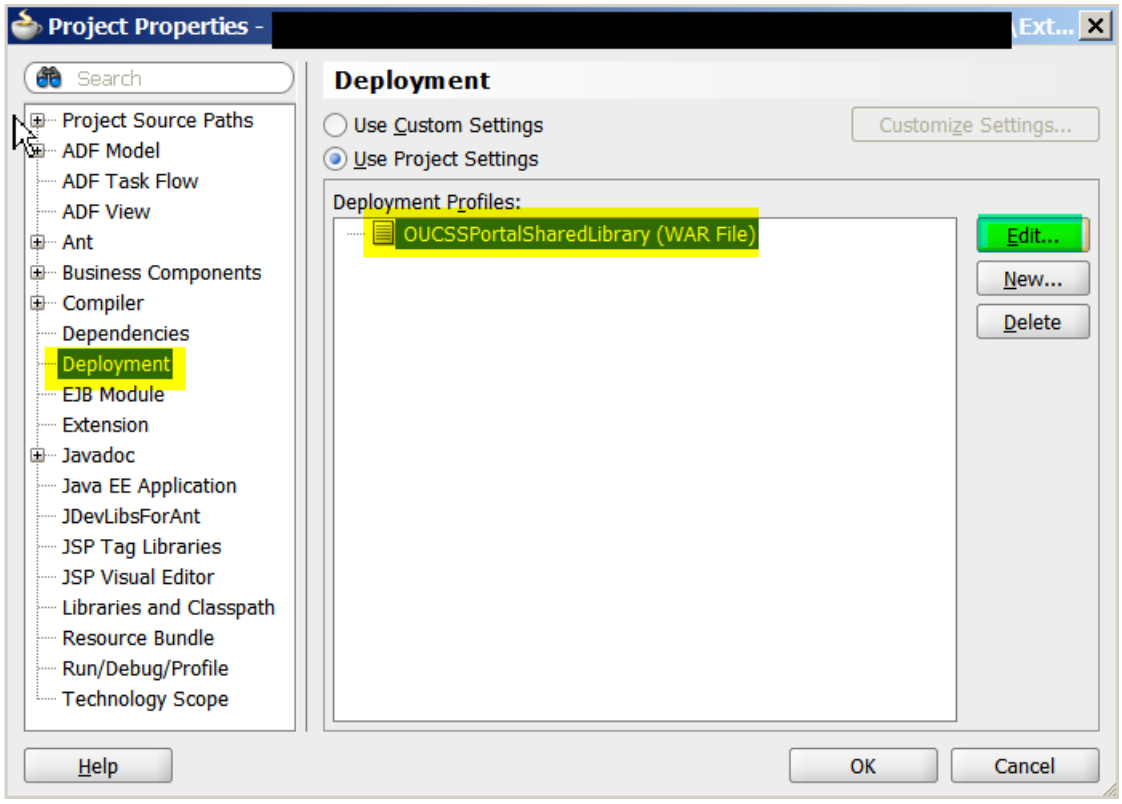
1 Modify the implementation version number in the `MANIFEST.MF` file in the **OUCSSPortalExtension** shared library (`<<Application_Home>>/src/META-INF/MANIFEST.MF`) before redeploying.

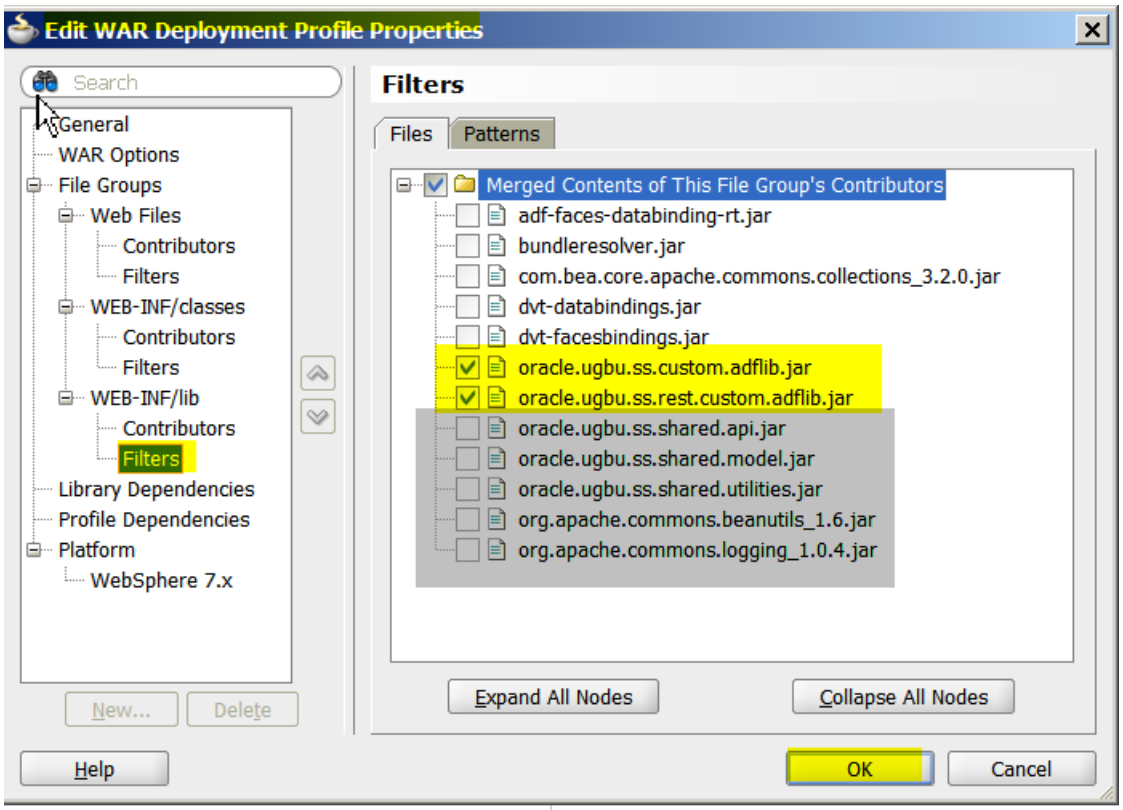
Note: The default implementation version number is 11.1.1.9.0. You can increment the major or minor version for every new revision deployed. For the purposes of this demo, the new implementation version number will be incremented to 11.1.1.9.1.



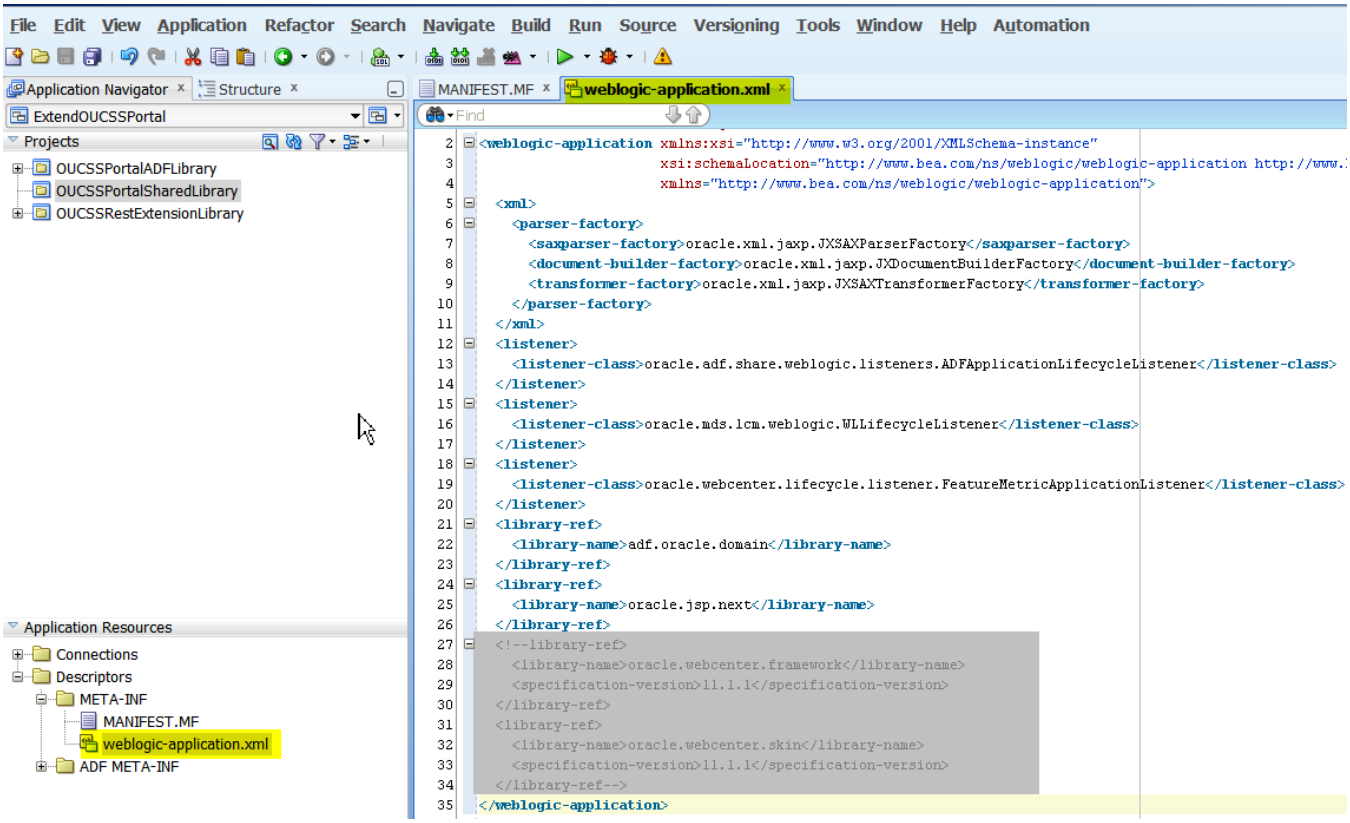
2 Edit the OUCSSPortalSharedLibrary project properties. Select **Deployments** and then edit the **OUCSSPortalSharedLibrary** deployment profile. Review the library references of the profile. Uncheck all jars references except **oracle.ugbu.ss.custom.adflib.jar** and **oracle.ugbu.ss.custom.rest.adflib.jar**.

Important: Remove any references to other jars from the extension library. If they are not removed, additional copies of these jars will be loaded at runtime and impact the application of patches and upgrades, and may cause other issues.

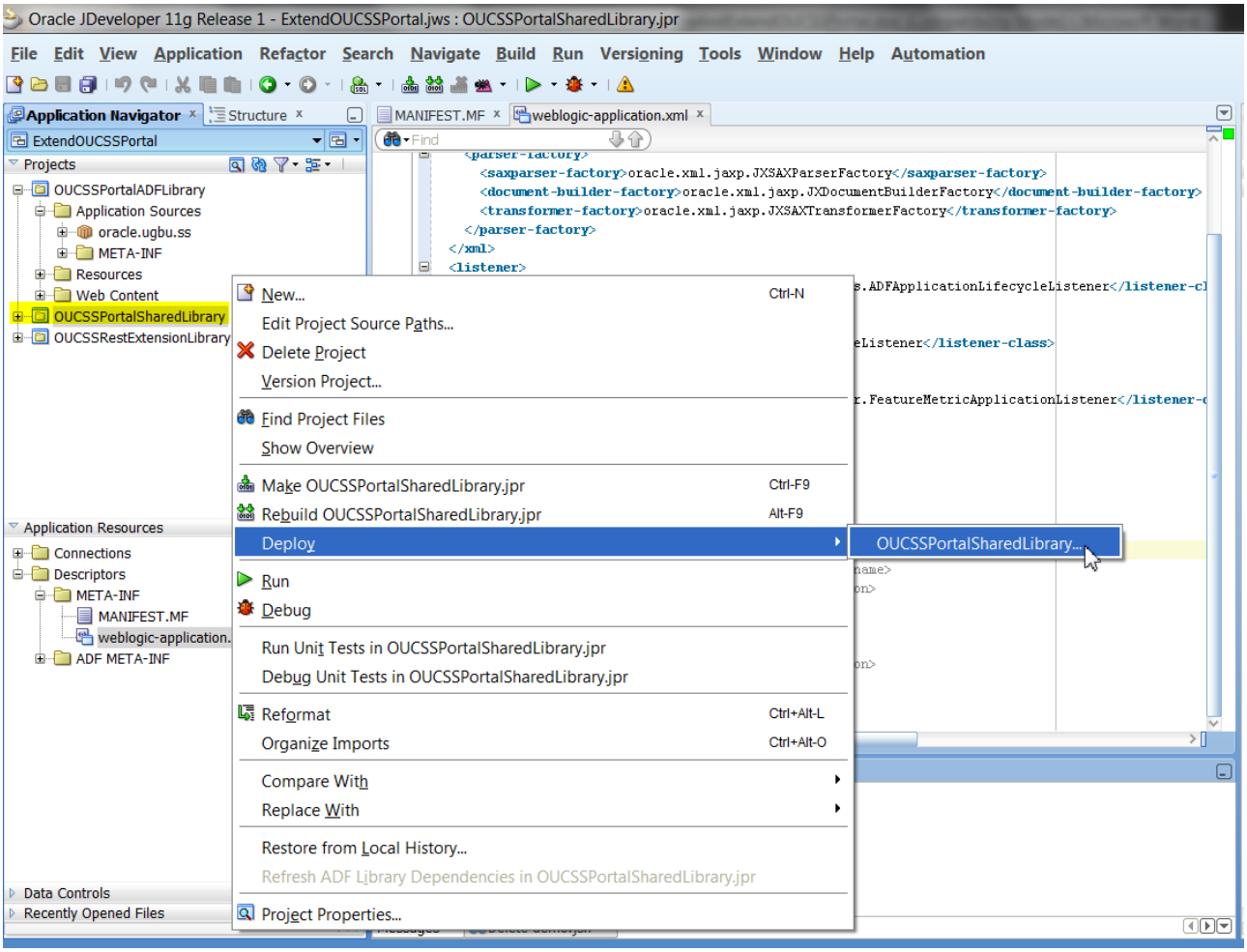




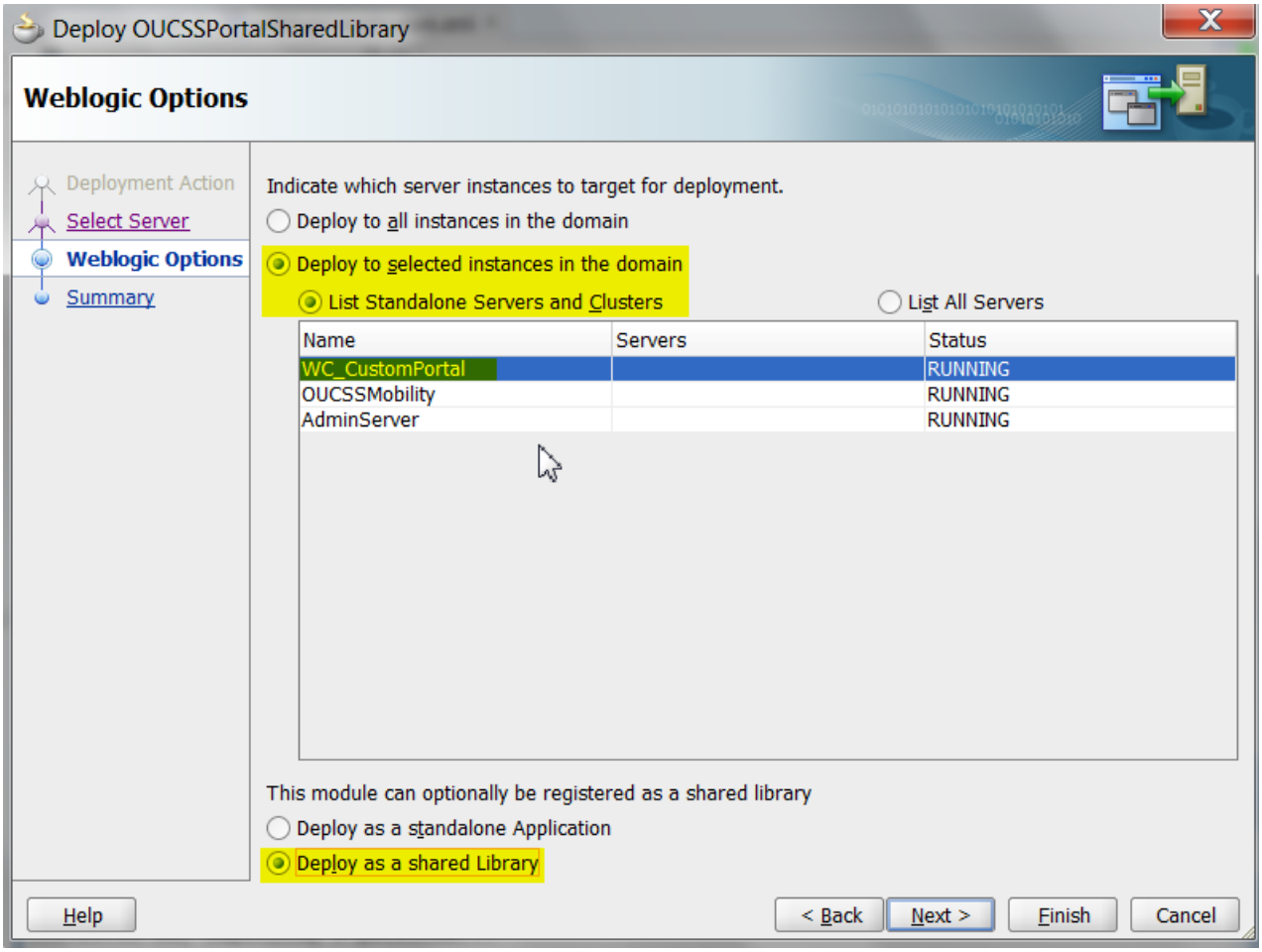
3 Make `weblogic-application.xml` and make sure that the library names `oracle.webcenter.framework` and `oracle.webcenter.skin` are commented and not active. Remove these entries if added by JDeveloper.



4 Deploy the OUCSS Portal Extension Shared Library to consume the taskflow in the portal. Right-click on the **OUCSSPortalSharedLibrary** project and select the **OUCSSPortalSharedLibrary** deployment profile to redeploy the extension to the managed server running the OUCSS Portal application.



5 Be sure to choose the option to deploy the war as a *shared library* and not as a standalone application.



6 Check **Deployments** in the WebLogic Console to verify that the deployment was successful and the library is **Active**.

7 The custom managed server must be restarted for the newly-created taskflows and pages to be consumed in the OUCSS Portal.

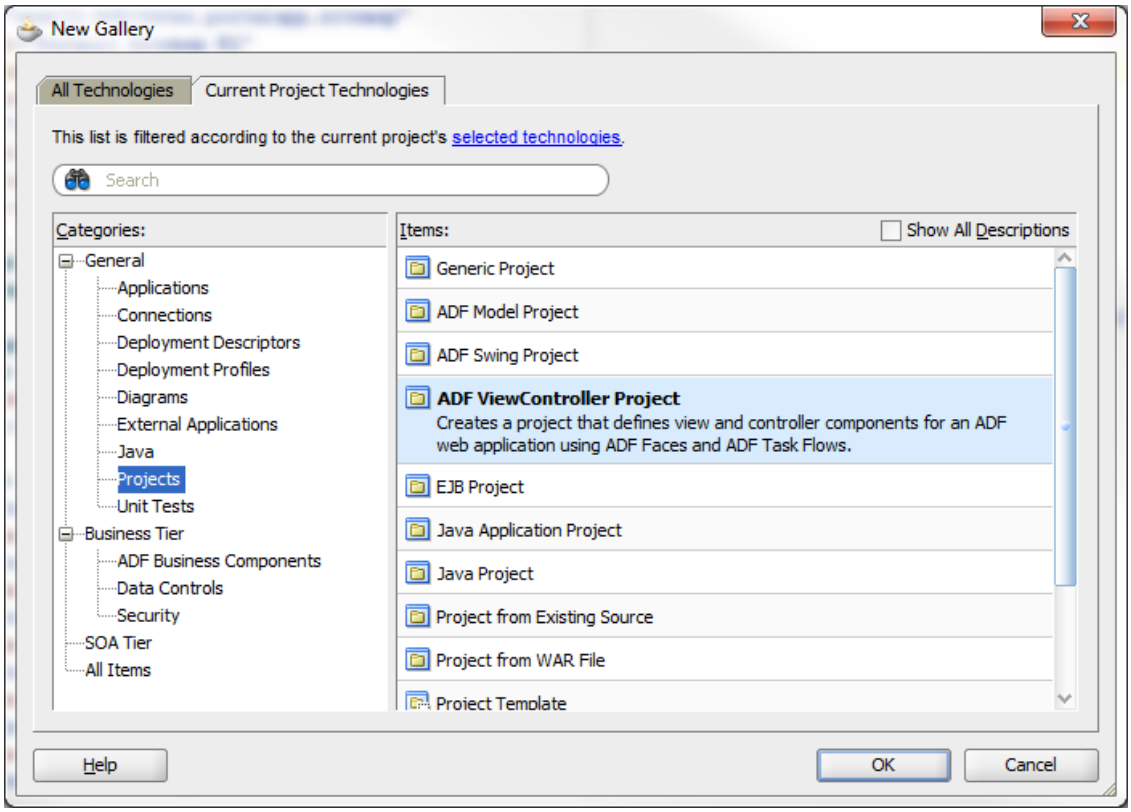
Parallel Development on the Portal Extension

The previous topic described how to use the portal extension project ExtendOUCSSPortal to create and deploy custom task flows and pages. This procedure describes how custom portal artifacts can be developed and maintained by multiple developers working in parallel, each creating individual projects inside the ExtendOUCSSPortal. The projects can then be bundled together to create an extend.oucss.portal application to be deployed onto the portal server.

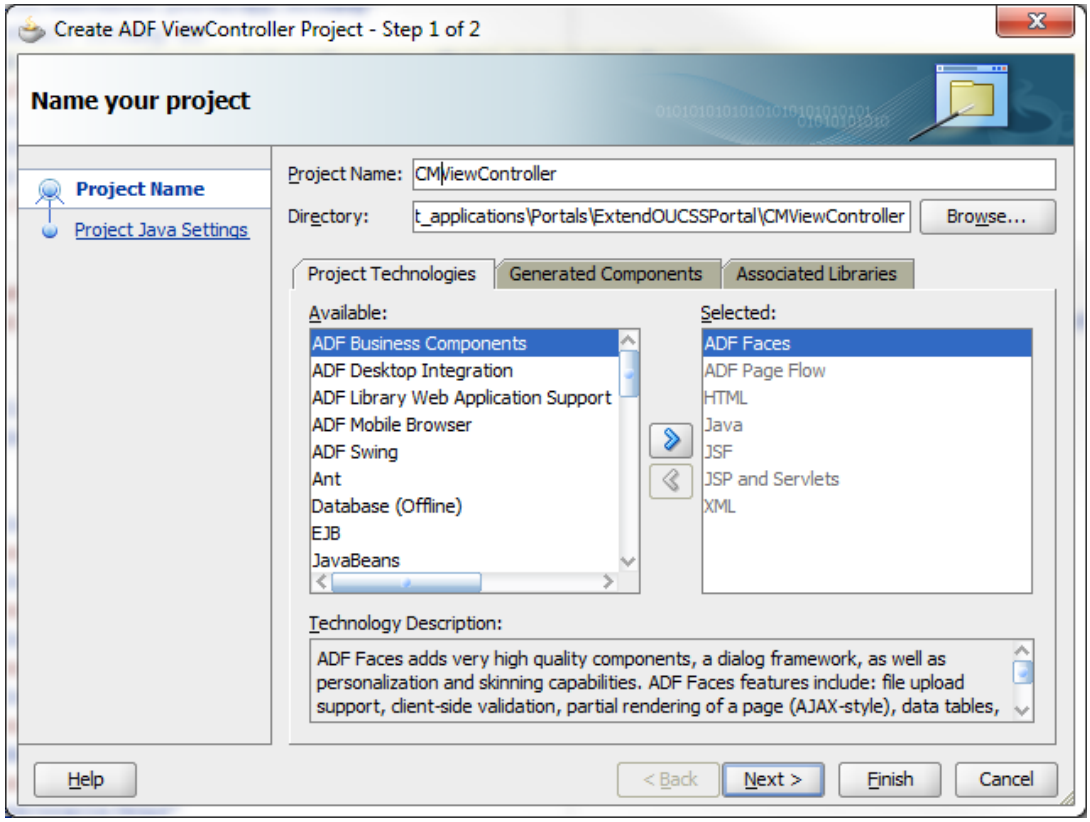
To enable parallel development on the portal extension:

1 Open the application **ExtendOUCSSPortal** in JDeveloper. Right-click on the application name under **Application Navigator**, then select **New**. From the **New Gallery** popup, navigate to the category **Projects** under **General** and select the project template for **ADF ViewController Project** or **ADF Model Project**.

To create new taskflows, pages, or Managed Beans, create an ADF ViewController Project.

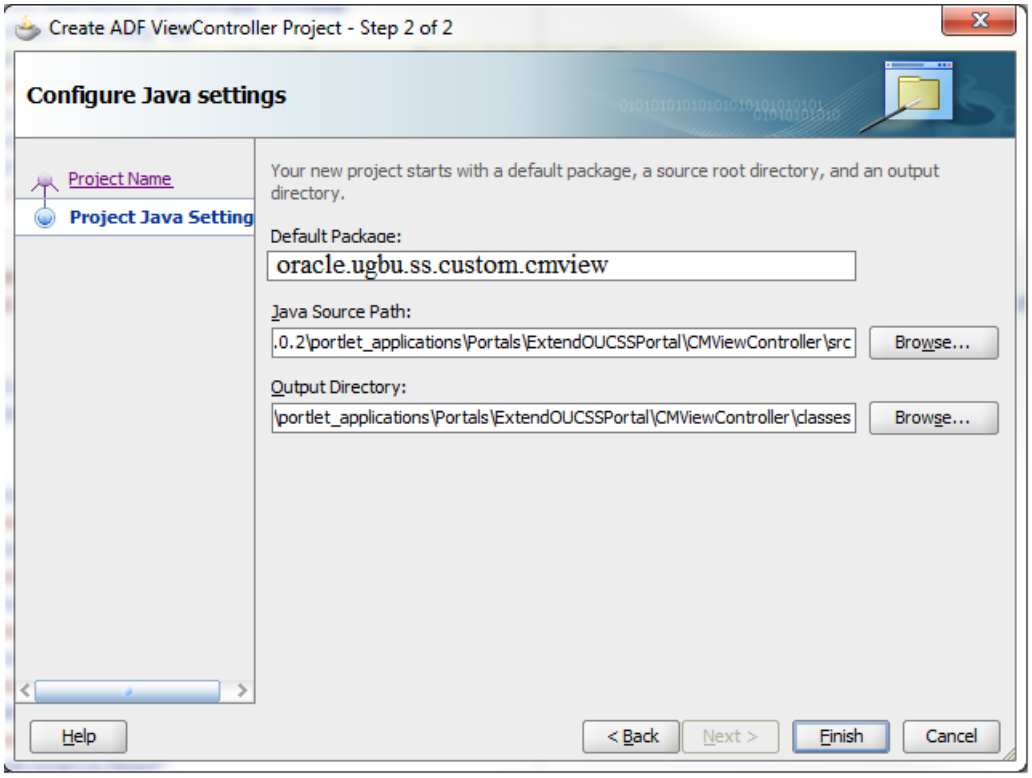


2 Since the project **Name** must be unique, choose a Name in coordination with other team members. For illustration purposes, the project name used in this procedure is **CMViewController**.

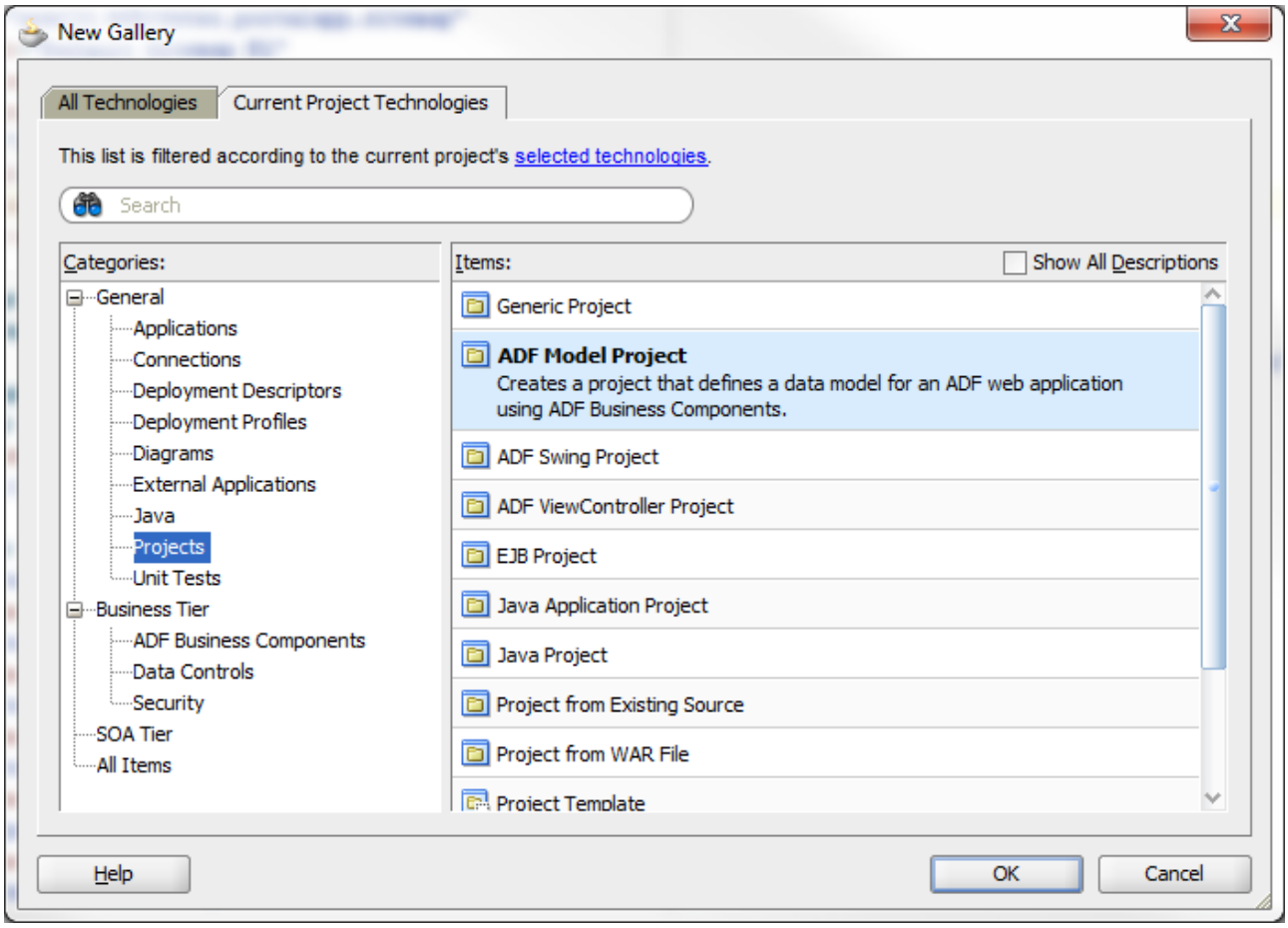


3 Select the default package, java source path, and output directory in the next screen to create the project.

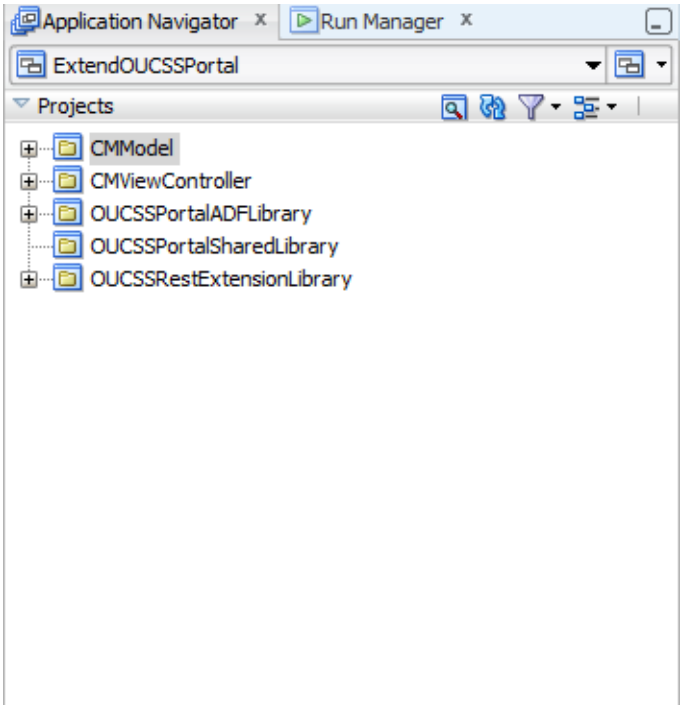
Important: To avoid ADF metadata conflicts (e.g., DataBindings.cpx, etc), ensure that the default package is unique among all custom projects. Choose the default package as `oracle.ugbu.ss.custom.<<Unique_Modulename>>`. For illustrative purposes, the default package used in this example is `oracle.ugbu.ss.custom.cmview`.



4 If required, create another project of category **ADF Model Project** following the same steps as described earlier for the ADF ViewController Project. Model projects are required when there is a need to create data controls, proxy classes, or another business model/ADFBC.

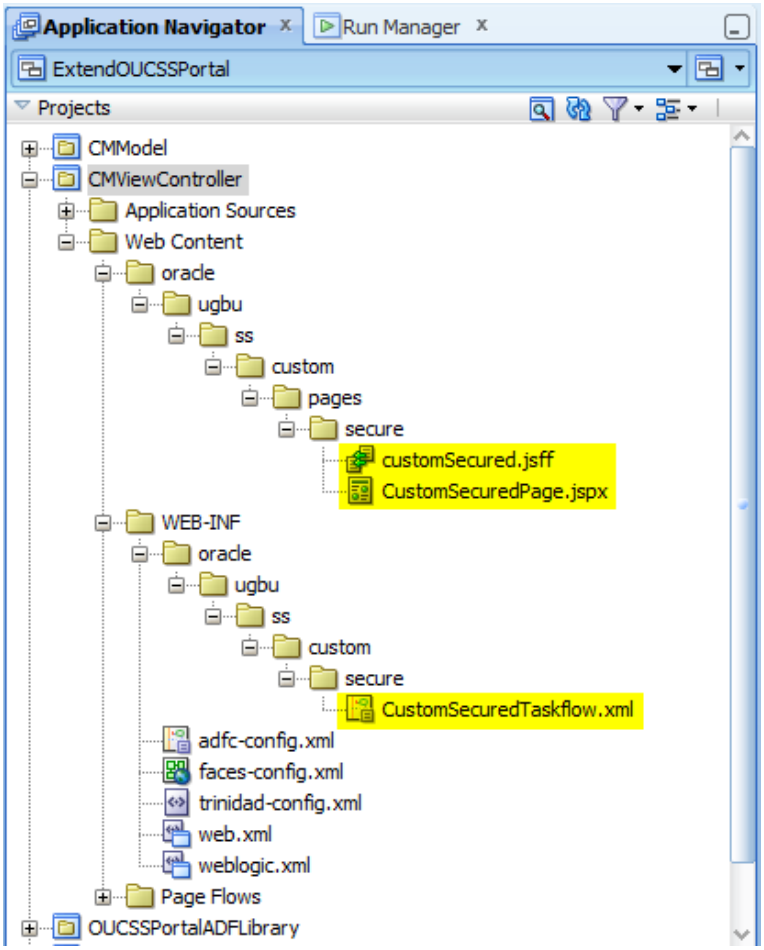


5 When completed, the ExtendOUCSSPortal should contain the new projects along with the out-of -the-box projects, as shown in the following image:

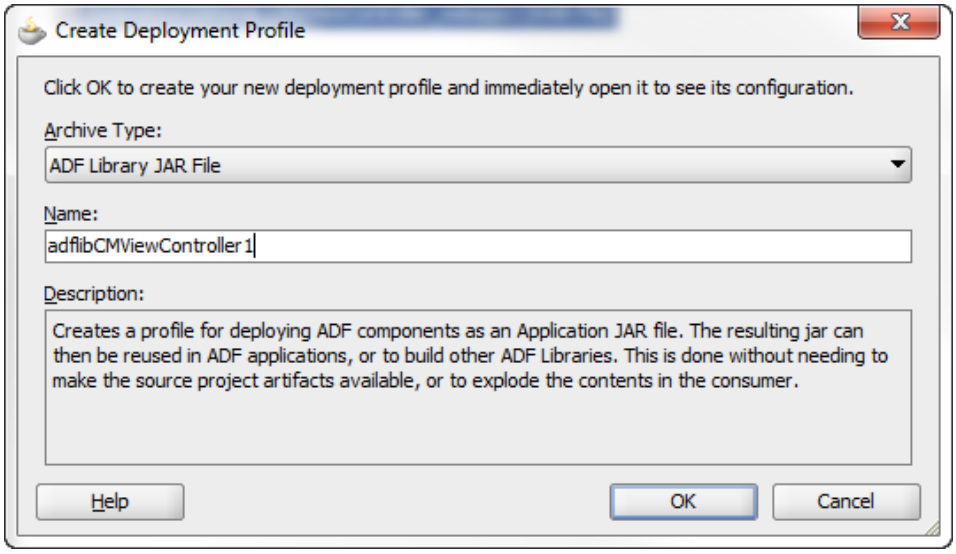


6 Create custom task flows and pages inside the new projects following the same steps as described in the Adding New Taskflows and Pages section of this document. Also review the About Security section to understand the folder structure that should be maintained while creating portal resources.

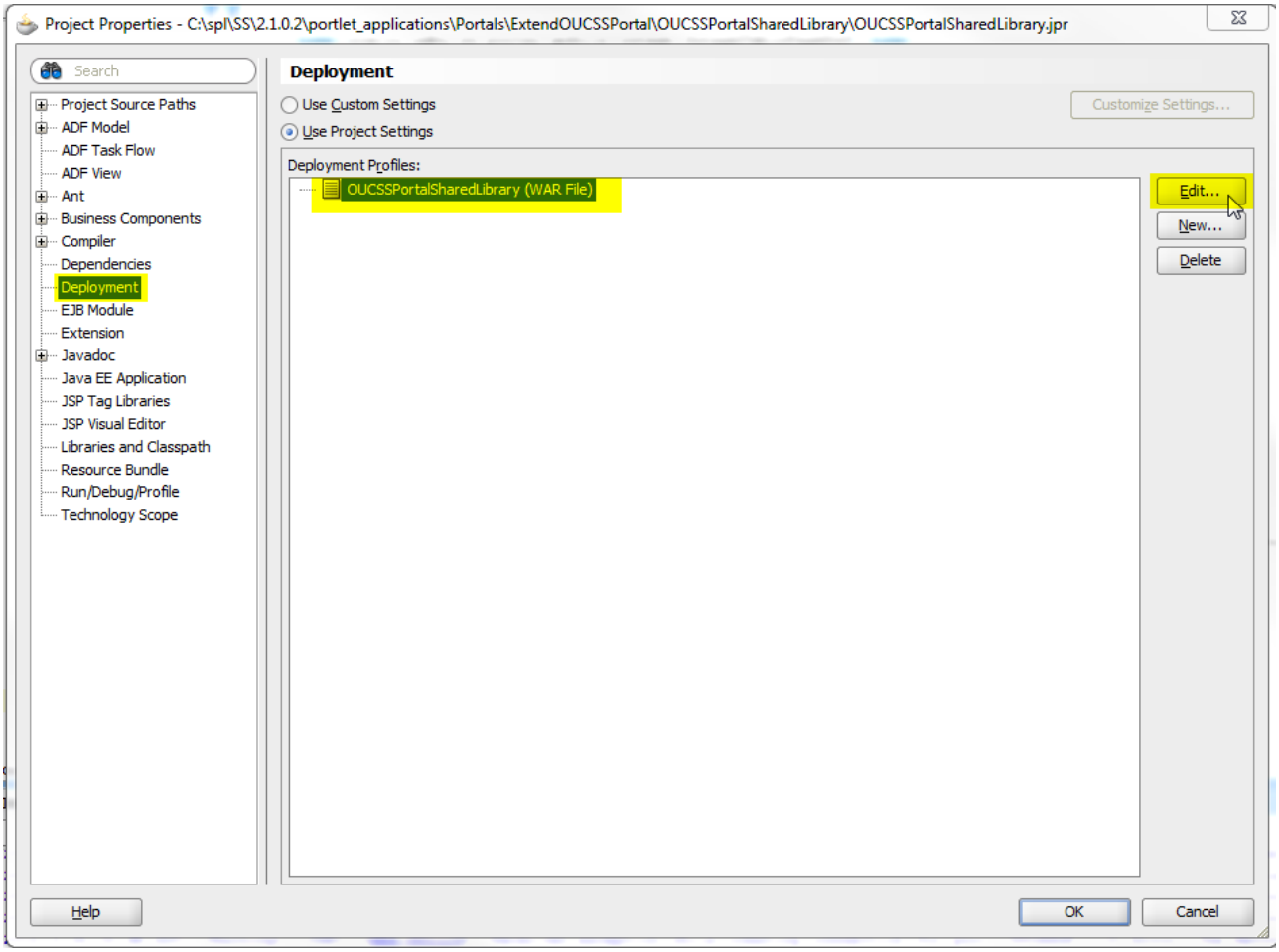
7 For the purposes of this example, a secured taskflow and page has been created inside the CMViewController project.



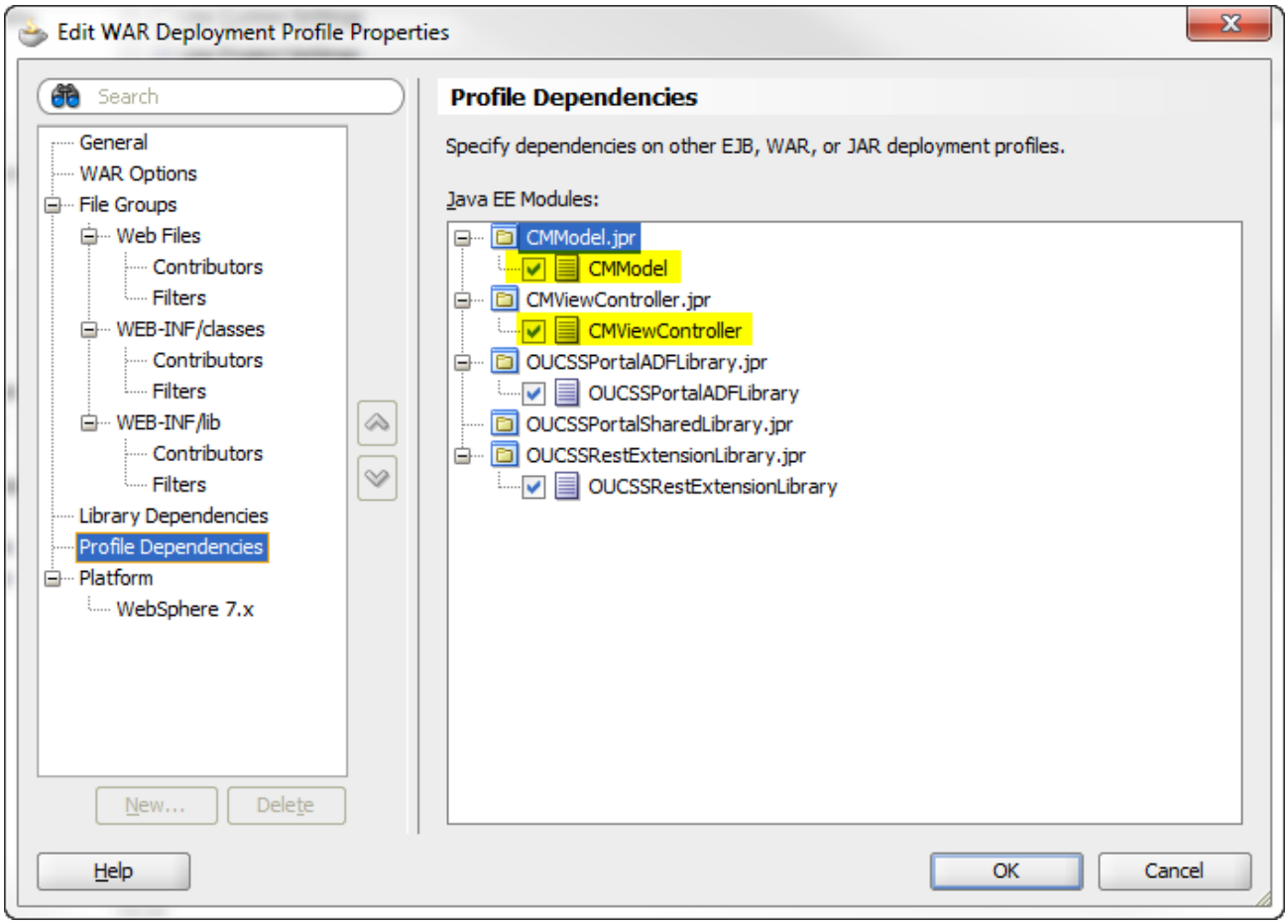
8 Create an ADF Library JAR deployment profile for the new projects.



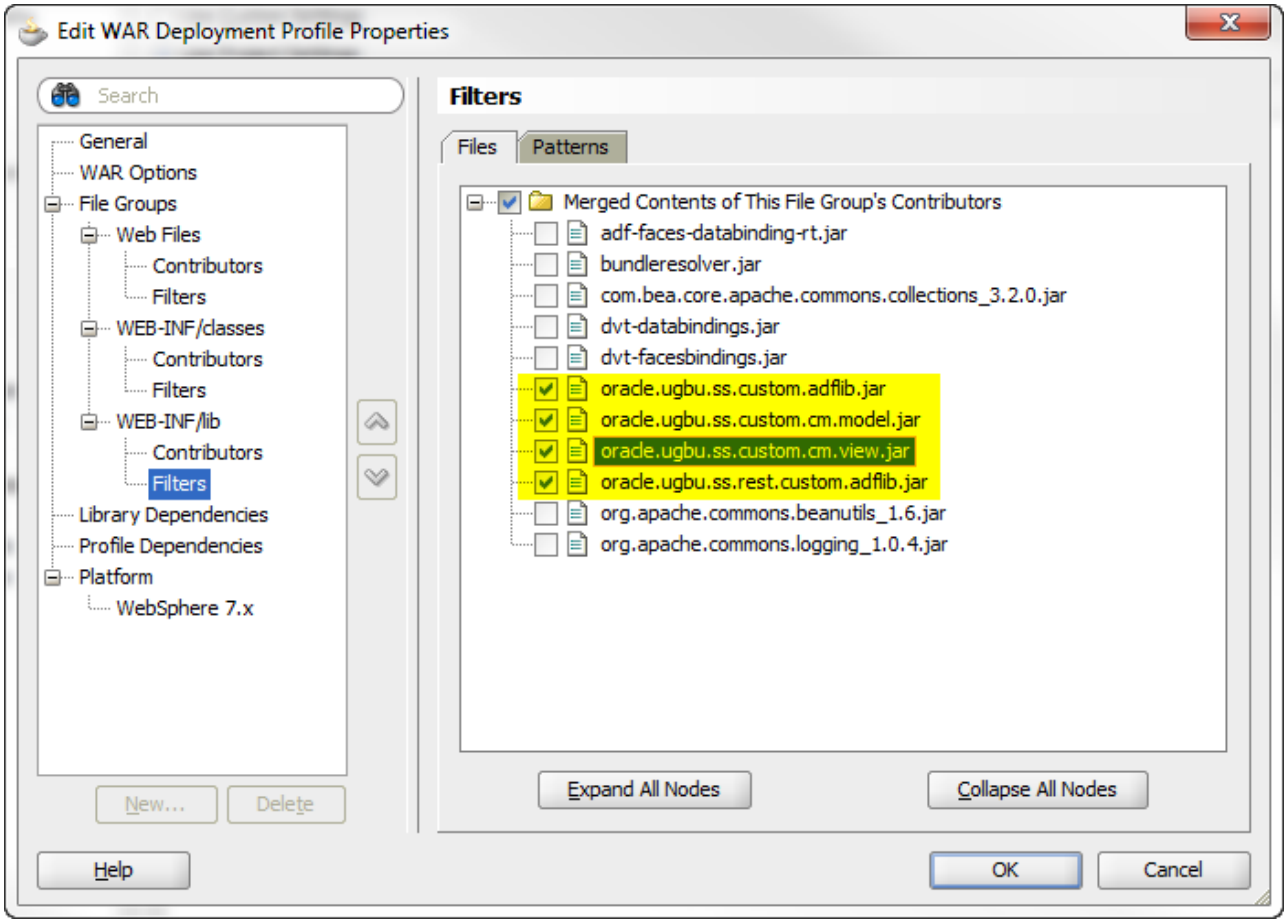
9 Edit **Project Properties** for the OUCSSPortalSharedLibrary, then navigate to **Deployment** and select and edit the deployment profile.



10 Review **Profile Dependencies** in the deployment profile to make sure all relevant projects inside ExtendOUCSSPortal are checked in.



11 Verify the library references for this profile and make sure the relevant libraries for projects inside ExtendOUCSSPortal are checked. The other libraries should not be checked, else those can have multiple copies during runtime creating issues.



12 Review the Deploying the extend.oucss.portal as a Shared Library section in this document for information on deploying ExtendOUCSSPortal as a shared library.

13 Review the Configure the Resource Catalog section of this document for information on consuming the taskflow in OUCSS Portal.

14 Review the Access Pages from Extend OUCSS Portal section of this document for information on accessing pages in the OUCSS Portal.

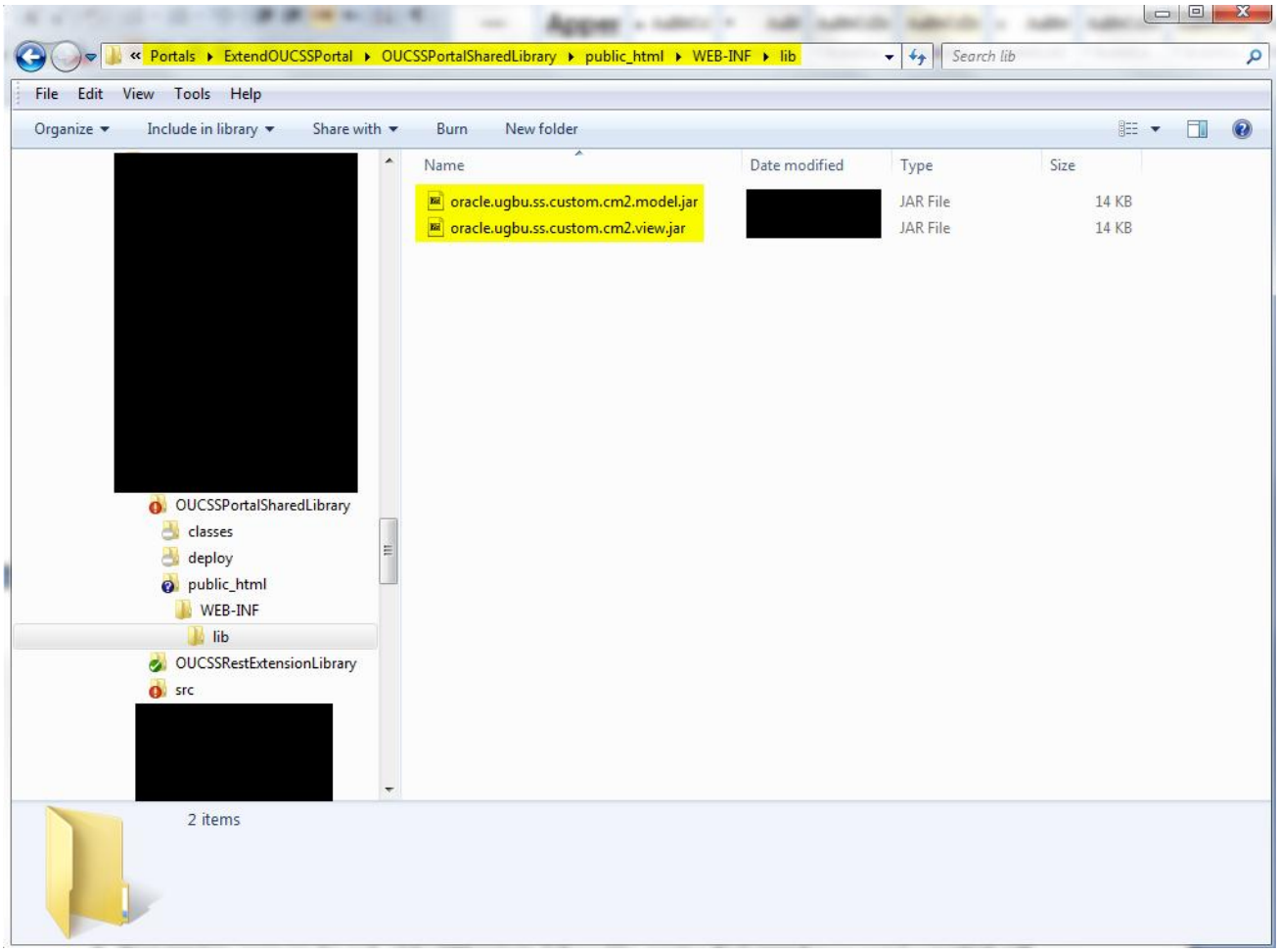
Deploying CM ExtendOUCSSPortal Projects

The previous topic described how individual developers can create and develop custom portal resources using CM projects. As multiple CM projects are created by developers, they need to be deployed together onto the portal server.

This procedure describes how to deploy multiple CM projects.

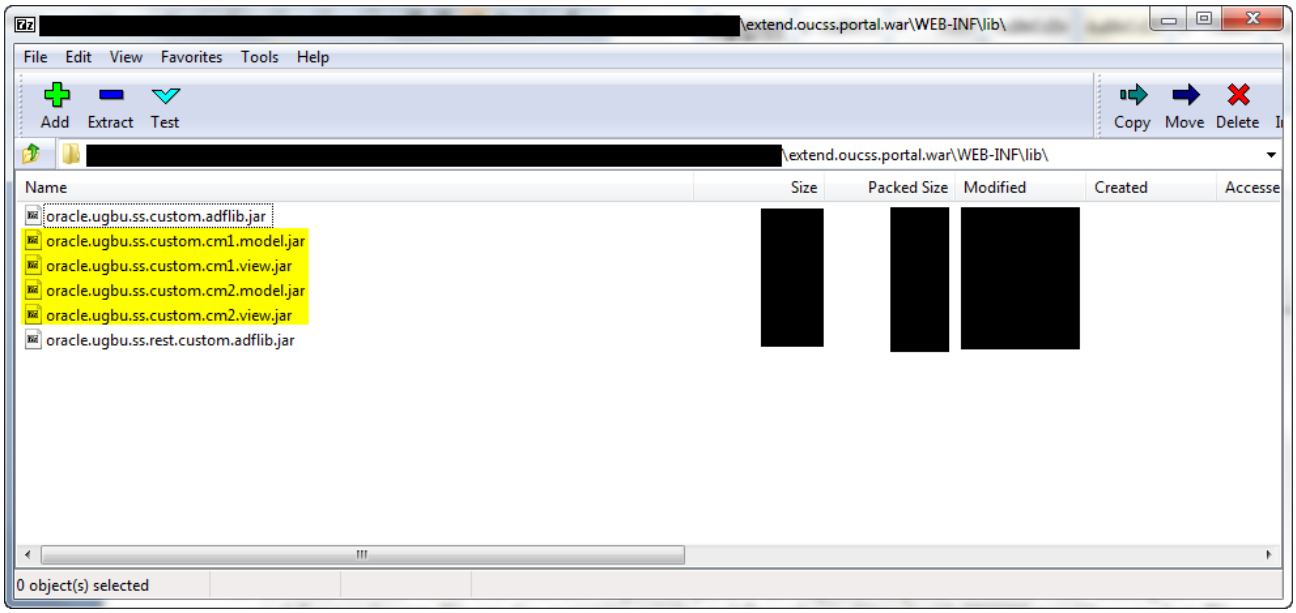
1 Create the folder structure `public_html\WEB-INF\lib` in `<ApplicationFolder>\OUCSSPortalSharedLibrary` and put the ADF library jars for all CM projects that are *not* part of this application into this folder. There is no need to copy the ADF library jar of the CM projects that are part of the same ExtendOUCSSPortal application, since they will already be linked with the OUCSSPortalSharedLibrary as described in the previous topic.

2 For illustration purposes, assume there are two CM projects, *CM1* and *CM2*, which need to be deployed to the portal server. Let's also assume that the ExtendOUCSSPortal application used to create final deployable `extend.oucss.portal.war` contains CM1 projects, and CM2 projects were created outside this application. In this case, the `public_html\WEB-INF\lib` folder created in the OUCSSPortalSharedLibrary project will contain only the ADF library jar of CM2 projects. You may use a shared folder or source control tool to store all such ADF library jars and include them into a lib folder while creating the final deployable war file.



3 Refer to the Deploying the extend.oucss.portal as a Shared Library topic earlier in this document for instructions on deploying ExtendOUCSSPortal.

4 To verify deployment of all CM projects (both internal and external to the ExtendOUCSSPortal application used for .war creation), you can expand the `extend.oucss.portal.war` and navigate to `WEB-INF\lib`.



Parallel Customization of Existing Taskflows

Previous topics described how to perform parallel development on a portal extension while creating new taskflows and pages. You can also customize existing taskflows that are provided with the OUCSS portal base product. This procedure describes how to perform such customization in parallel.

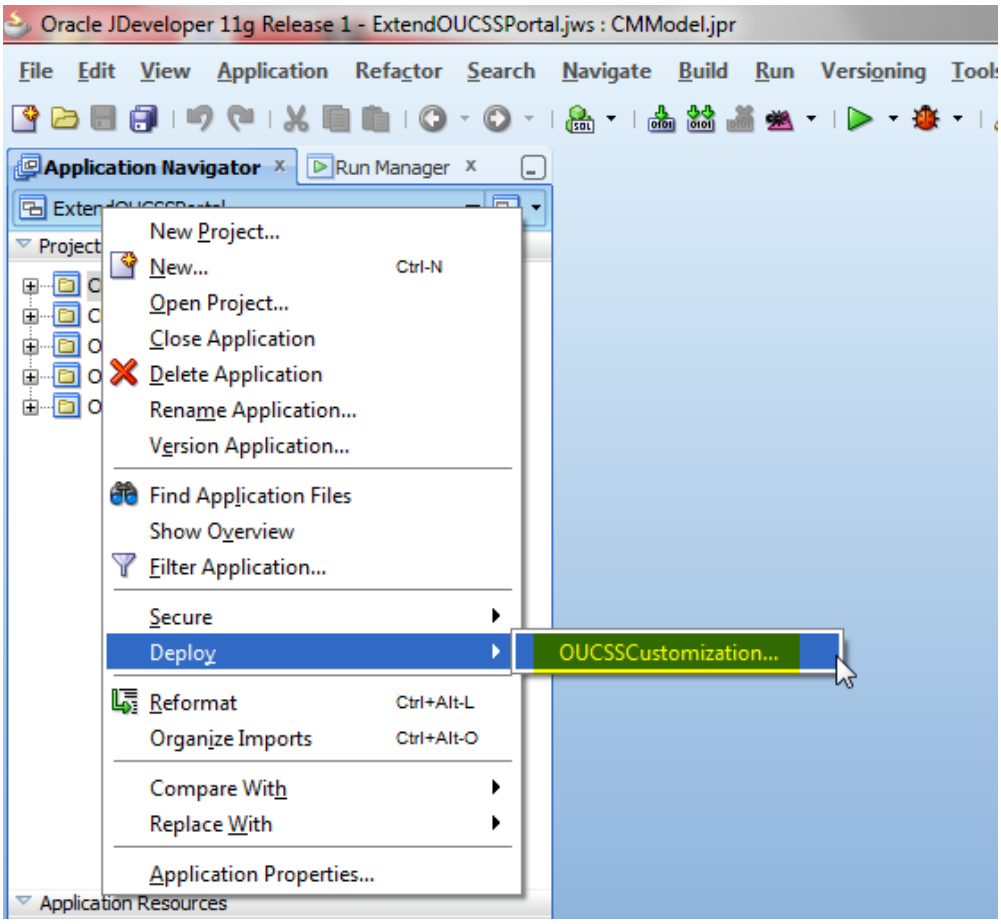
1 Review **Chapter 9 – Customization and Extension** in the *OUCSS Implementation Guide* for the steps to follow for customizing existing taskflows.

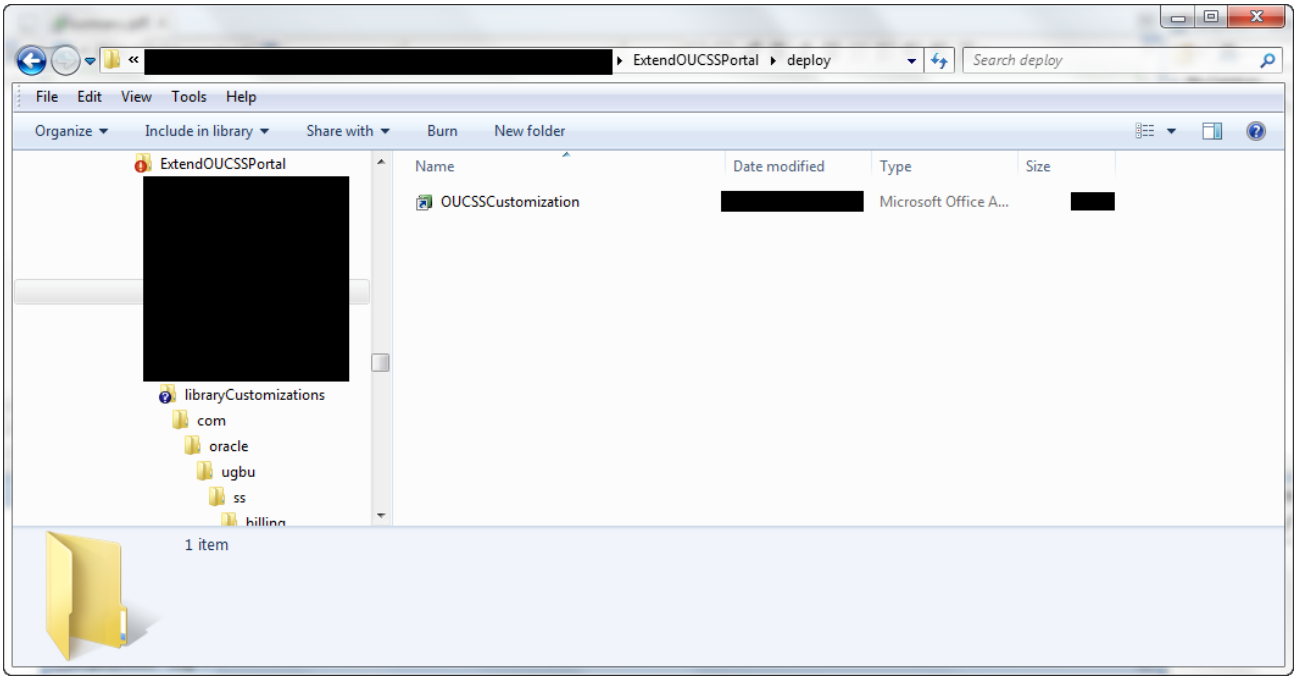
2 Once a customization is performed on a taskflow, JDeveloper will create a customization file containing only the differences between the original and the customized portal artifact. These files need to be uploaded to MDS and can be found in the `<ApplicationFolder>\OUCSSPortalADFLibrary\libraryCustomizations` folder.

3 The customization files generated by JDeveloper will be XML files, typically named `*.jsff.xml` or `*pageDef.xml.xml`. These files are created in a folder structure within the `<<ApplicationFolder>>\OUCSSPortalADFLibrary\libraryCustomizations` folder. It is important to keep this folder structure unaltered.

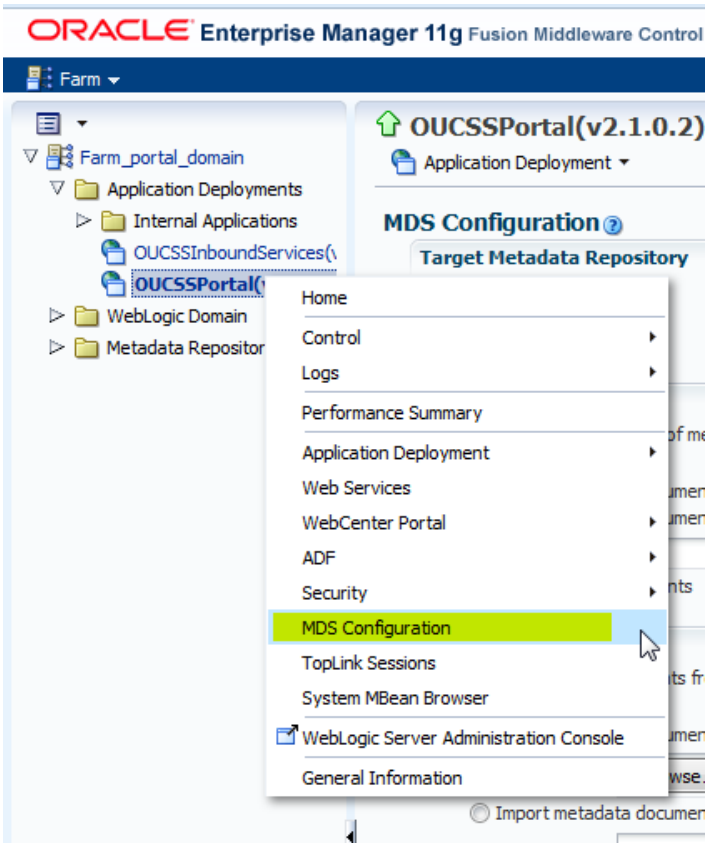
4 Combine all the customization artifacts created by multiple developers into the `libraryCustomizations` folder of the ExtendOUCSSPortal application which will be used to create the final deployable `.MAR` file. In case of any conflict, merge the customization artifacts to make sure one doesn't override another customization.

5 Right click on **ExtendOUCSSPortal** under **Application Navigator** and select **Deploy**, then **OUCSSCustomization**. ExtendOUCSSPortal already contains the deployment profile to create the `.MAR` file from its customization folder.

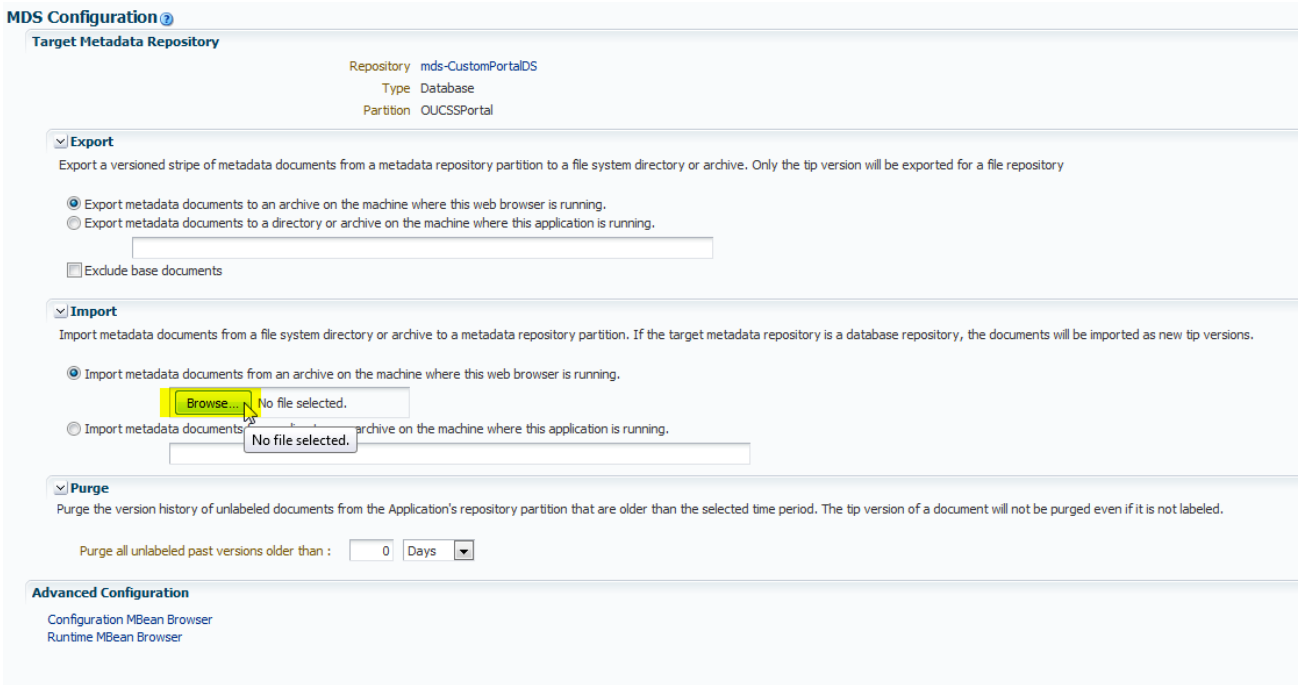




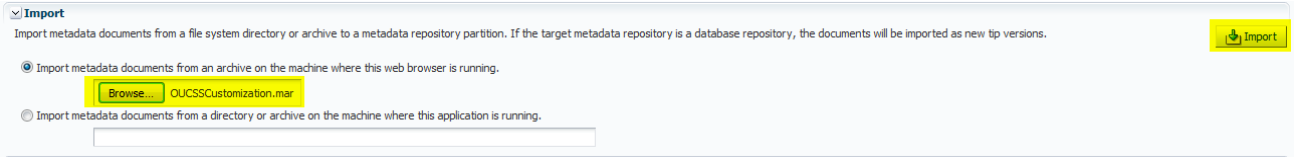
6 To deploy the `.MAR` file to the portal server, log in to the Enterprise Manager (EM) console of the server running the OUCSS portal, then navigate to the portal application under **Application Deployment**. Right-click and select **MDS Configuration** from the pop-up menu.



7 Browse to and select the `.MAR` file to be uploaded to MDS.



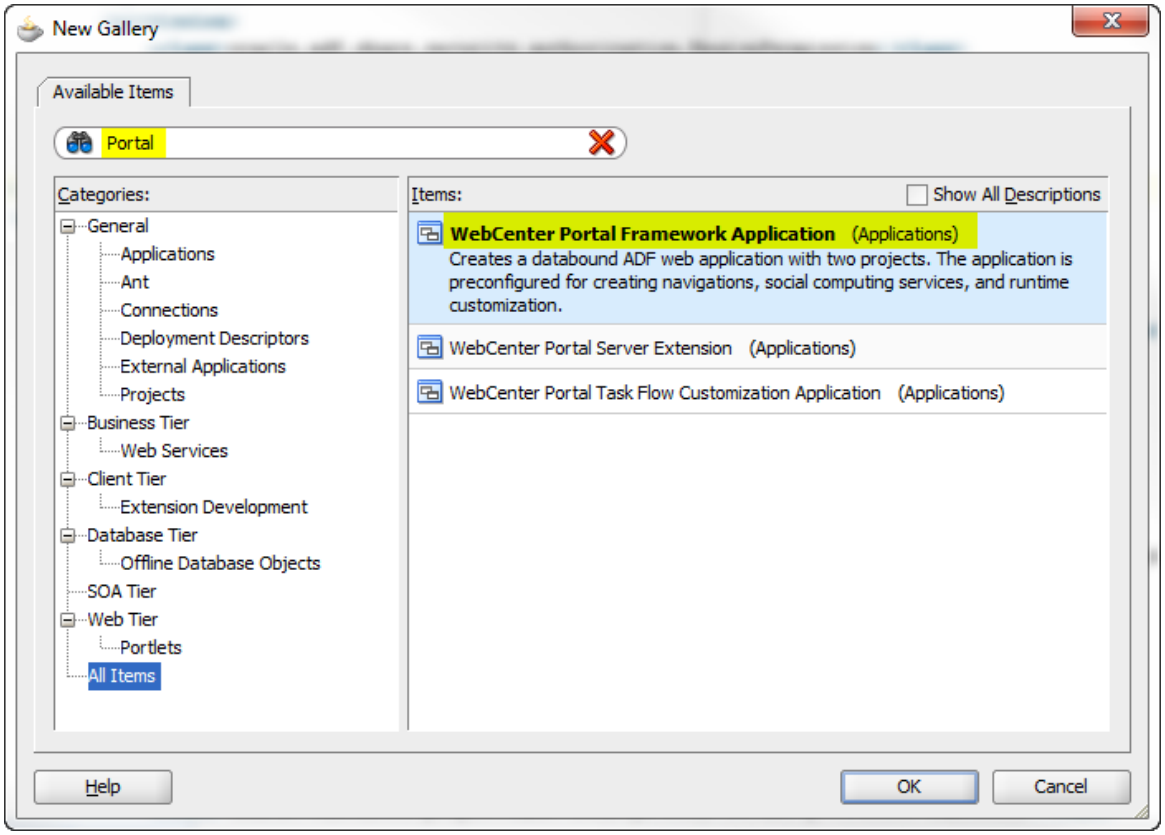
8 Click on **Import** to import the `.MAR`.



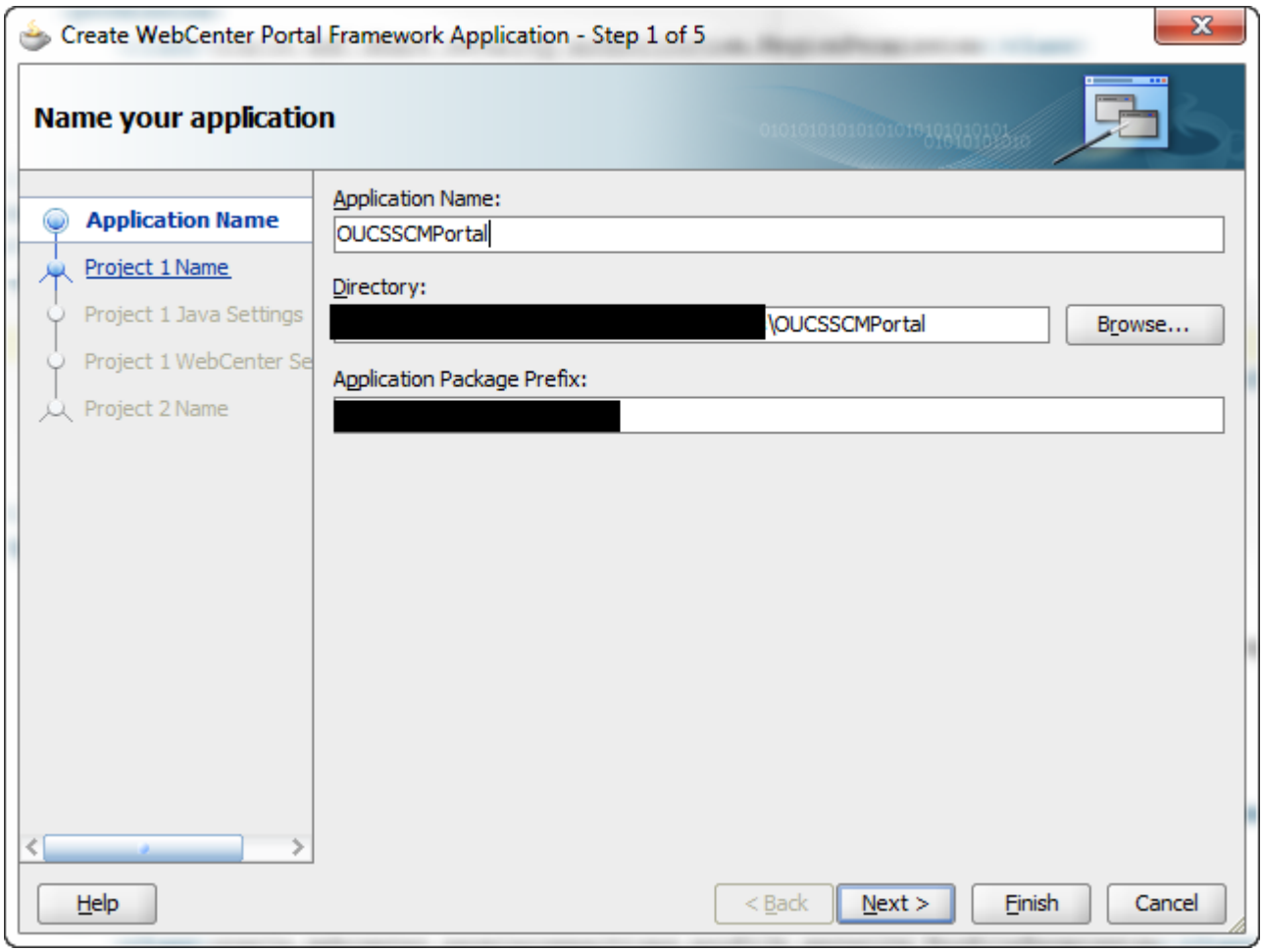
WebCenter Portal Application Local Instance

Custom taskflows and pages created using the procedures described in earlier topics can be tested by deploying the portal extension project to the portal server and creating portal resources to consume and display the artifacts. If multiple developers are working on creating and testing the portal extension on the same portal server, it could be tedious to coordinate the parallel deployments on the same server or will require each developer to host an instance of the OUCSS Portal. The steps below describe how the portal artifacts from an extension project can be tested locally in JDeveloper by creating local instance of the WebCenter Portal Framework Application.

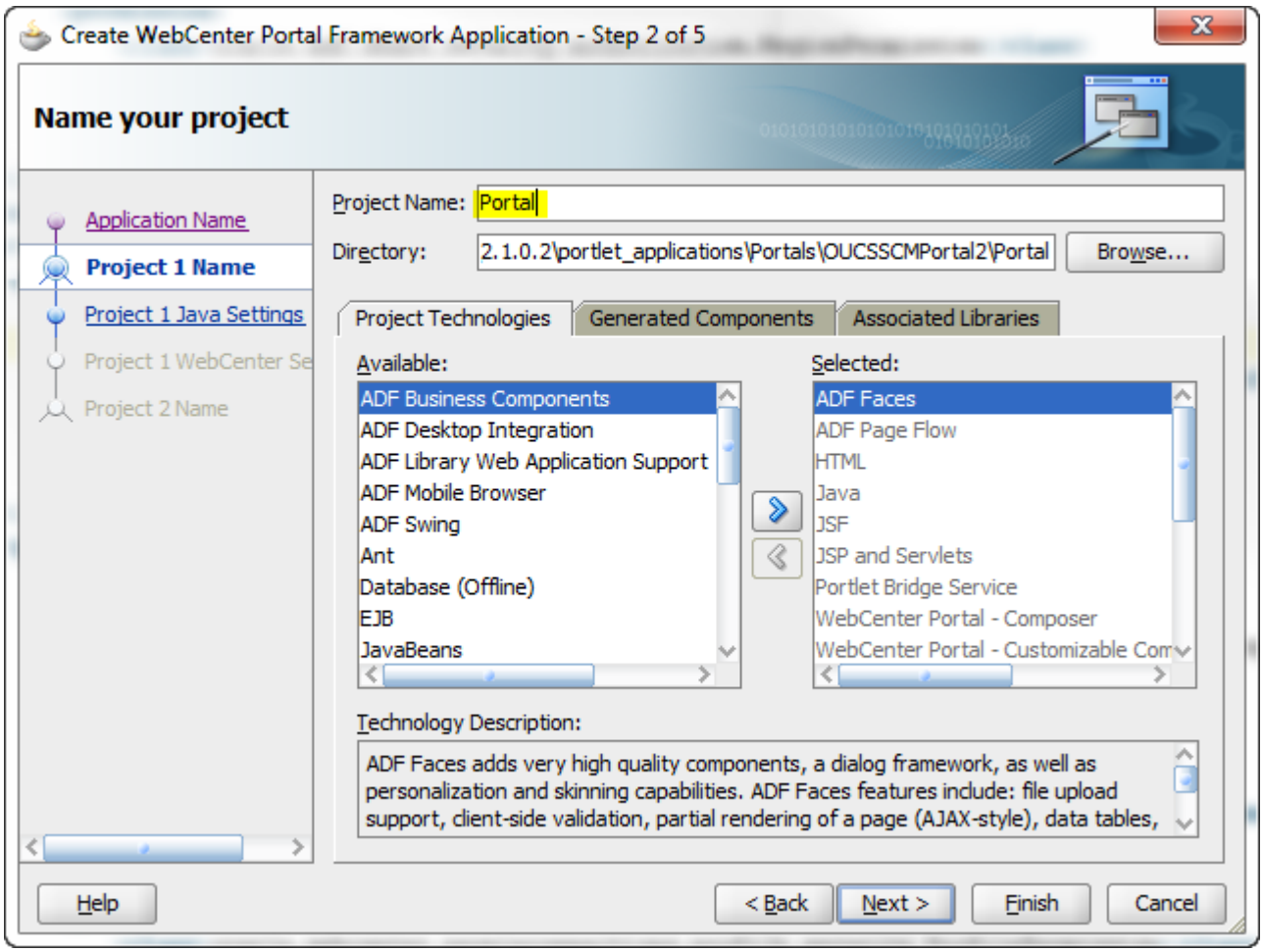
1 Create a new portal application by selecting **File** followed by **New** from the JDeveloper menu bar. Search for **Portal** in the **New Gallery** and select the application template for **WebCenter Portal Framework Application**.



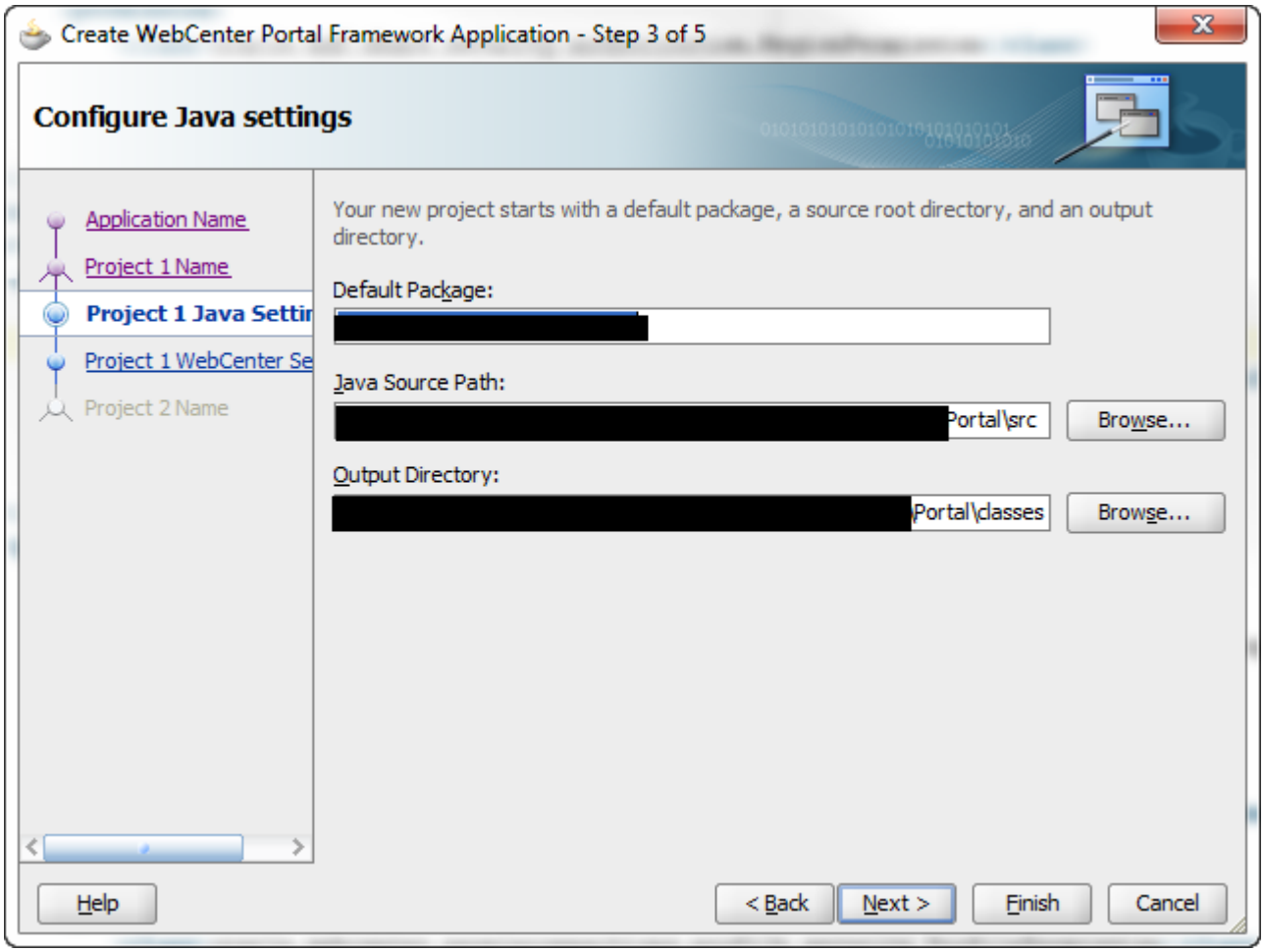
2 Enter a name for this portal application and specify the source directory and package prefix.



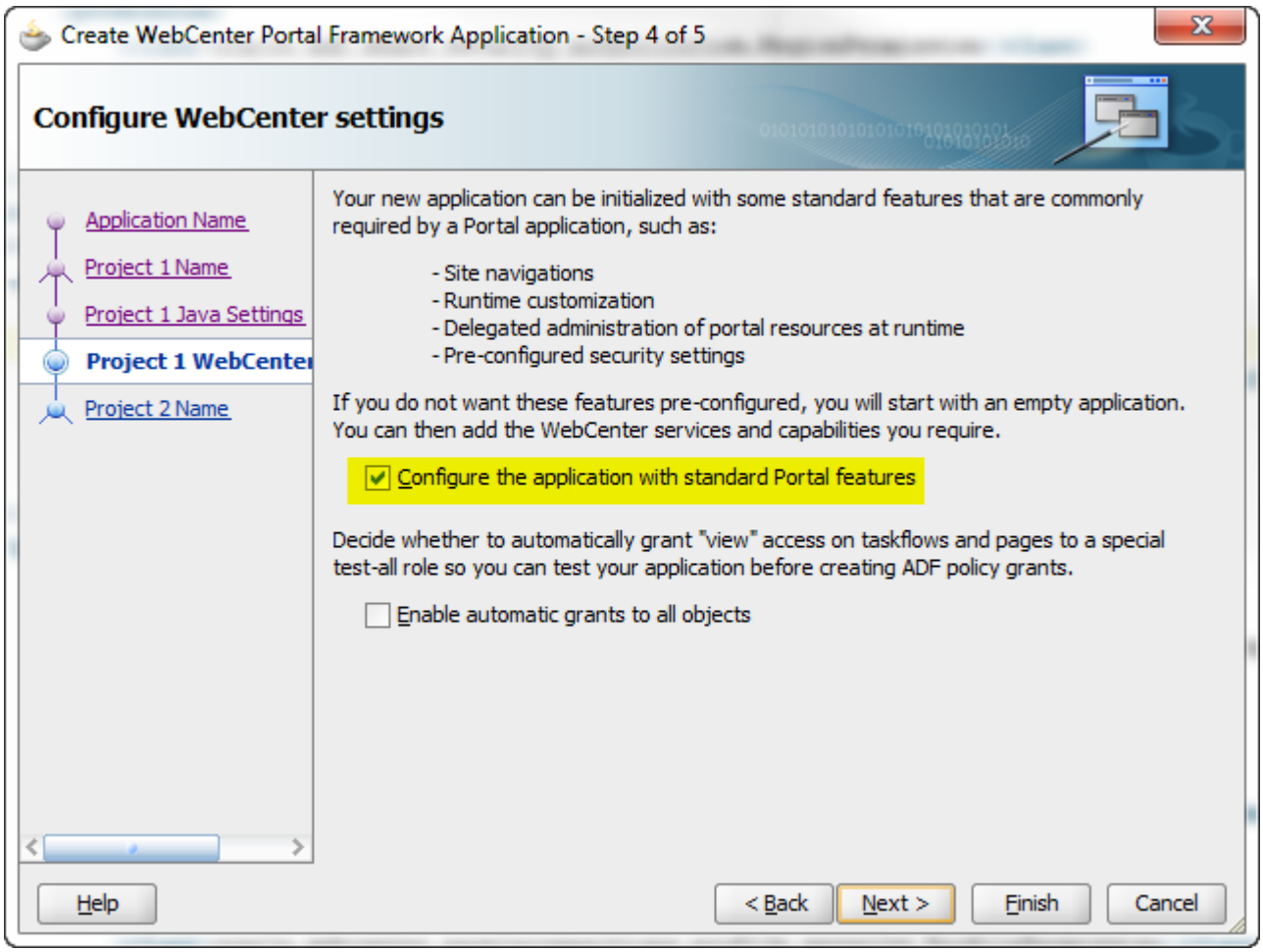
3 Select a name for the project. For this example, the name is set to **Portal**.



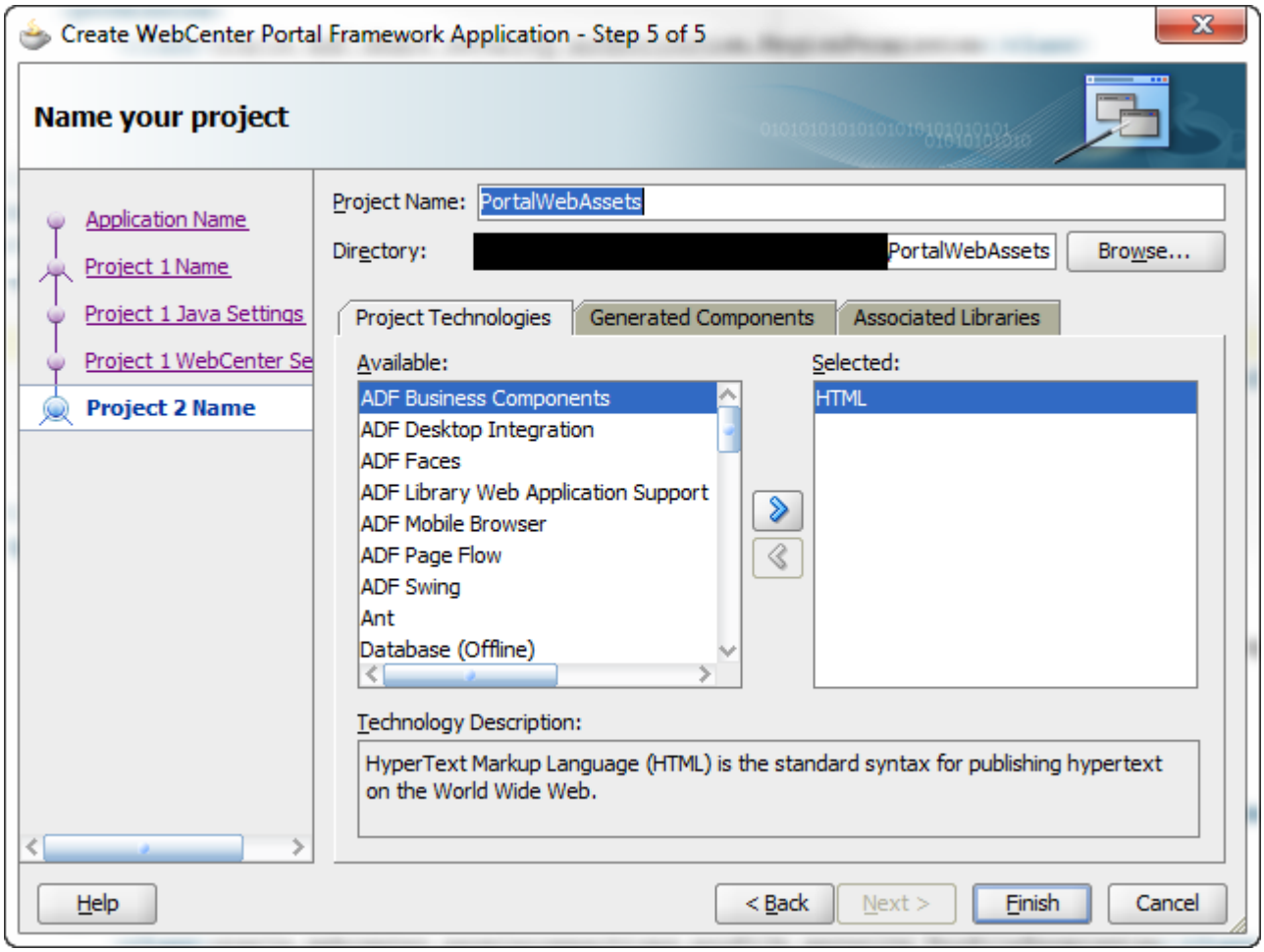
4 Enter the default package, Java source path, and output directory. Since the Portal is tested locally, the default package name can be anything.



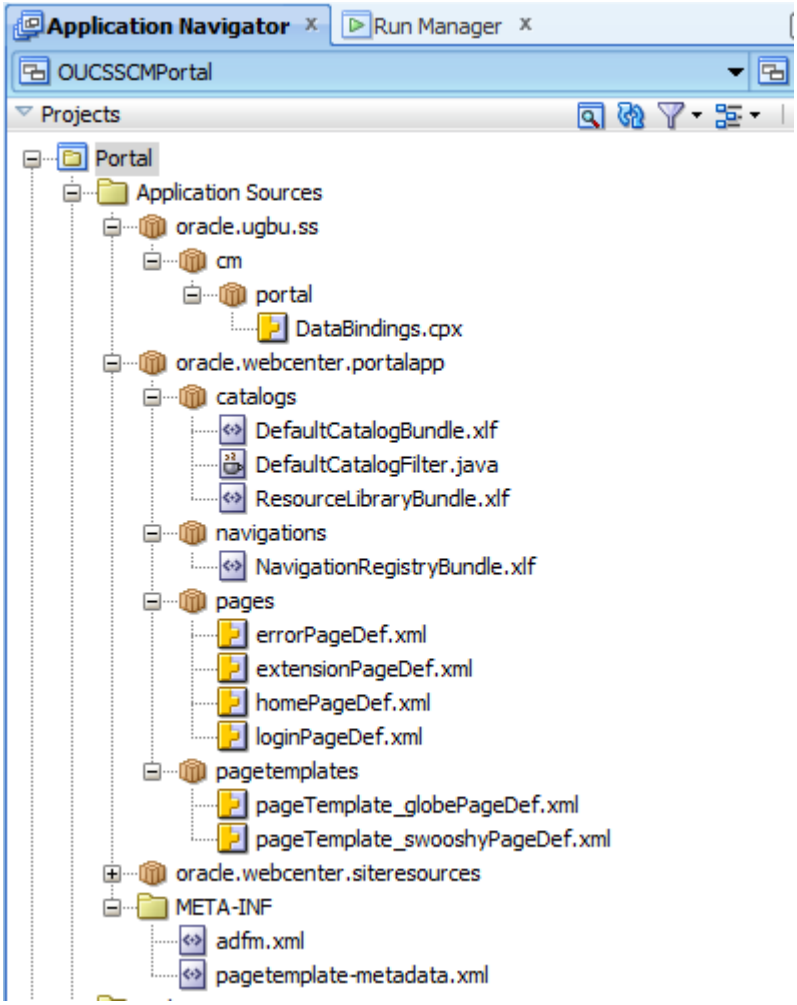
5 Verify that the option for **Configure the application with standard Portal features** is checked, then press **Next**.



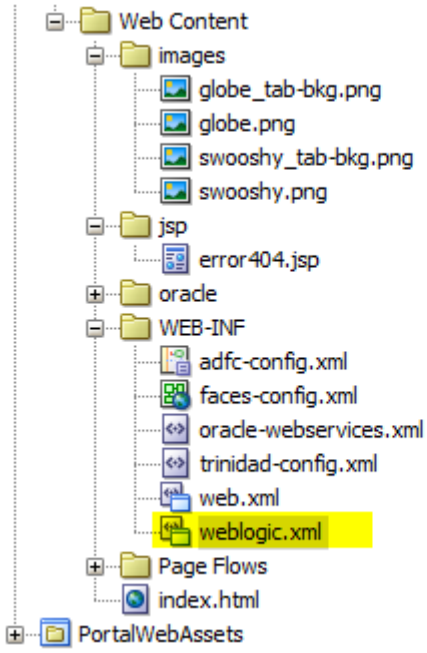
6 Select the appropriate project name for a project containing static portal assets (you may keep the default name **PortalWebAssets** as is), then click **Finish**.



7 The portal application may be viewed in the **Application Navigator**.



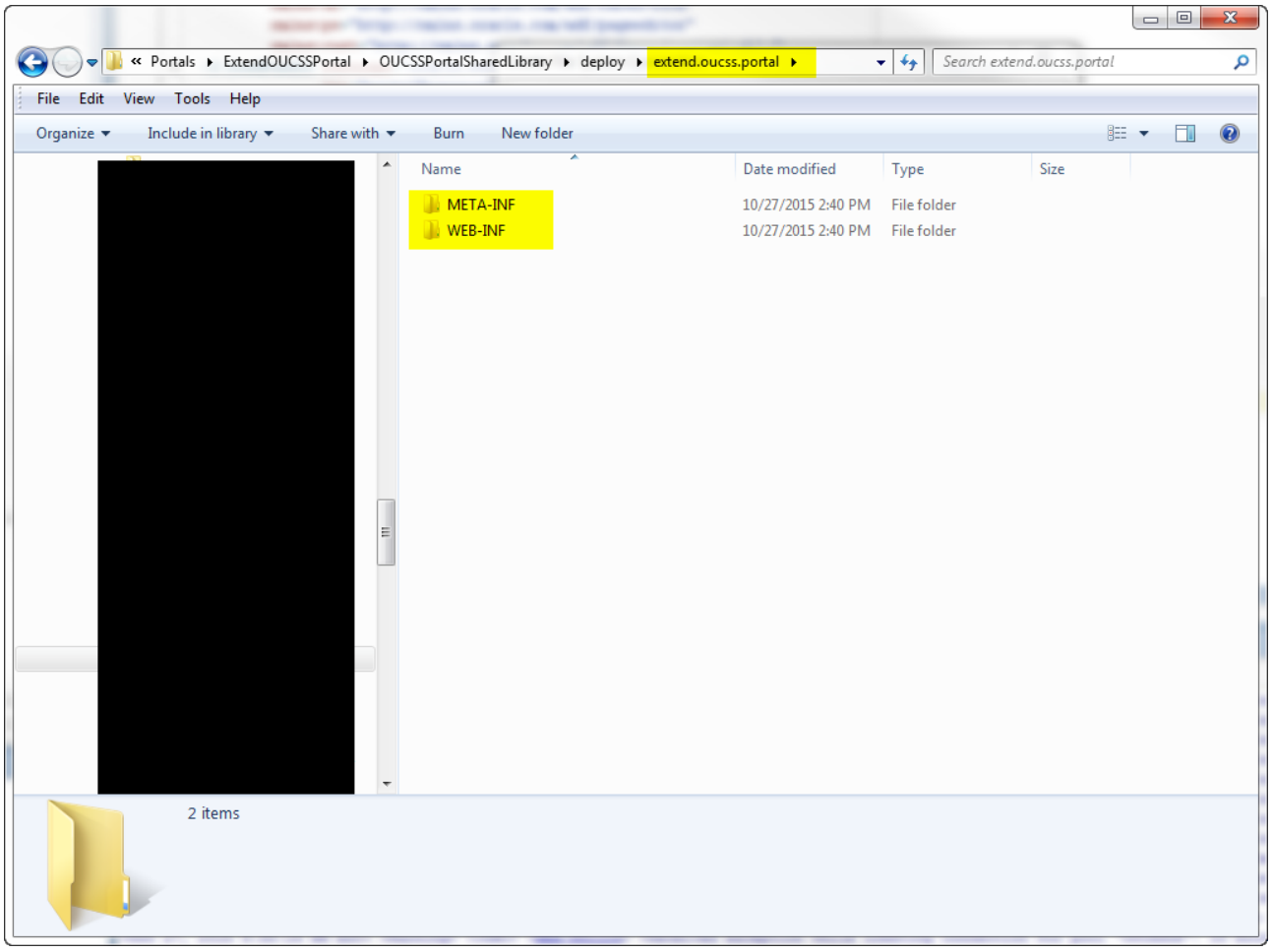
8 Edit the weblogic.xml file in the folder `Web Content/WEB-INF`, creating library references to OUCSS portal shared libraries.



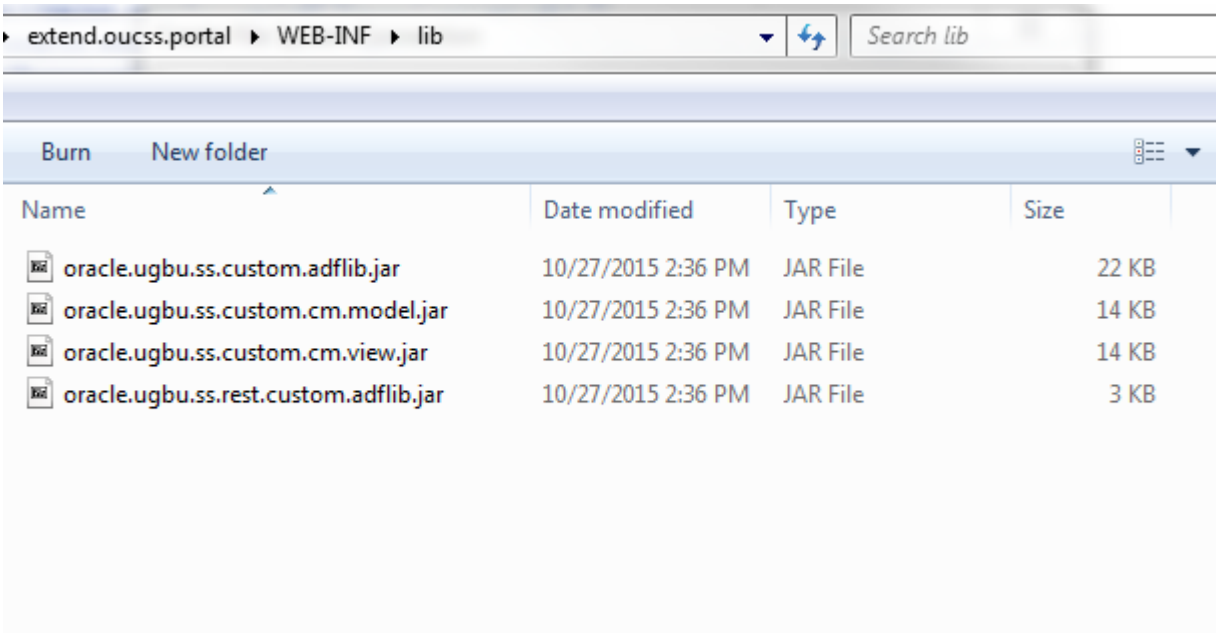

```
<library-ref>
  <library-name>com.oracle.ugbu.ss.lib</library-name>
</library-ref>
<library-ref>
  <library-name>com.oracle.ugbu.ss.residential.lib</library-name>
</library-ref>
<library-ref>
  <library-name>extend.oucss.portal</library-name>
</library-ref>
<library-ref>
  <library-name>com.oracle.ugbu.ss.commercial.lib</library-name>
</library-ref>
```

9 To consume the task flow or pages created in the portal extension project, a file link needs to be created to local file system storing ADF library jars which contain the custom taskflows.

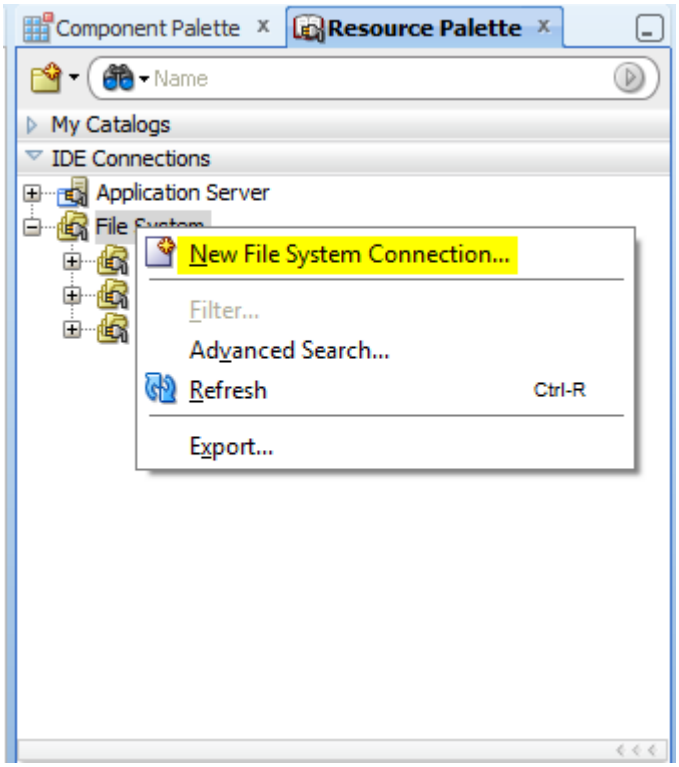
10 Navigate to the folder containing portal extension .war created after deploying ExtendOUCSSPortal project. Explode the .war into a folder in your file system.



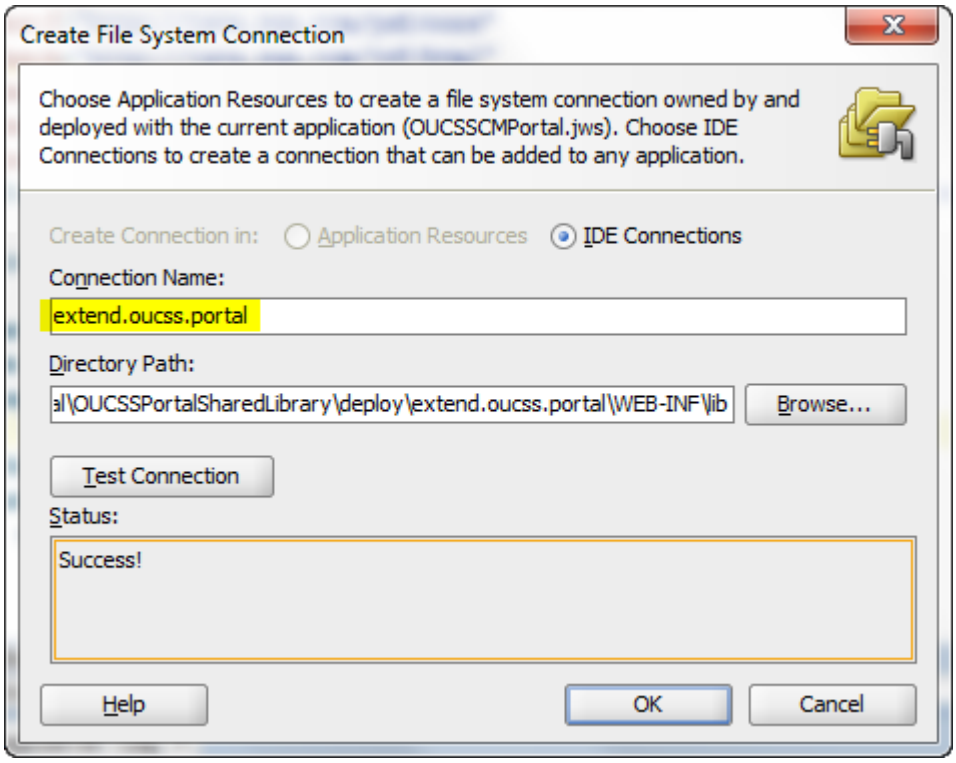
11 Navigate to WEB-INF/lib and verify all portal extension jars are present in this folder.



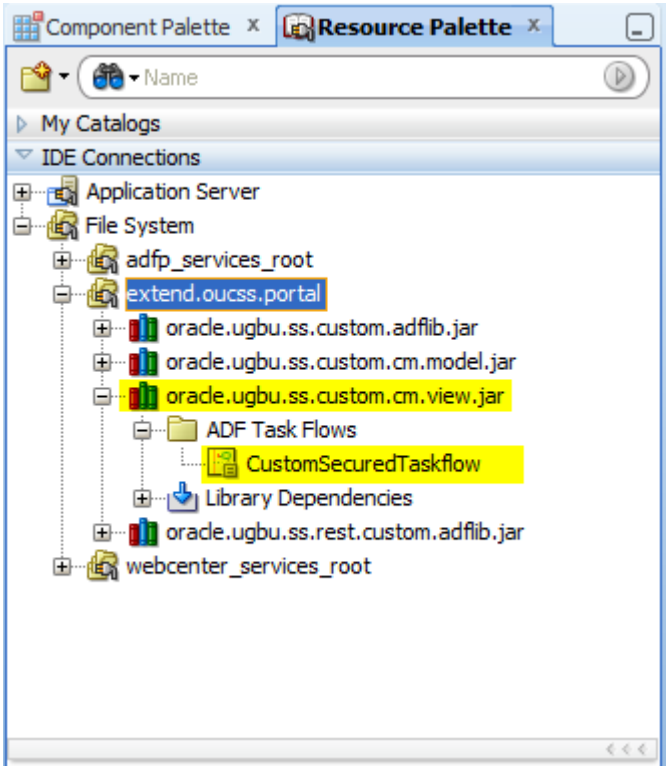
12 Right click on File System under Resource Palette and select 'New File System Connection'



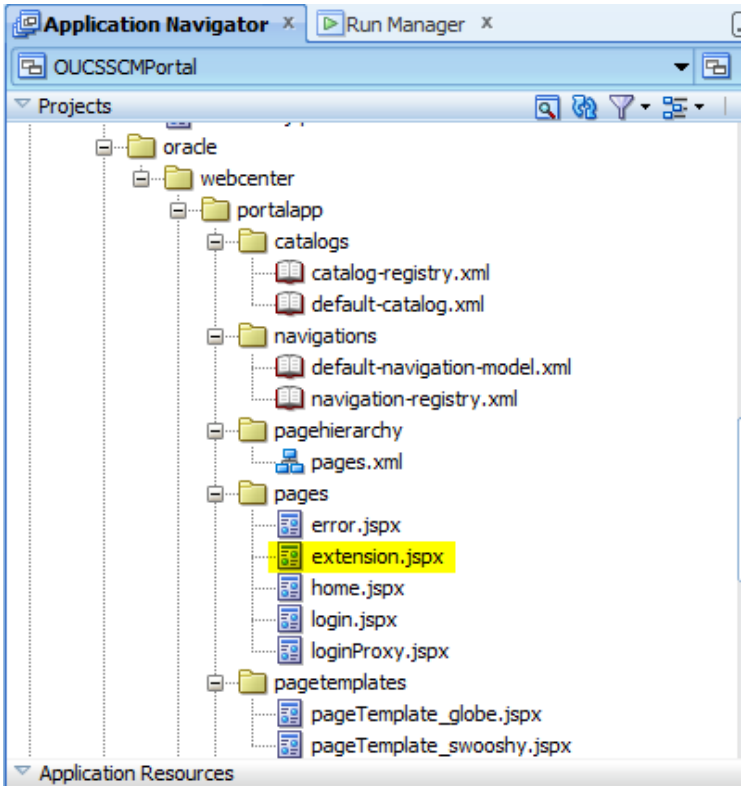
13 Enter Connection Name as 'extend.oucss.portal' and directory path should be the folder containing ADF library jars from portal extension project.



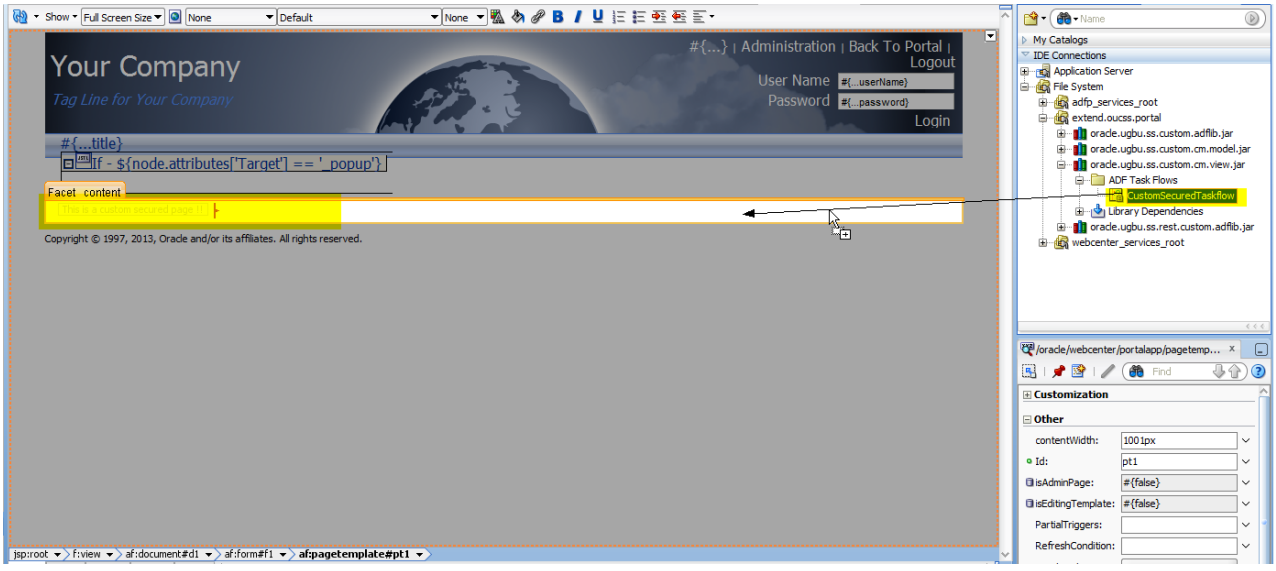
14 Expand the file link under file system in resource palette and verify the custom portal artifacts are present in ADF library.



15 For this illustration, we will test the CustomSecuredTaskFlow within custom portal extension project (note that this task flow is not shipped out of the box, this has been created only for illustration). We will create a portal page (e.g. extension.jspx) to serve as a placeholder to test this task flow. Please review 'Adding New Taskflows and Pages' section of this document for additional details on how to create a page and page definition file.

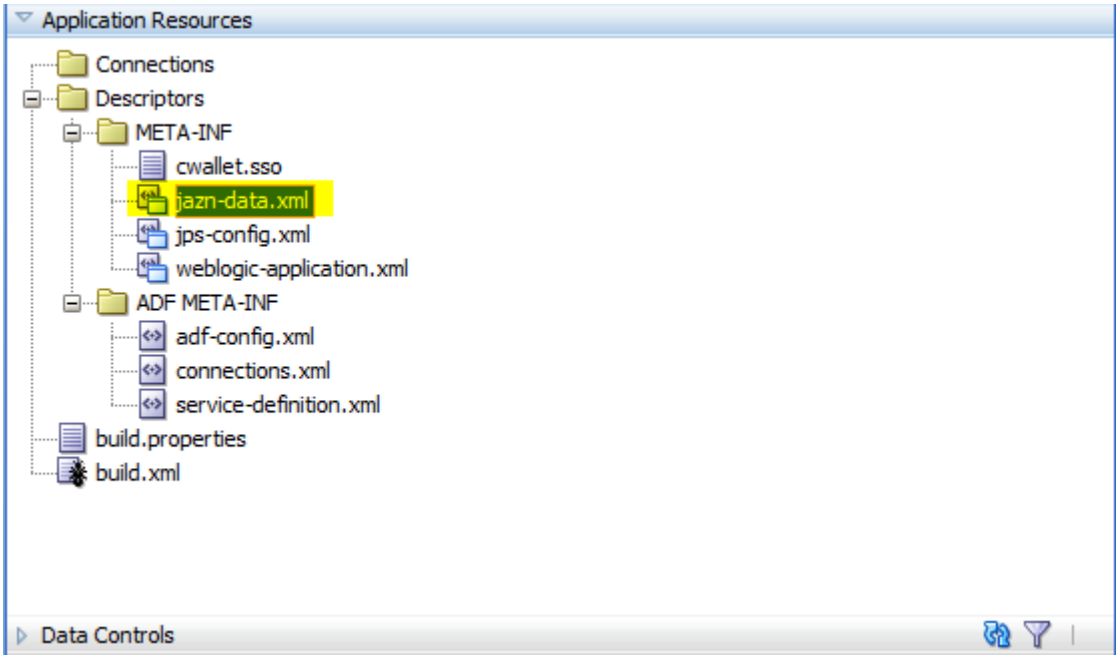


16 Drag and drop **CustomSecuredTaskFlow** into the portal page. To see the taskflow runtime, deploy the custom portal application and navigate to the portal page.



17 Proper permissions need to be granted to pages and taskflows to enable display in the portal. Please review the About Security section in this document and follow the recommended folder structure when creating custom taskflows and pages. The security settings in the custom portal needs to be updated to follow the same convention as well.

18 To update portal security settings, edit **jazn-data.xml** in the META-INF folder via the **Application Resources** window.



19 Find the principal name **authenticated-role** and add the permissions below. Do not change the other permissions for this principal.

```
<grantee>
  <principals>
<principal>
  <class>oracle.security.jps.internal.core.principals.JpsAuthenticatedRoleImpl</class>
  <name>authenticated-role</name>
</principal>
  </principals>
</grantee>
<permissions>
  <permission>
    <class>oracle.adf.controller.security.TaskFlowPermission</class>
    <name>/WEB-INF/oracle/ugbu/ss/custom/secure/.*</name>
    <actions>view</actions>
  </permission>
  <permission>
    <class>oracle.adf.share.security.authorization.RegionPermission</class>
    <name>oracle.ugbu.ss.custom.pages.secure.*</name>
    <actions>view</actions>
  </permission>
```

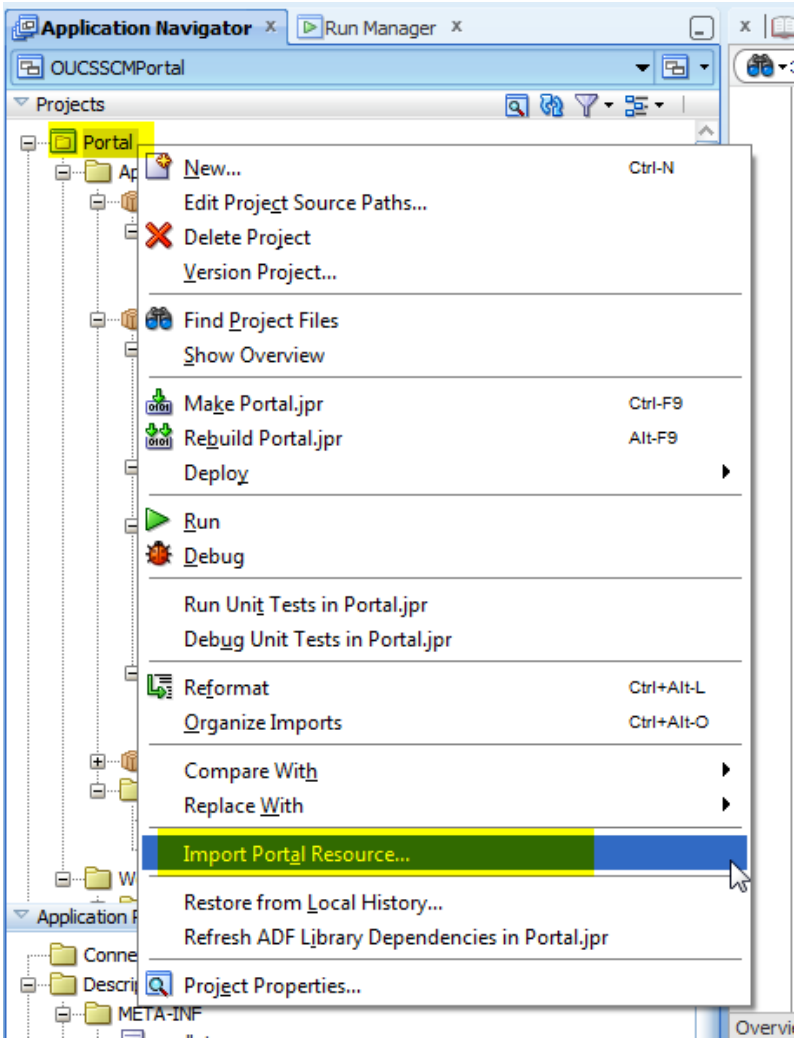
20 Find the principal name **anonymous-role** and add the permissions below. Do not change the other permissions for this principal.

```
<grantee>
  <principals>
<principal>
  <class>oracle.security.jps.internal.core.principals.JpsAnonymousRoleImpl</class>
  <name>anonymous-role</name>
</principal>
  </principals>
</grantee>
<permissions>
  <permission>
```

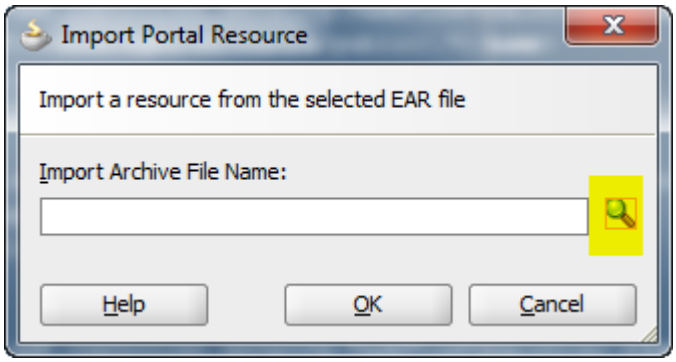
```
    <class>oracle.adf.controller.security.TaskFlowPermission</class>
    <name>/WEB-INF/oracle/ugbu/ss/custom/public/.*</name>
    <actions>view</actions>
  </permission>
  <permission>
    <class>oracle.adf.share.security.authorization.RegionPermission</class>
    <name>oracle.ugbu.ss.custom.pages.public.*</name>
    <actions>view</actions>
  </permission>
```

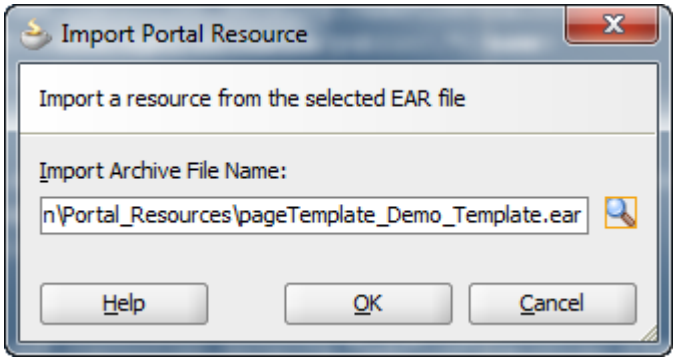
21 Export the portal resources (e.g. templates, skin, etc.) from the base portal and import them into the local portal instance. This way, local portal pages will have same look and feel as the base portal. See the Portal Rebranding section for the precedure describing how to Export portal resources from the base portal.

22 Right click the **Portal** project in JDeveloper and select **Import Portal Resources**.



23 A pop-up dialog for importing portal resources is displayed. Click on the Search icon and select the exported portal resource (in .ear format) that was saved in your file system.

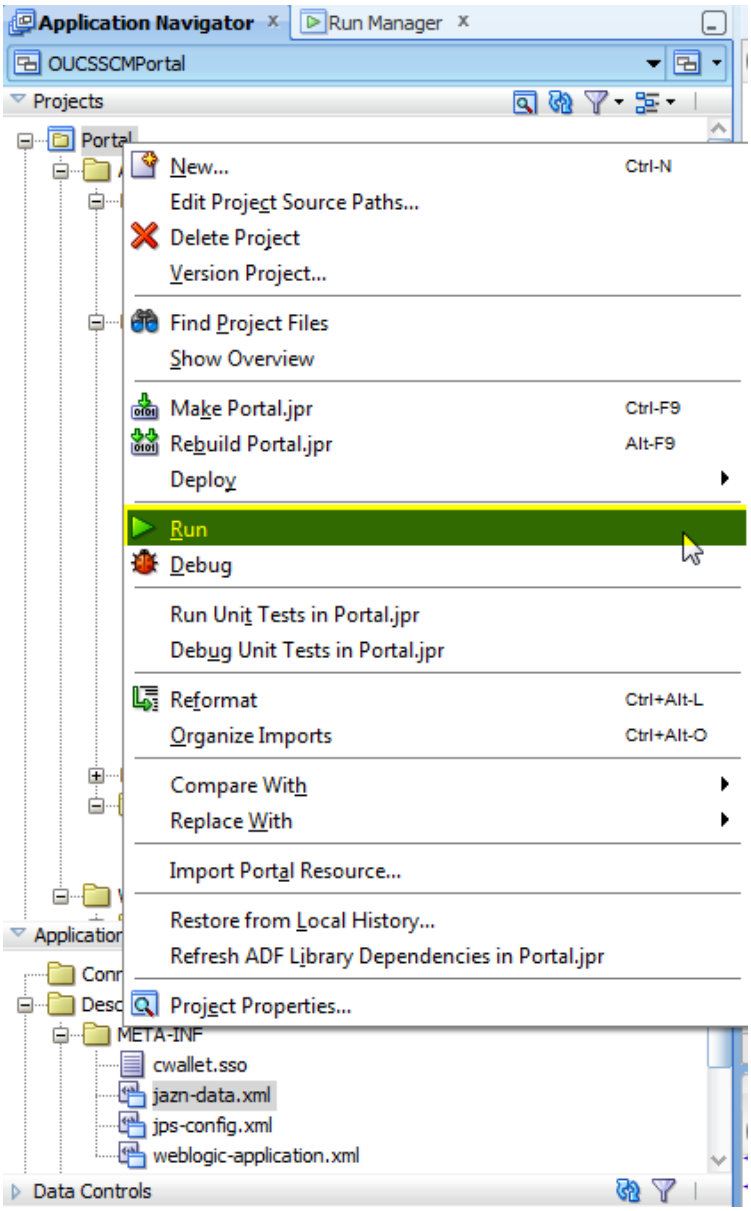


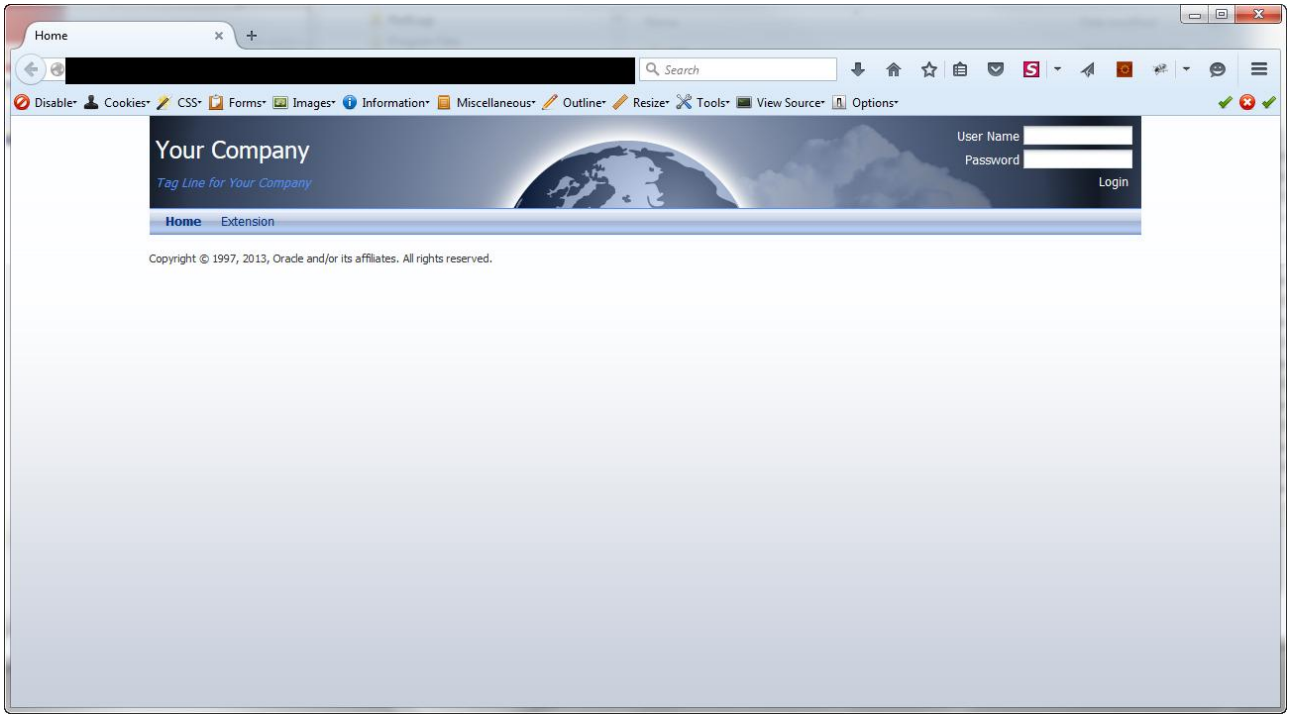


24 Repeat the previous step for importing all portal resources from the base portal.

25 Right-click the **Portal** project again and selct **Run**. This should bring up the local portal instance. It is assumed that the developer has created/deloyed the following artifacts in the DefaultDomain of the Integrated WLS Server when running the portal:

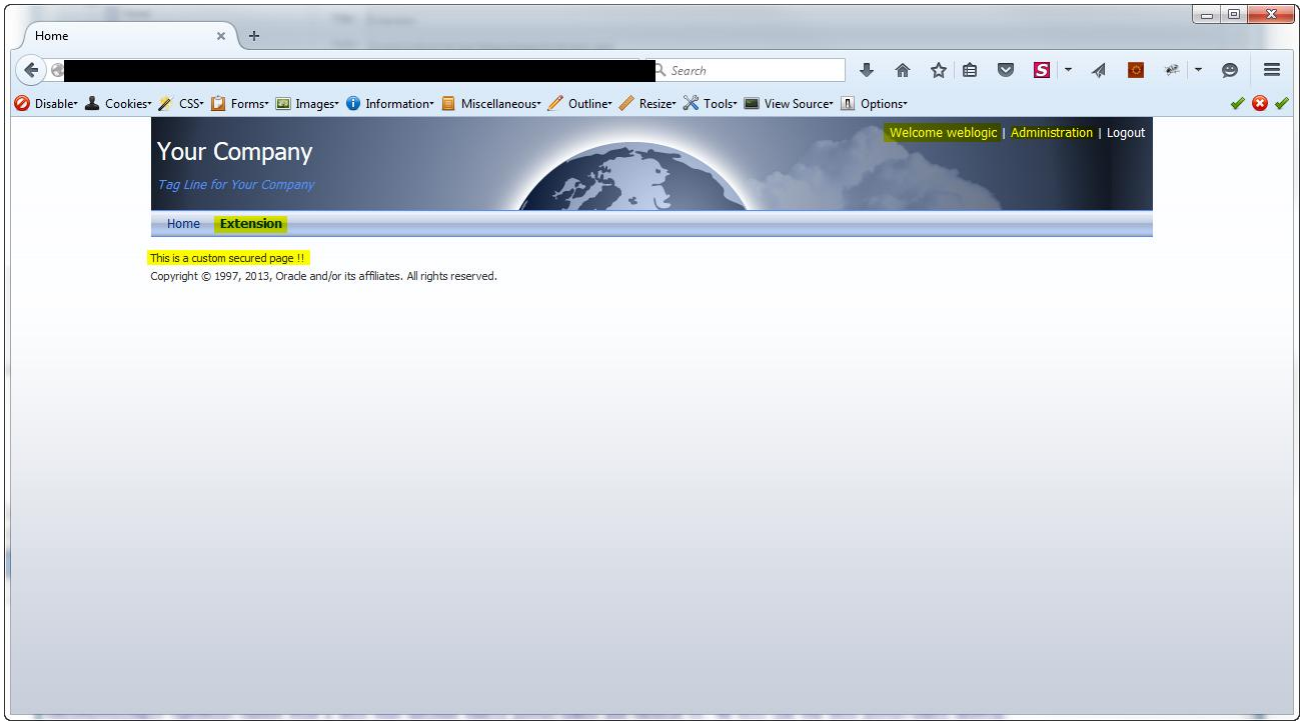
- Data Source with OUCSSDS and JNDI jdbc/OUCSSDS pointing to the OUCSS schema.
- All OUCSS shared libraries that were included in weblogic.xml (step 8), along with the version of **extend.oucss.portal.war** that needs to be tested.
- The OUCSS Mail Session connection, in case sending email is to be tested.





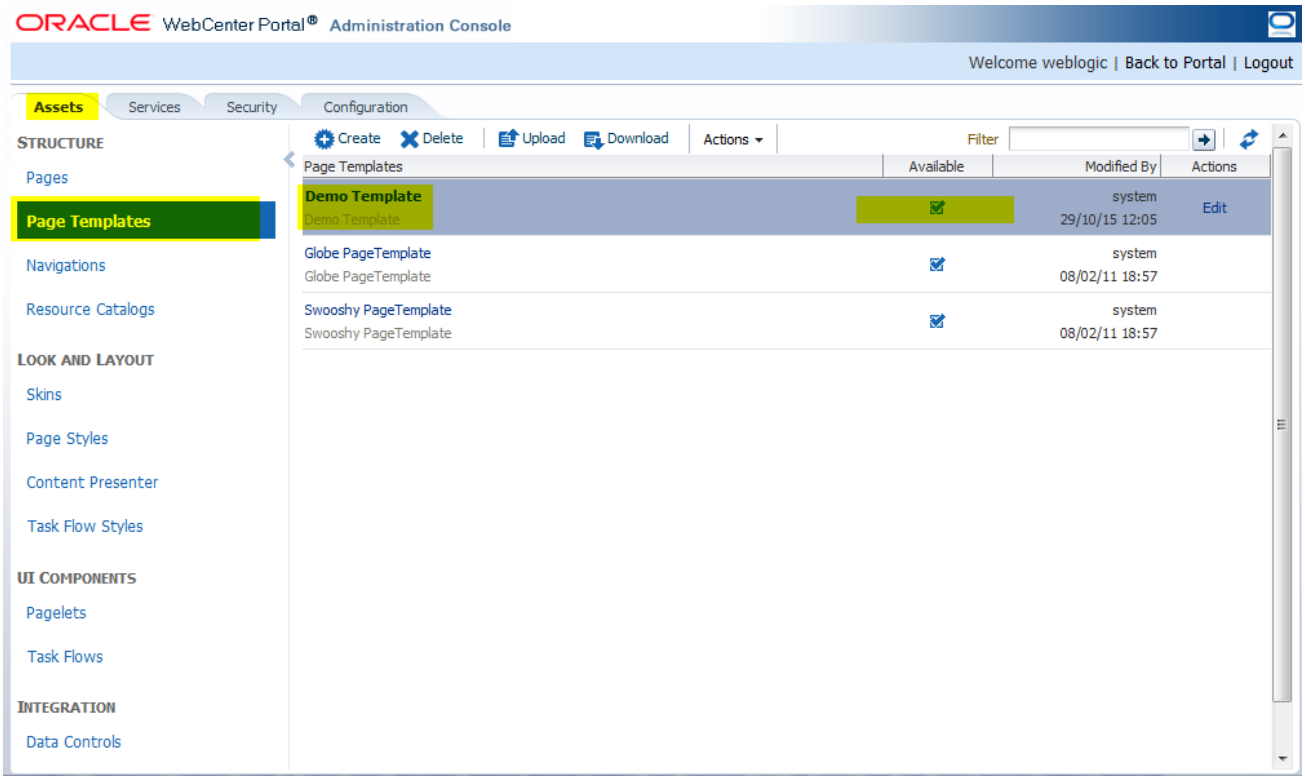
26 Log in to the portal to view the secured content. The login user should be present in the integrated WLS LDAP. After logging in, navigate to the portal page that contains the secured taskflow. For illustration purpose, a portal page **Extension** has been created in JDeveloper and added to the default navigation. The secured task flow being tested is added to this page.

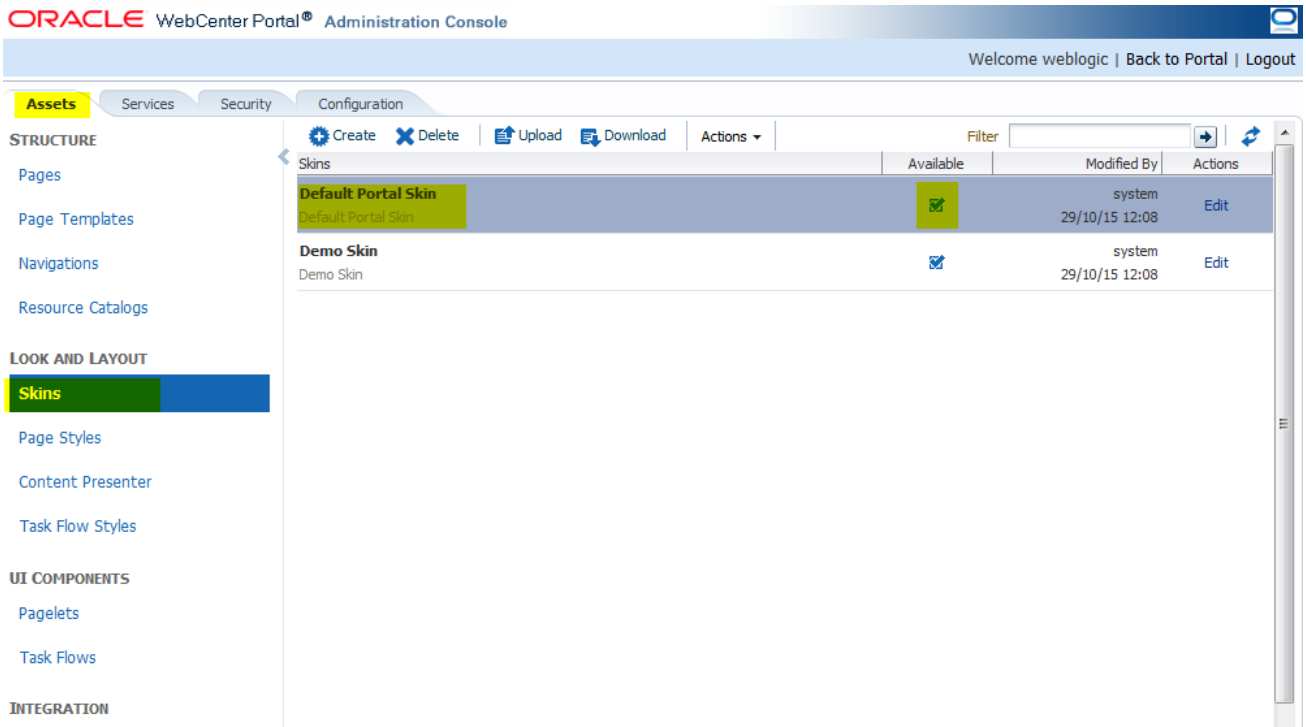
27 Navigate to the **Extension** page after logging into the portal. Notice that the content of the secured taskflow is displayed on the page.



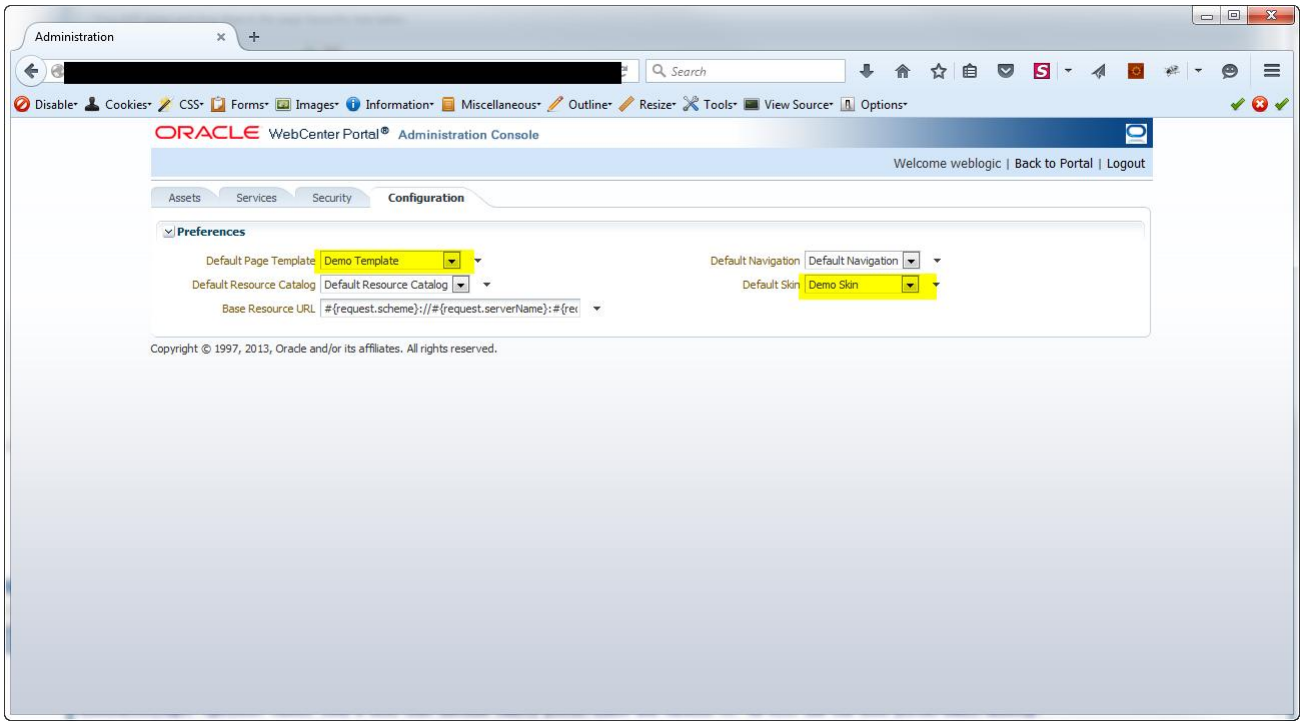
28 Note that the portal look and feel shows the default branding. To change the look and feel to the base portal, go to the **Administration** page using the link at the top-right corner.

29 Verify that the imported portal resources are all checked as **Available** in the **Assets** category.

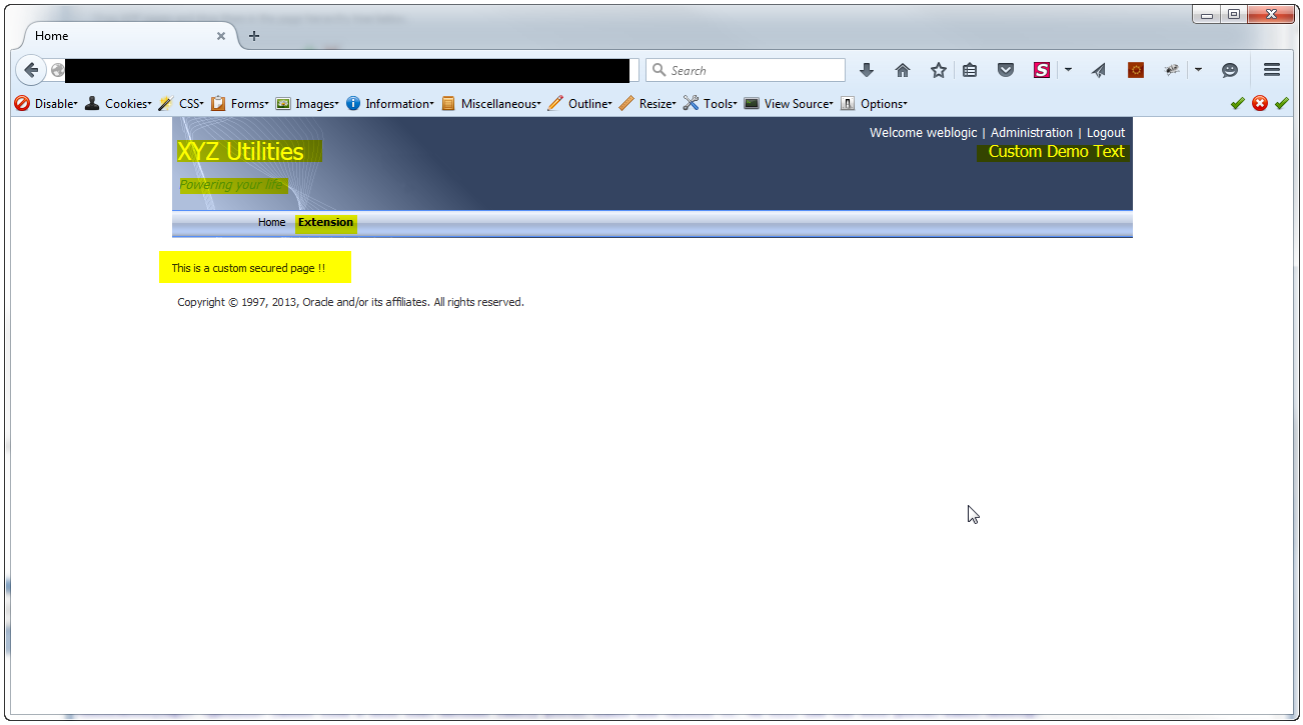




30 Go tothe **Configuration** tab under **Administration** and select the portal resources from the base portal.



31 Select **Back to Portal** on the top-right of the administration page to navigate back to the portal. Notice the portal branding is now changed to conform to the base portal. Go to the **Extension** page to view the taskflow being tested.



Portal Rebranding

To rebrand the portal, implementation tasks include:

- Customize the Page Template
- Customize the Navigation Model
- Customize the Resource Catalog
- Customize the Skin

Customizing the Page Template

The default OUCSS Portal contains Global and Swooshy site templates from the Oracle WebCenter Framework Application template. The default page templates can be replaced by:

- Customizing the existing page templates
- Creating a brand new page template.
- Importing new page templates exported from another instance or JDeveloper.

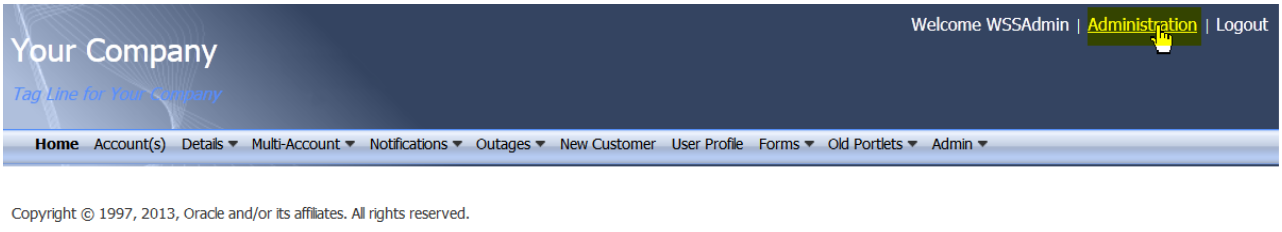
For more information on default and page template customization, see
http://docs.oracle.com/cd/E29542_01/webcenter.1111/e63259/jpsdg_pagetemp.htm

Customizing an Existing Page Template

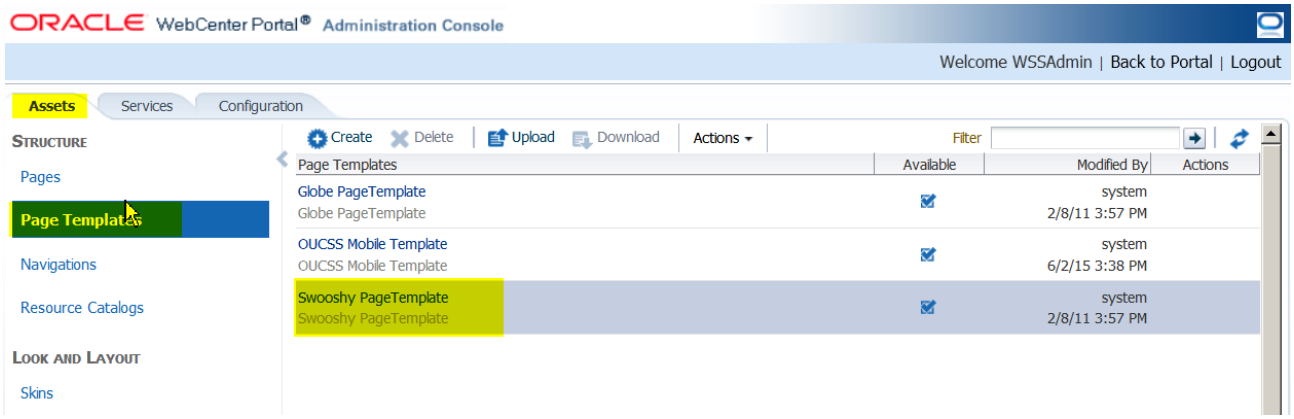
The following steps describe how to customize the Swooshy page template that exists on a deployed OUCSS Portal.

1 Log in to the OUCSS Portal with WebCenter Administrator permission (WSSAdmin user is seeded with the required permission).

2 Click the **Administration** link on the top-right section of the screen.

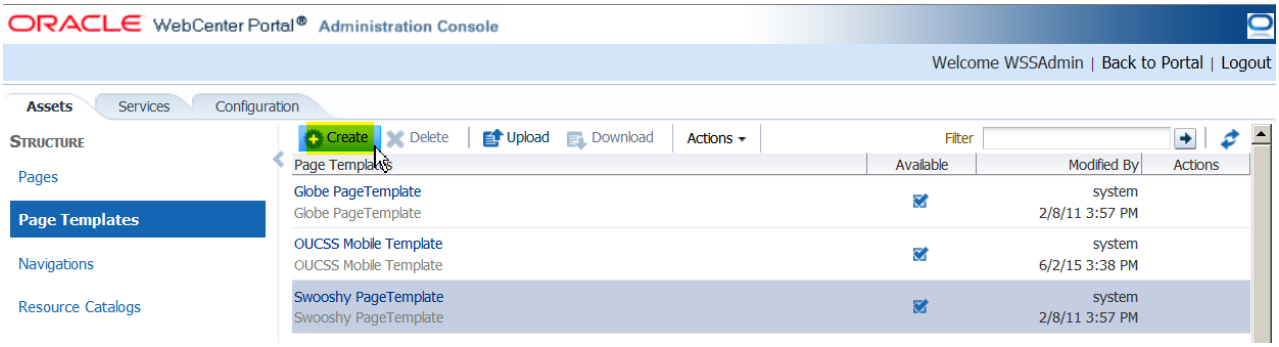


3 In the **Administration** console, click the **Resources** tab, then choose **Page Templates** in the **Structure** section:



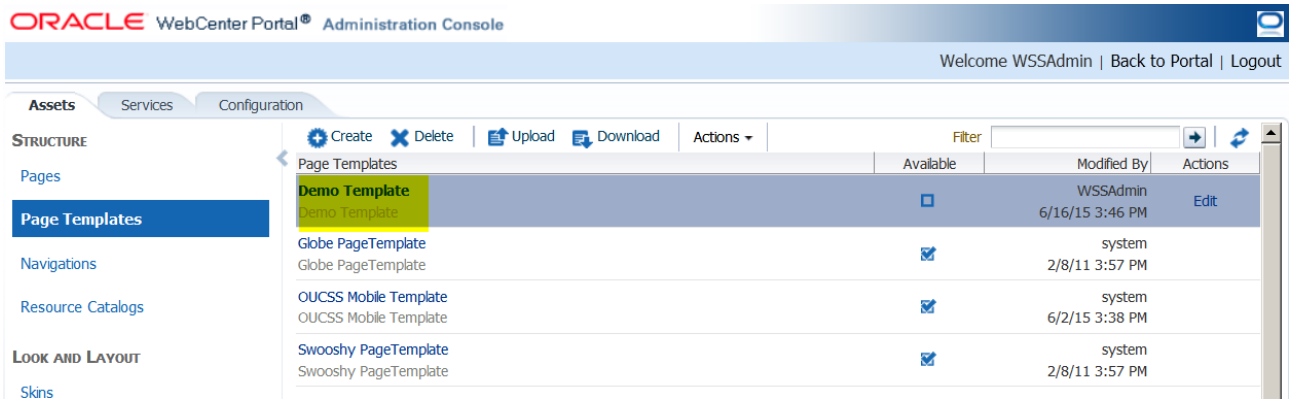
4 The **Swooshy Page Template** is provided with the product and is set to read-only. To create a new customized template based on the Swooshy template:

A Click **Create**.

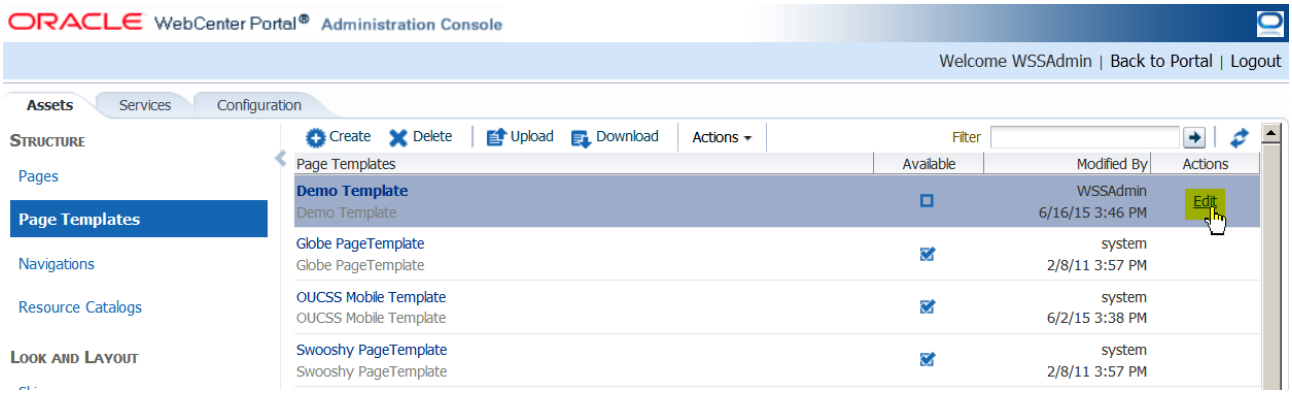


B Enter a name and description (optional), then select the **Swooshy PageTemplate** from the **Copy From** list box:

C Click **Create** to create the new template.



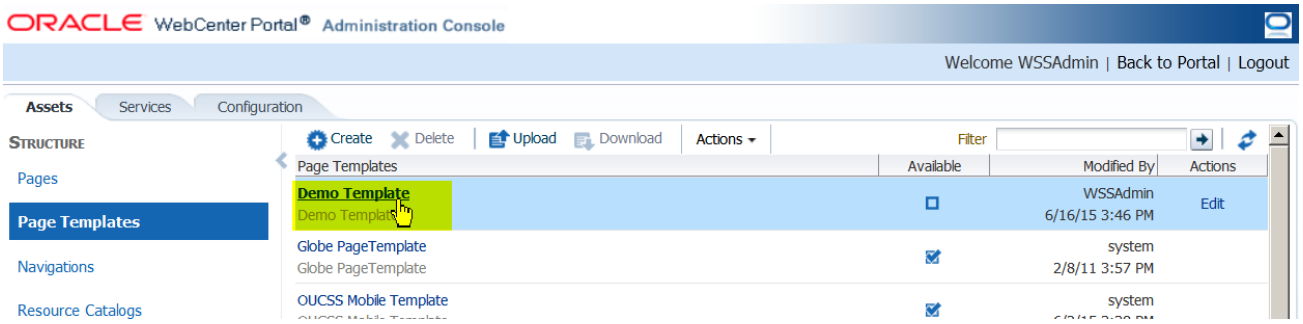
5 Select the new template, then click the **Edit** menu and choose **Edit** to open the template for customization.



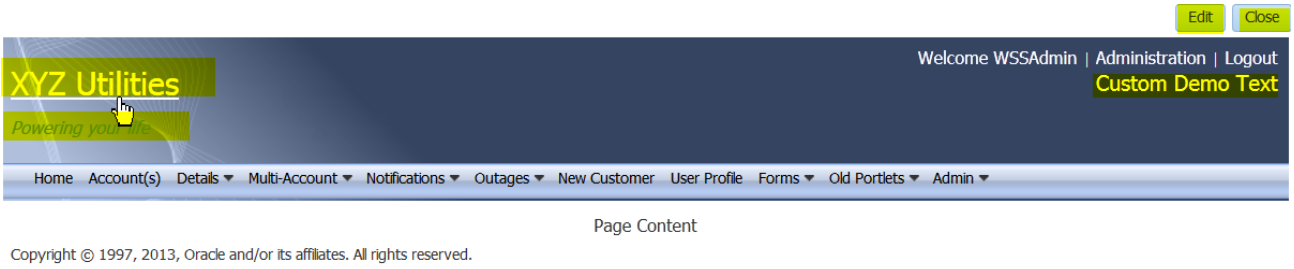
Demo Template in Edit Mode

6 Customize as required, and then click **Save**, and then **Close**.

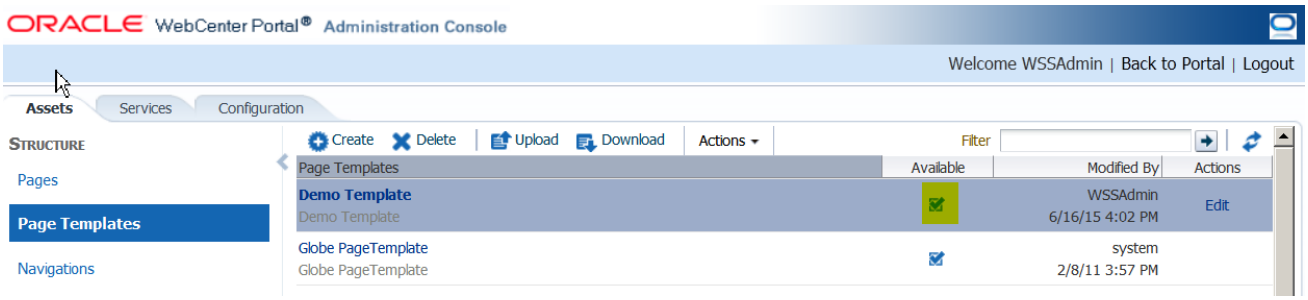
7 Click on the page template name to preview the changes.



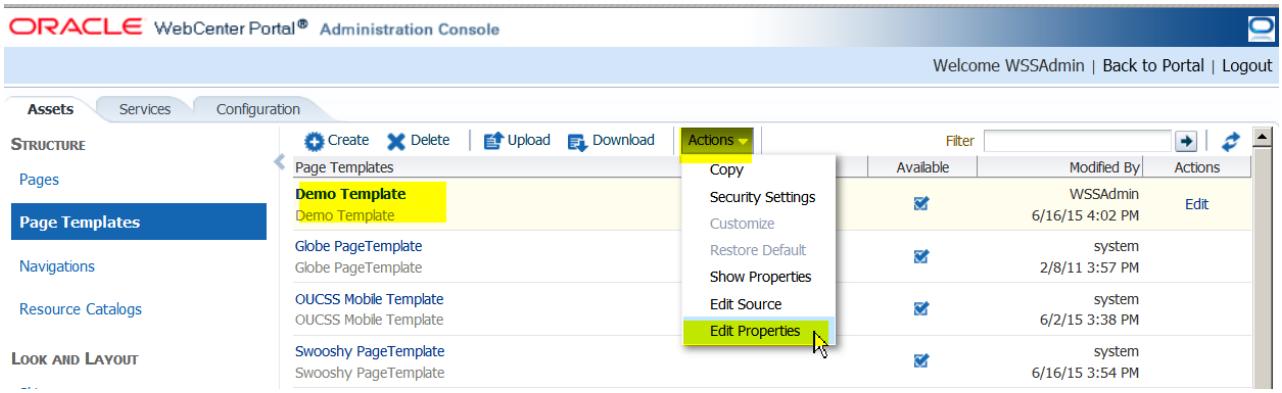
8 Click **Edit** to further modify the template. When complete, click **Save** and **Close**.



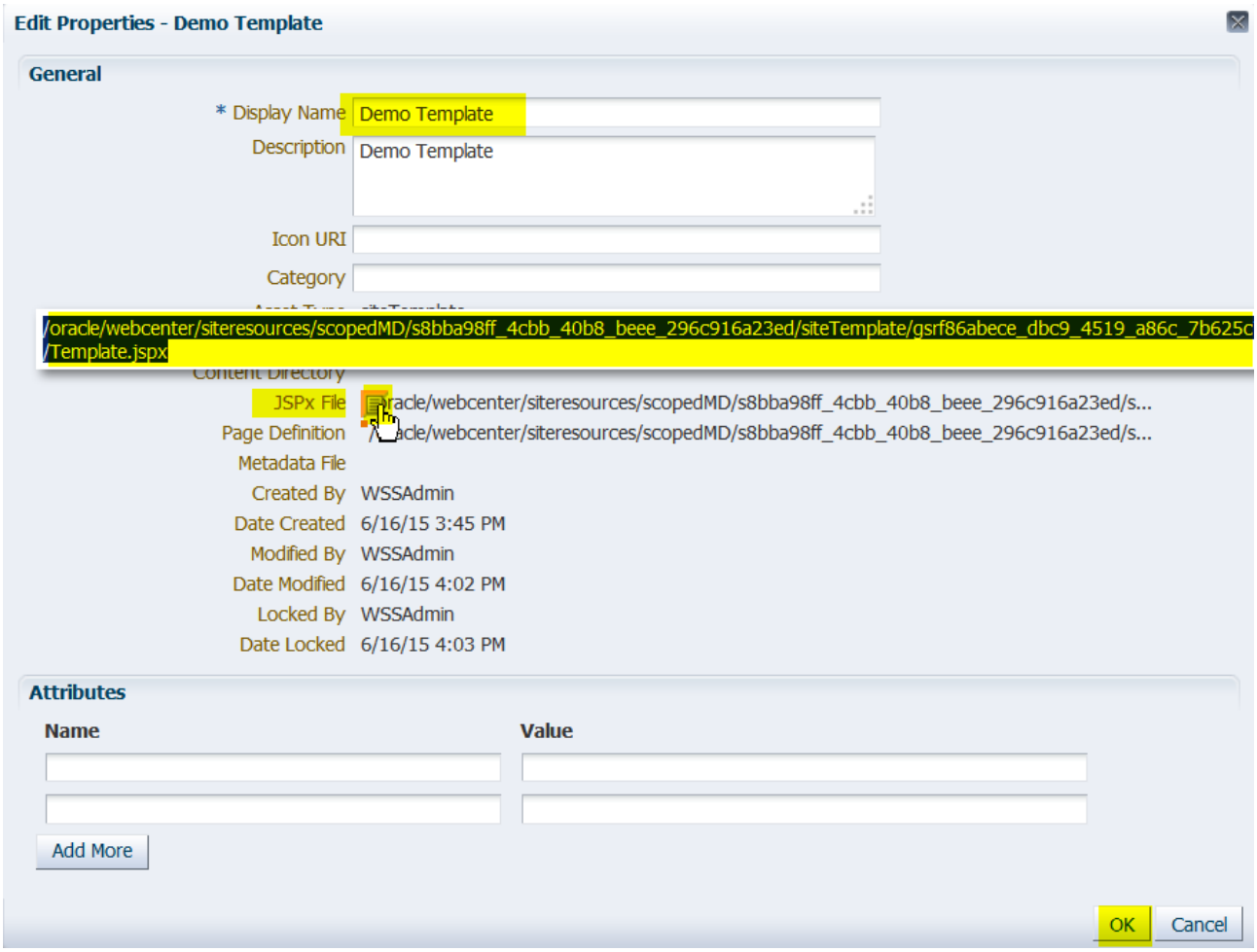
9 The newly created templates are hidden by default. To unhide the template, check the **Available** checkbox.



10 To activate and use the new Demo Template as the site template for one or more contexts, copy the template path of the new template, then choose **Action** > **Edit Properties**.



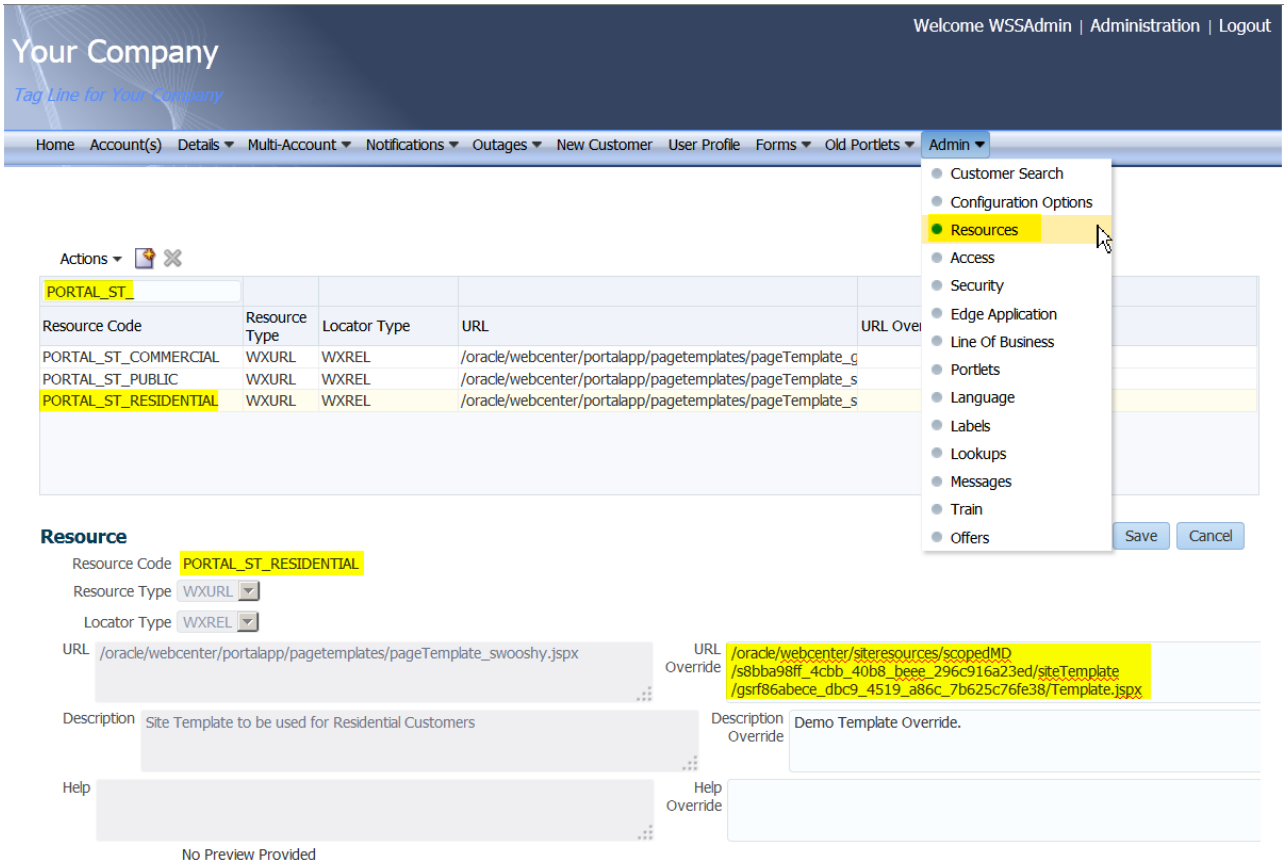
11 Copy the path from the **JSPx File** field, then click **Cancel** to close the popup.



12 Click the **Back to Portal** link and go to the **Admin** > **Resources** page.

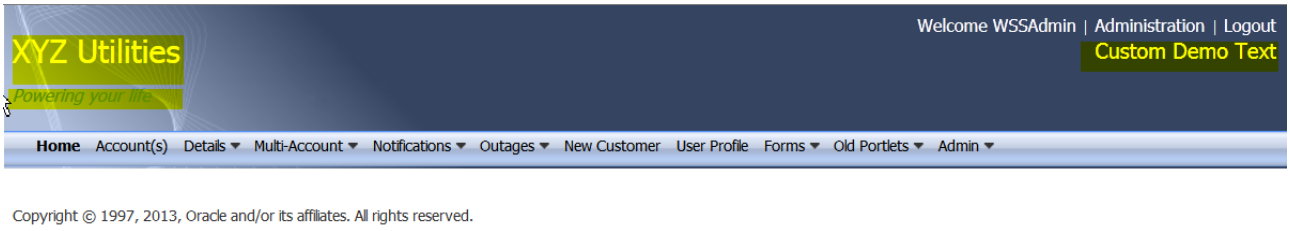
13 Filter the resources with resource codes **PORTAL_ST_**. Update the **URL Override** property by selecting a **PORTAL_ST_<<Context>>** (**RESIDENTIAL, PUBLIC**, etc.) resource with the JSPx path that was copied in step 10.

Note: To update **Mobile** page template, search for the appropriate resource code, e.g., `MOB_B_PORTAL_ST`.



14 Repeat the process to update other contexts with a custom **Site Template**.

15 Restart the OUCSS Portal managed server and verify that the changes are reflected in the portal.



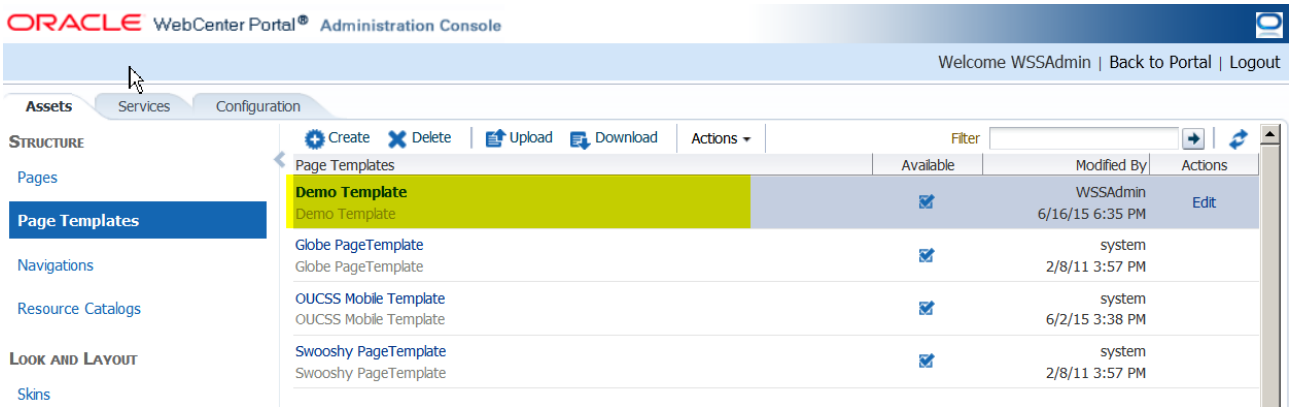
Exporting Page Templates

Page templates from other WebCenter portal instances, or from JDeveloper, can be exported and then imported into the OUCSS Portal for rebranding.

To export an existing template:

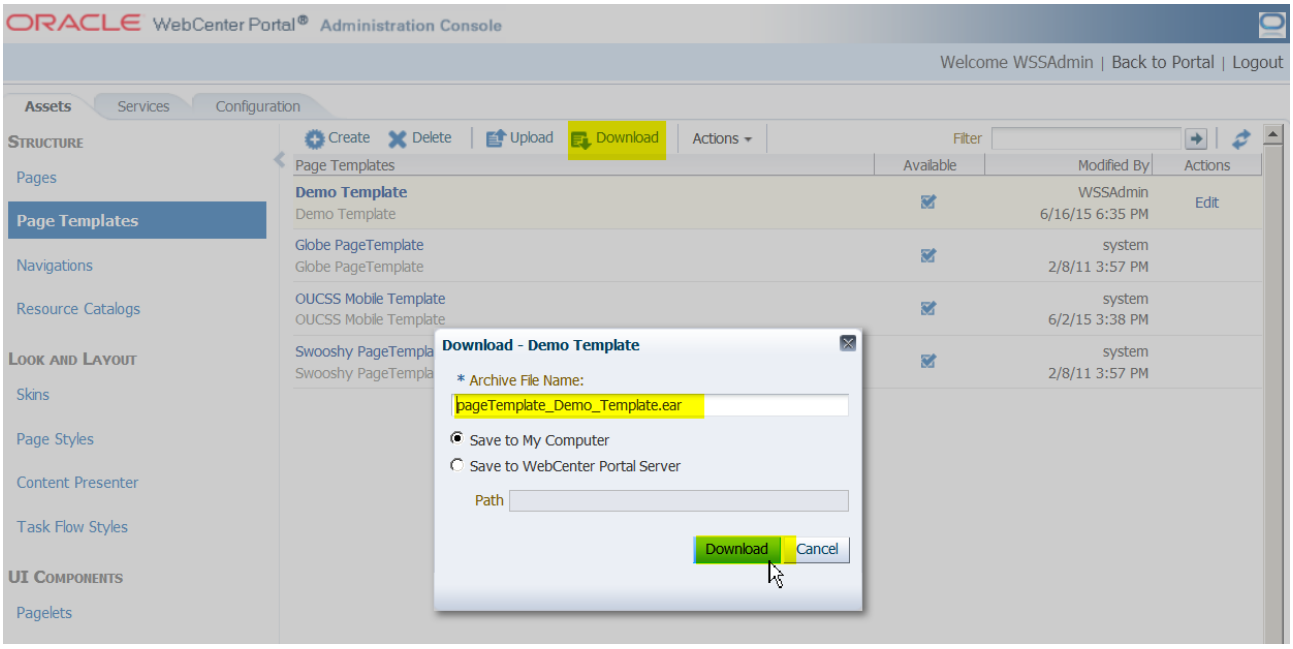
1 Log in to the instance of the existing page template with Administrator privileges, and then click on the **Administration** link to launch the Administration Console.

2 Choose the **Resources** tab, then click the **Page Templates** link in the **Structure** section. Select the template to be migrated.

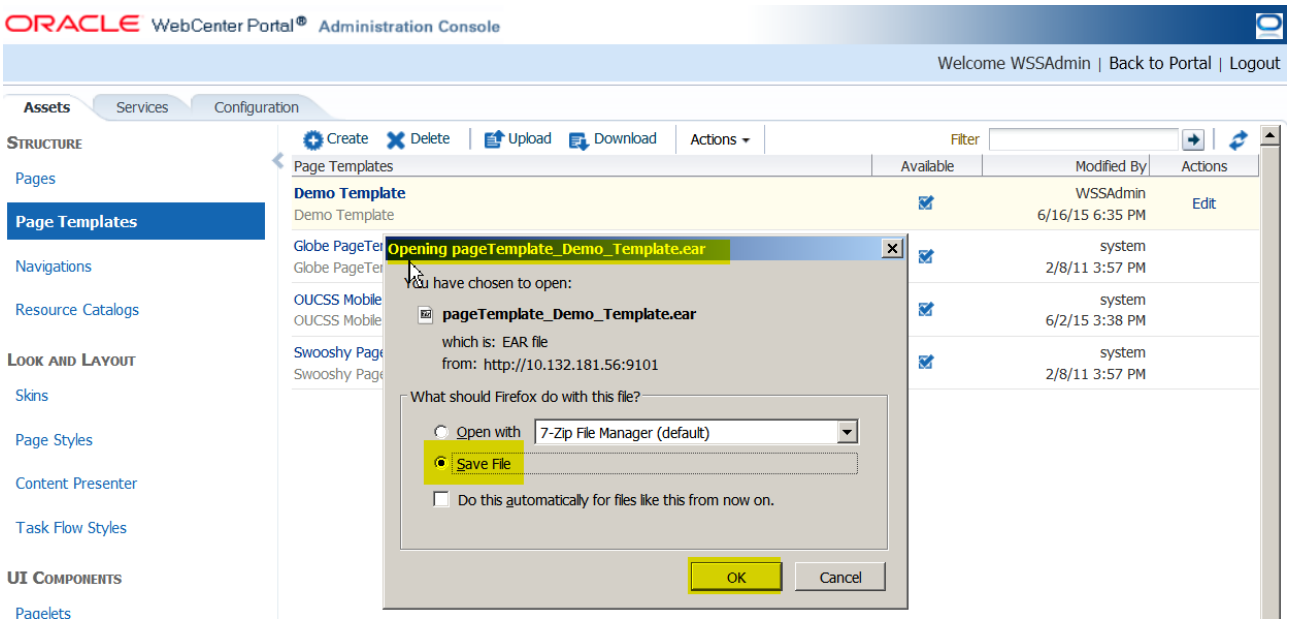


3 Click **Download** on the toolbar to open the **Download** dialog.

4 Enter the **Archive File Name**, choose the location for the exported archive on your local machine, then click **Download**.

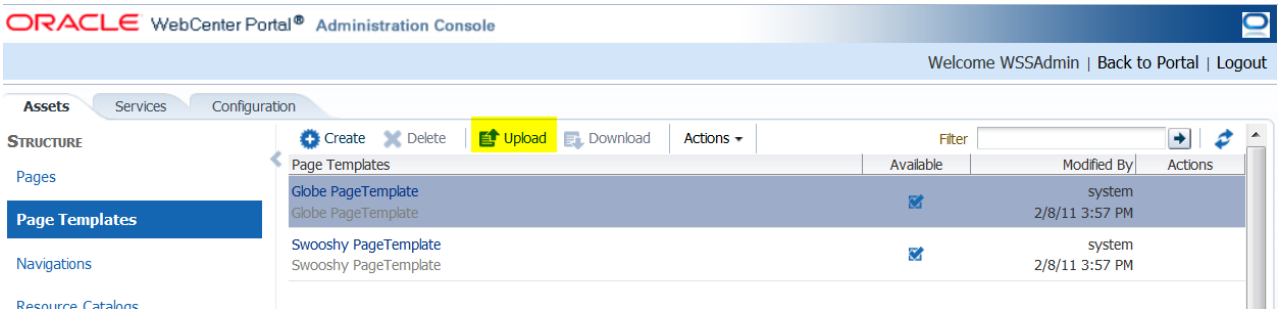


5 In the **Opening** dialog, choose **Save File** to save the exported template, then click **OK**.

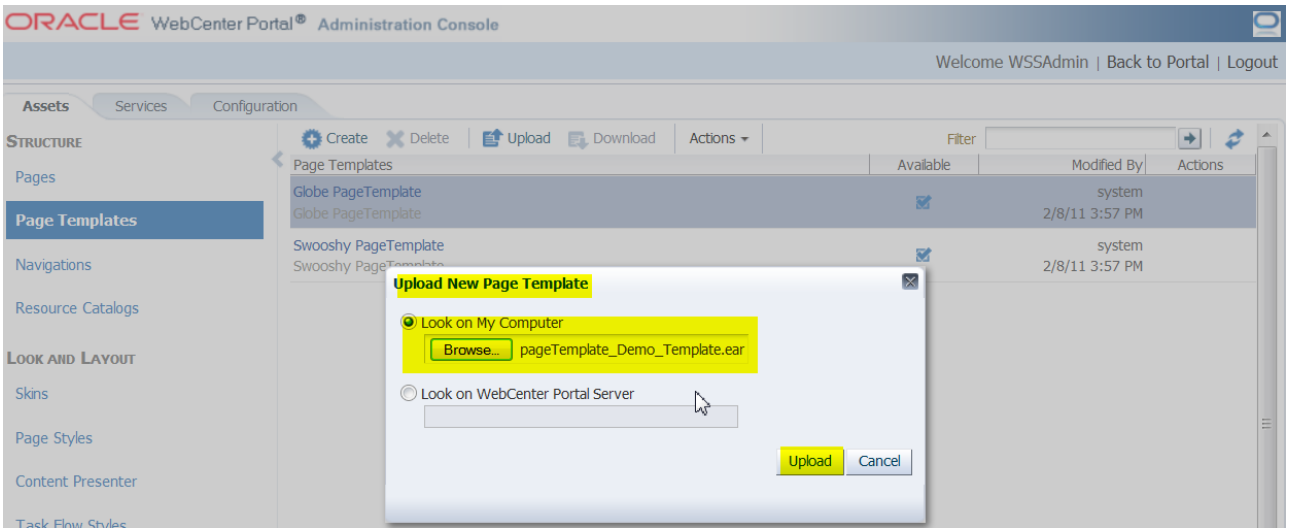


Importing Page Templates

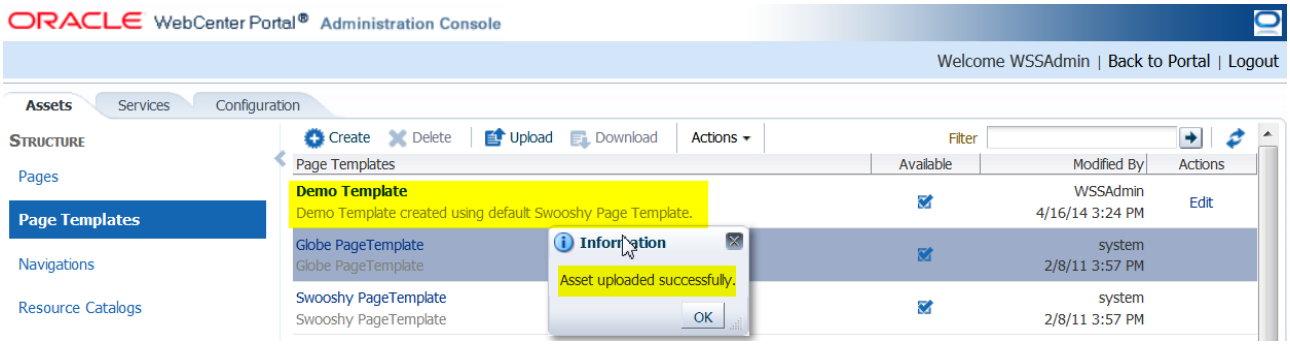
1. Log in to the portal instance with Administrator privileges in the location in which the template is to be migrated/imported.
2. Click **Administration** and go to the **Resources** > **Page Templates** page.
3. Click **Upload**.



4. Select the **pageTemplate_Demo_Template.ear** or **Page Template** export archive that was downloaded from the previous portal instance, then click **Upload**.



5 Click **OK** to confirm. The newly-uploaded **Page Template** should be visible on the list.



6 To configure the page template, follow steps 10 to 15 from the topic Customizing an Existing Page Template.

Customizing the Navigation Model

One of the primary activities in rebranding the portal is to customize the navigation model. This procedure involves adding new links and/or deleting or renaming existing links. If the provided out-of-the-box navigation model requires many changes, then a new navigation model can also be created from scratch using JDeveloper. The new navigation model can then be exported from JDeveloper and imported into a Portal instance. Alternatively, a navigation model can be created using the **Assets** page in the WebCenter Administrator Console.

For additional information on customizing the navigation model, see
http://docs.oracle.com/cd/E29542_01/webcenter.1111/e63259/jpsdg_navigation.htm

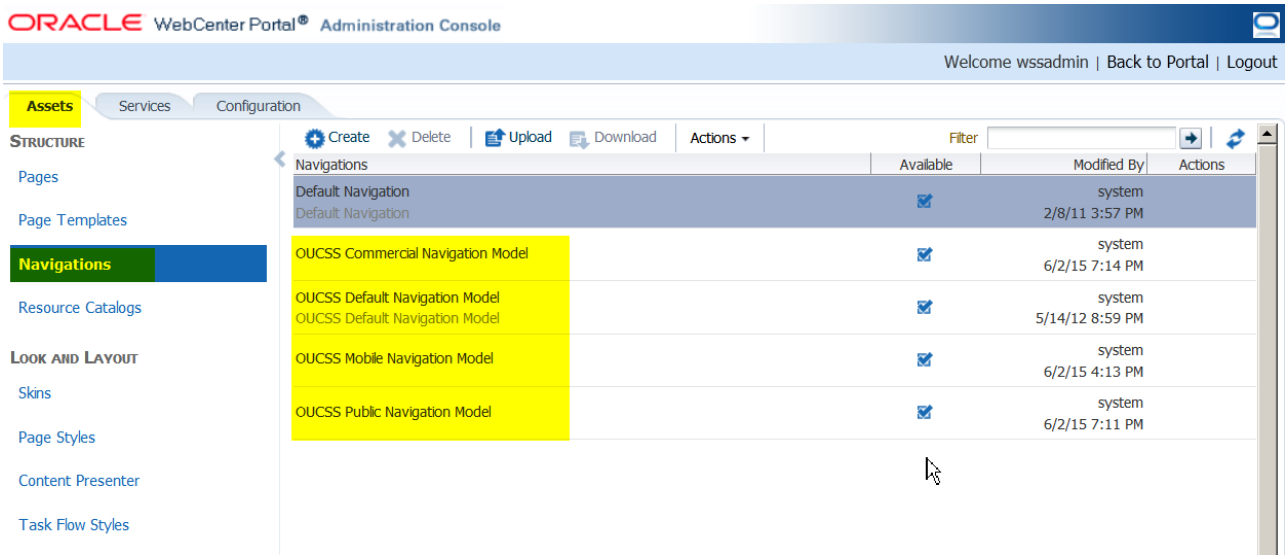
Customizing Provided/Creating New Navigation Models

The default OUCSS Portal is configured with four different navigation models. These navigation models are seeded and are read-only.

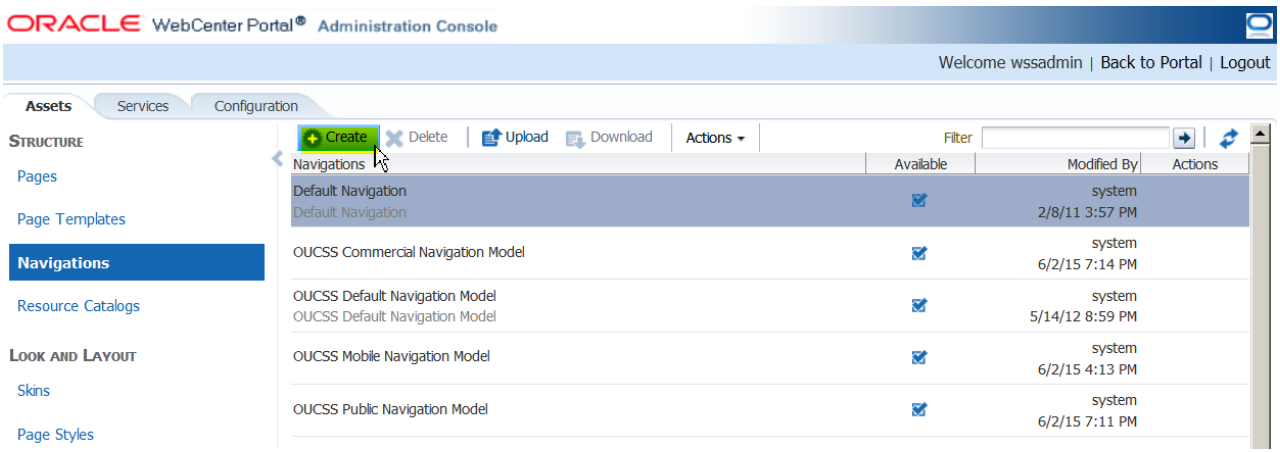
To customize provided navigation models:

1 Log in to the OUCSS Portal with Administrator privileges and go to the Administration Console.

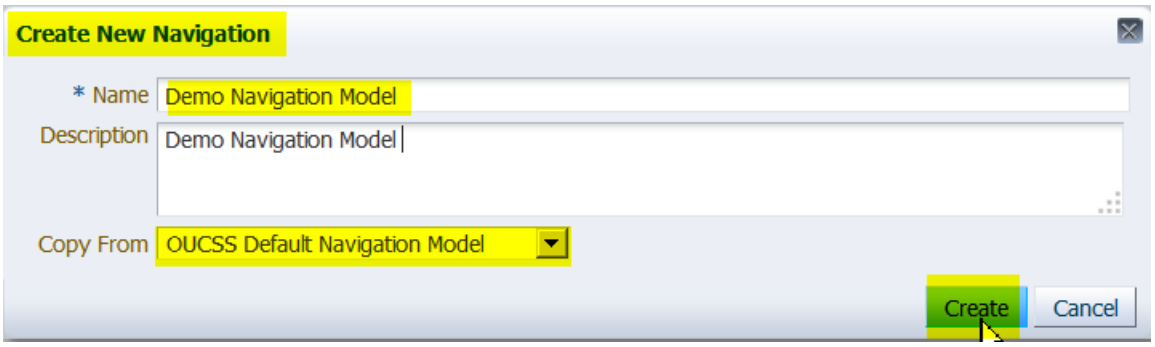
2 Choose the **Assets** tab, and then click the **Navigations** link in the **Structure** section.



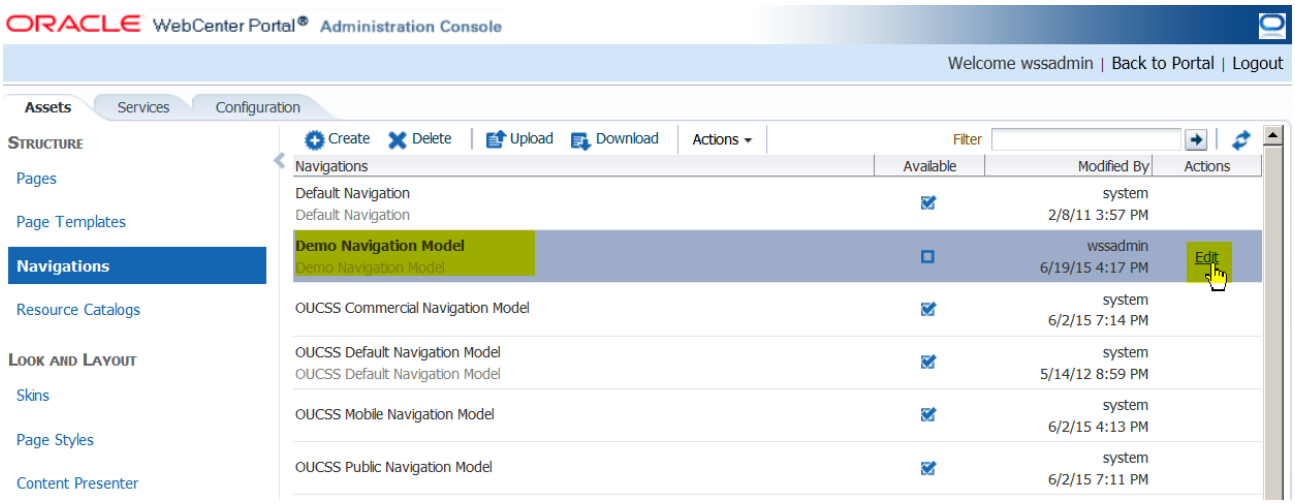
3 Click **Create** to create a new **Navigation Model**.



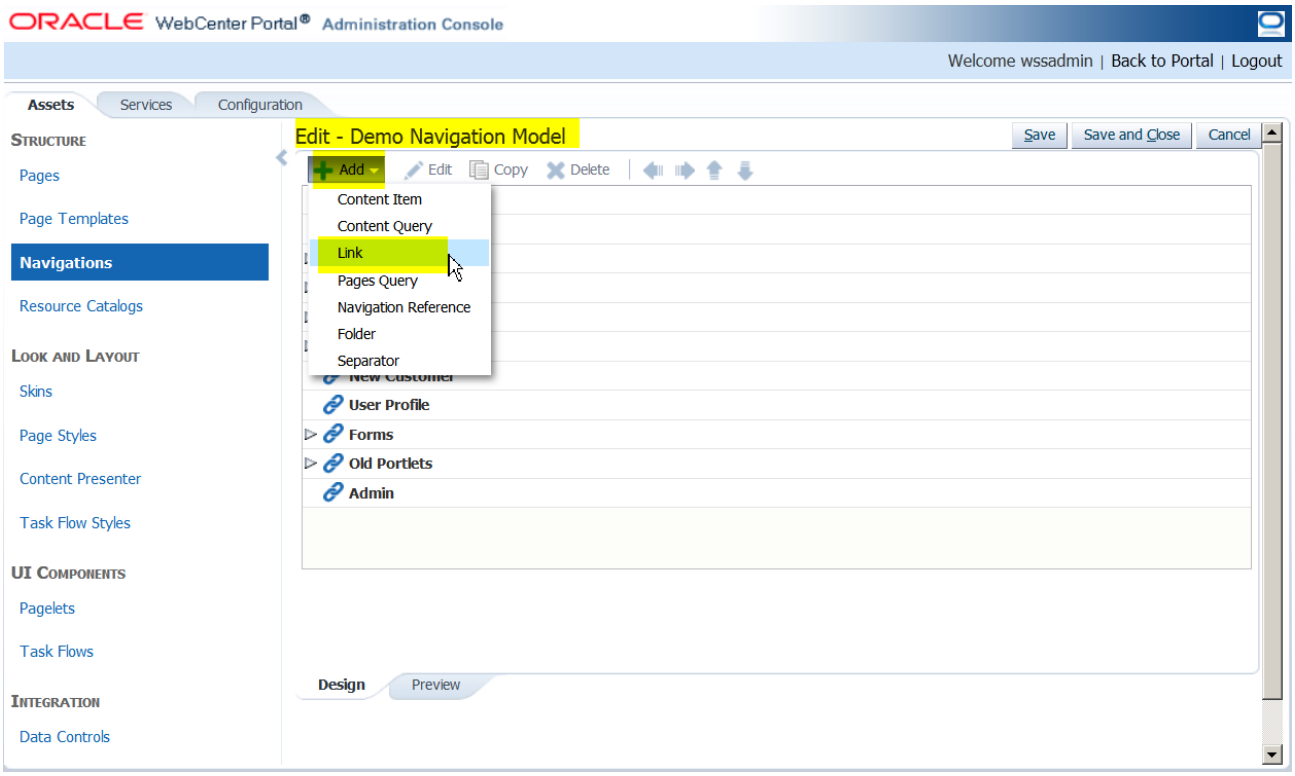
4 In the **Create** dialog, enter a name and description for the new navigation model, and choose the Navigation Model to be customized in the **Copy from** list box. Click **Create** to create the model.



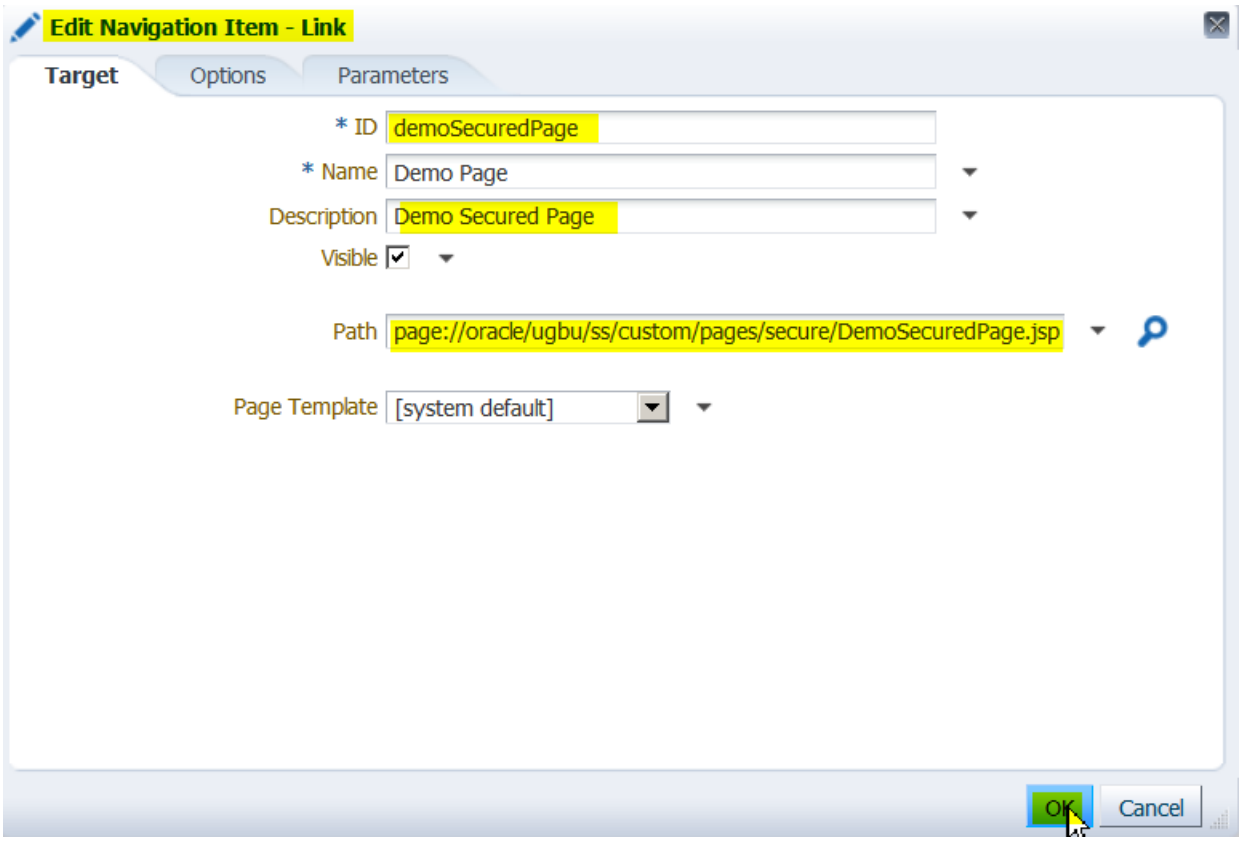
5 To begin the customization, click **Edit**.



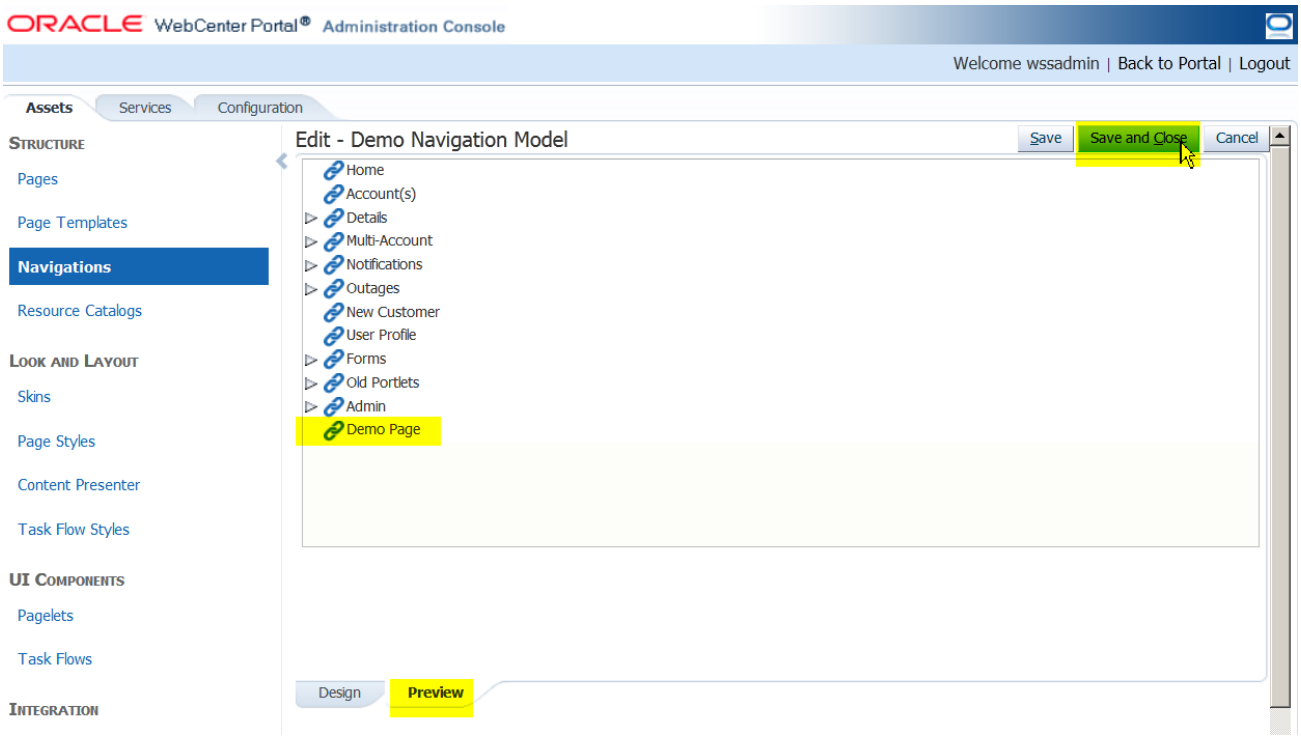
6 In the **Edit** view, customize the navigation model per your implementation requirements.



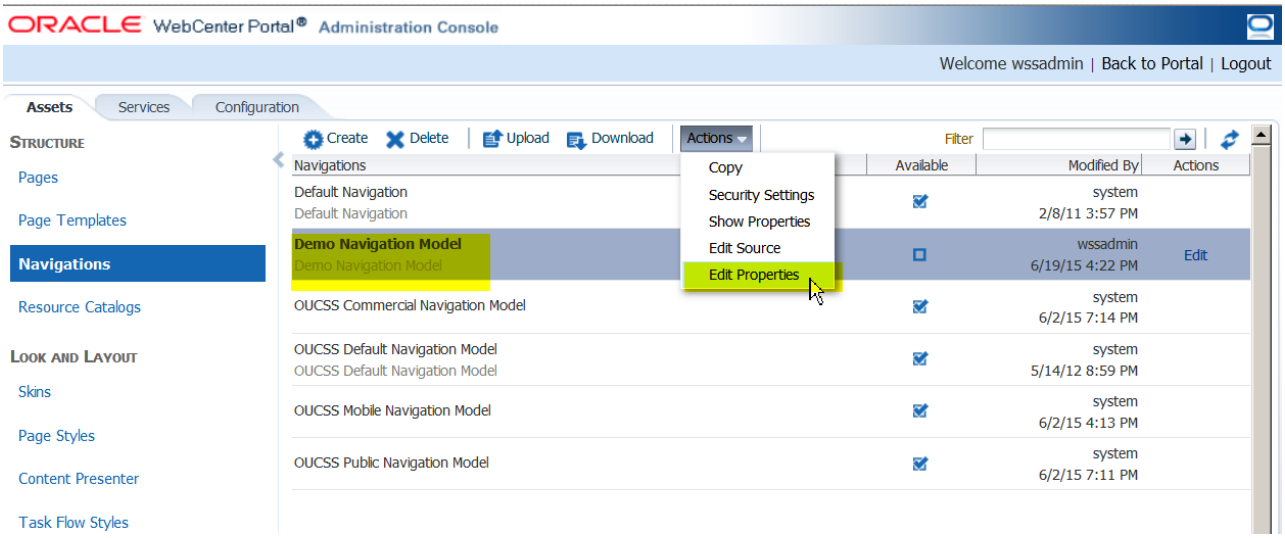
In this example, the `DemoSecuredPage.jspx` from the ExtendOUCSSPortal is added to the Demo Navigation Model.



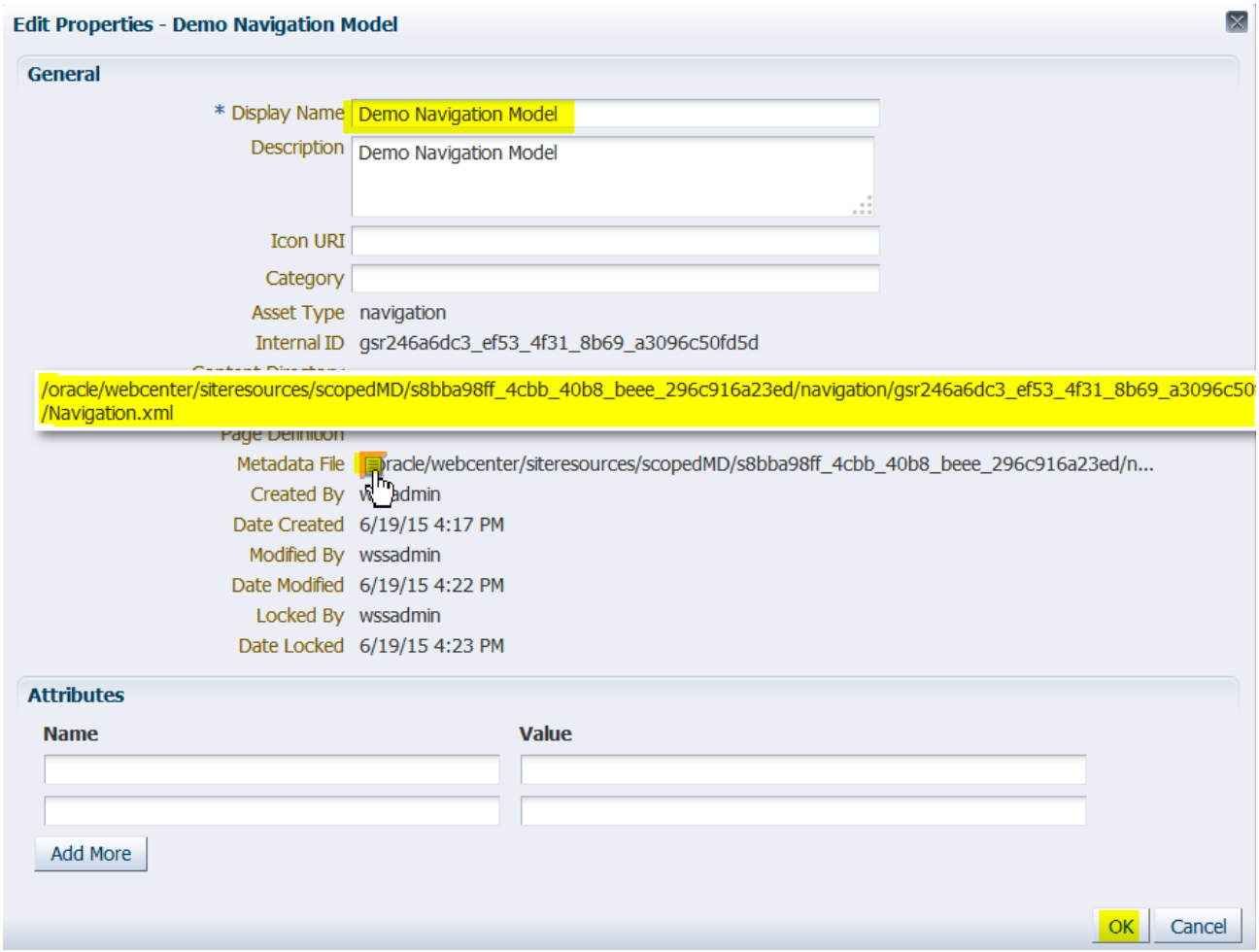
7 Choose the **Preview** tab at the bottom of the **Edit** list to verify the customized model. Click **Save and Close** when done.



8 To activate and use the new Navigation Model for one or more contexts, choose **Actions** > **Edit Properties** to access the XML path of the navigation model..



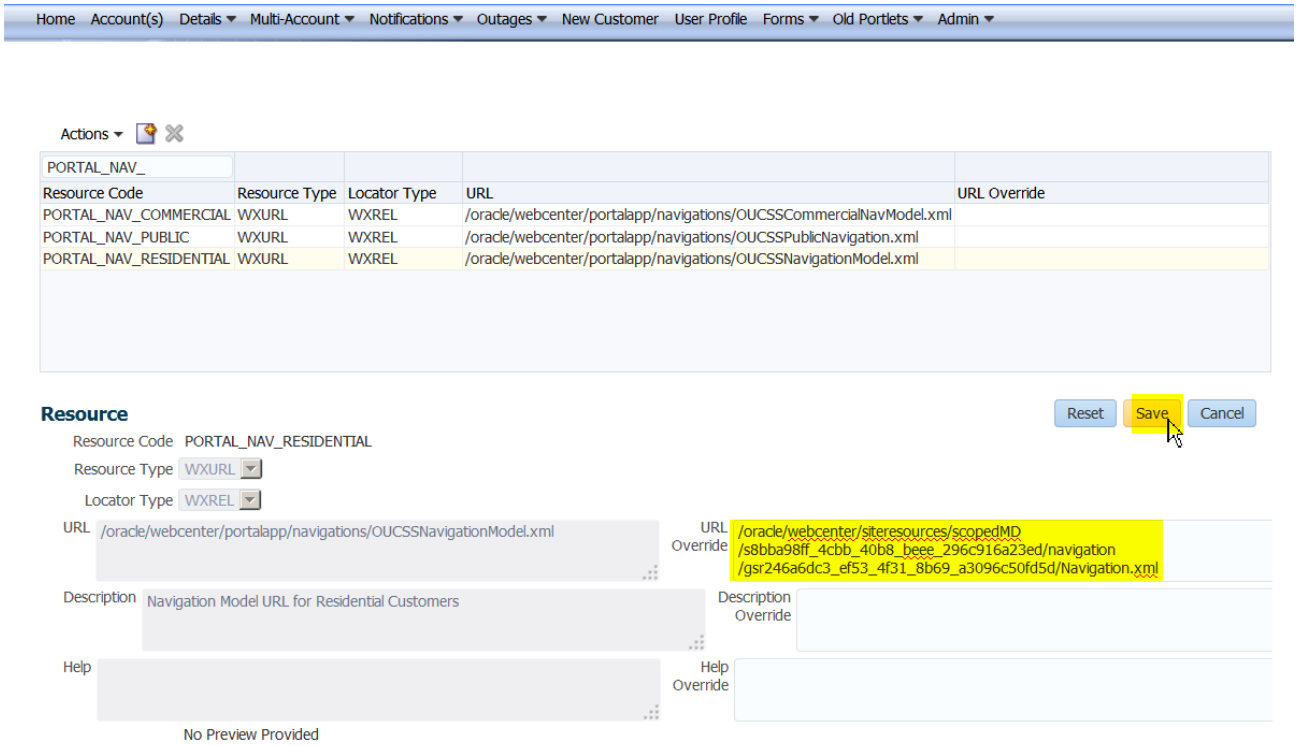
9 Copy the complete **Metadata** path of the **Navigation Model** from the edit dialog.



10 Click the **Back to Portal** link to exit the Administration Console and return to the portal application, then select the **Admin** > **Resources** page.

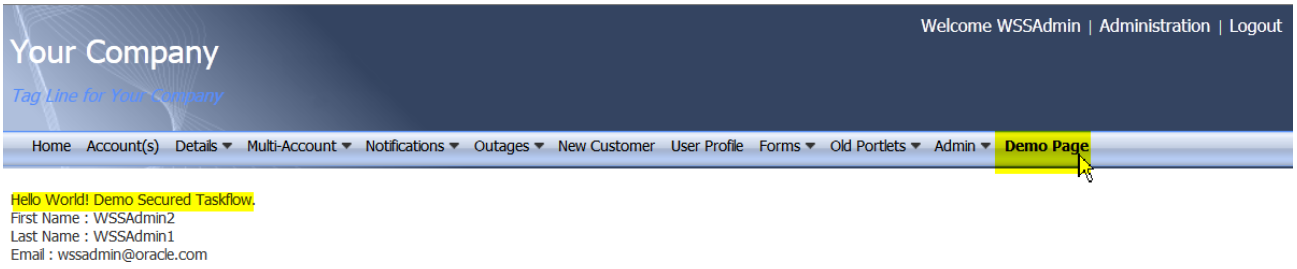
11 Filter the resources with resource codes **PORTAL_NAV**. Update the **URL Override** property by selecting a **PORTAL_NAV_<<Context>>** (**RESIDENTIAL, PUBLIC**, etc.) resource with the **Metadata** path copied from step 9.

Note: To update the **Mobile Navigation Model**, search for the appropriate resource code, e.g., `MOB_B_PORTAL_NAV`.



12 Repeat the process to update the Navigation Model for other LOB contexts.

13 Restart the OUCSS Portal managed server and verify that your changes are reflected in the portal.

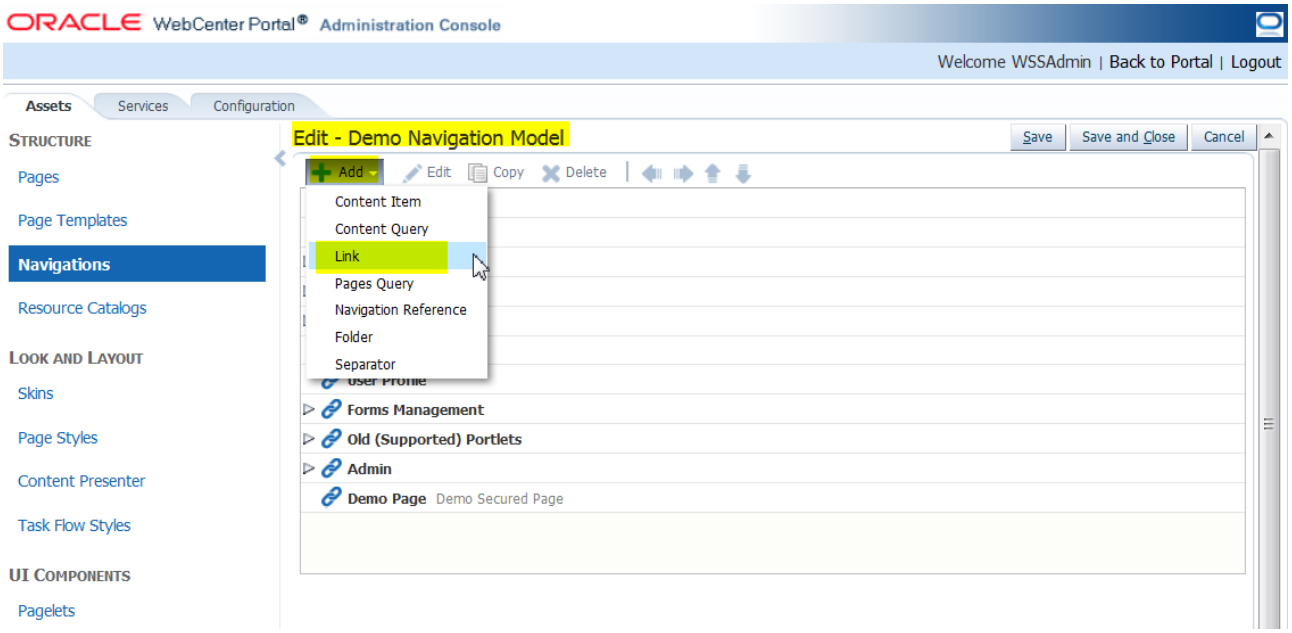


Accessing Pages from the Extended OUCSS Portal

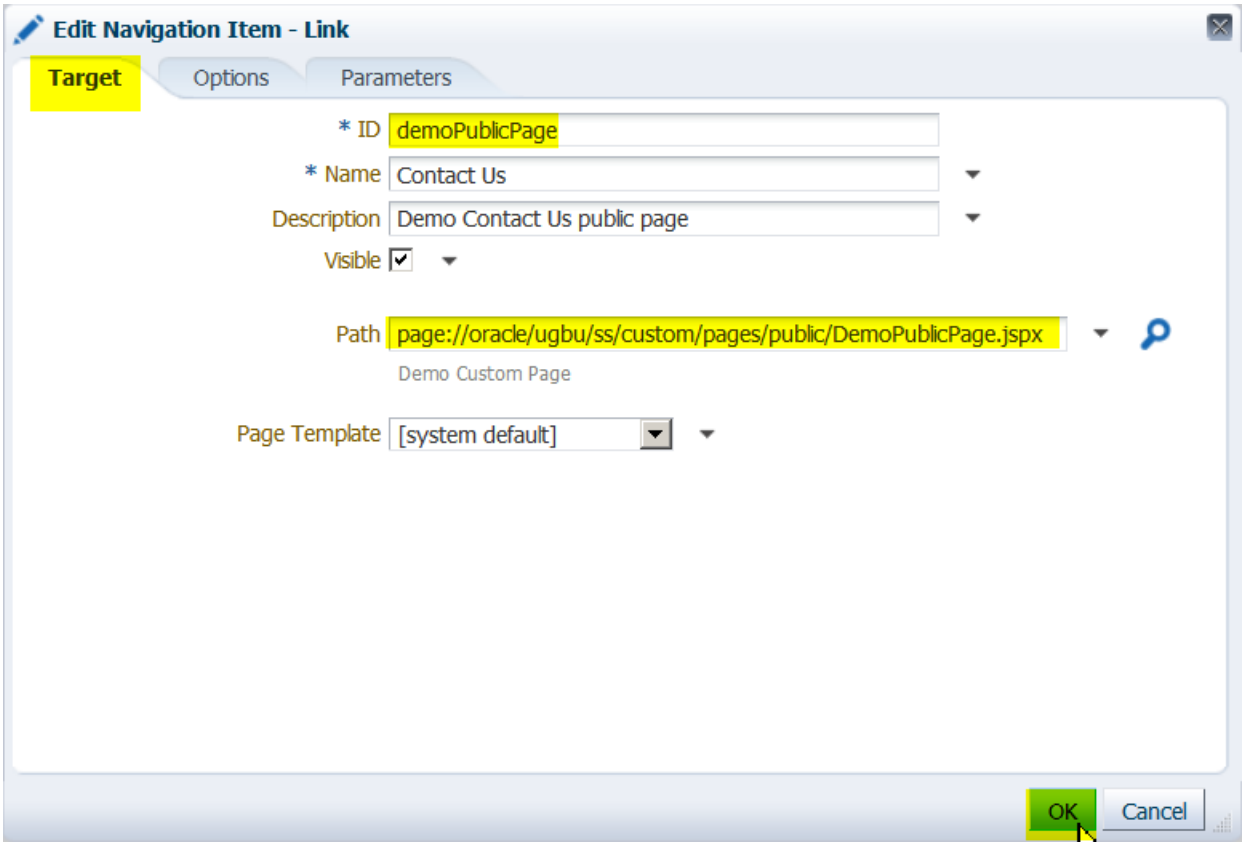
Newly-added pages can be accessed by customizing the navigation model and adding links to the customized pages.

1 Log in to the OUCSS Portal and Edit the **Demo Navigation Model**.

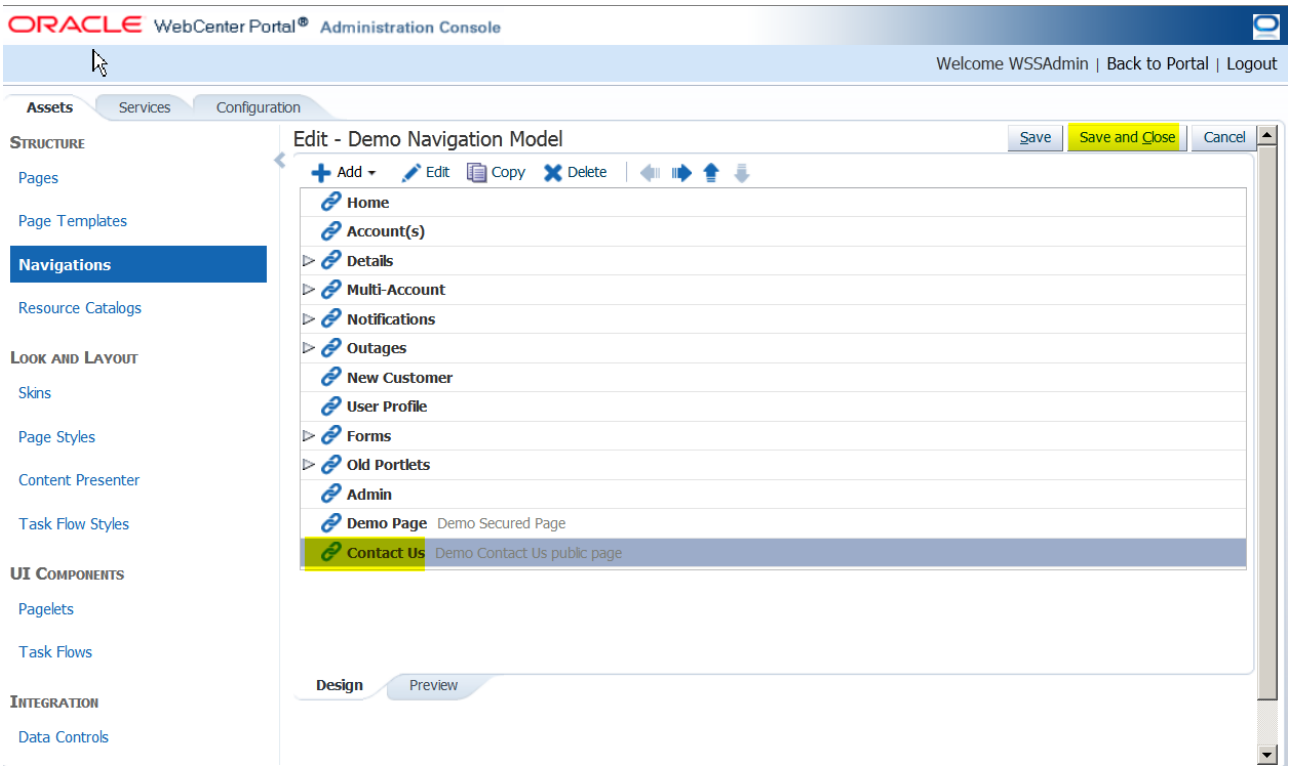
2 In the **Edit** mode, click **Add** > **Link**.



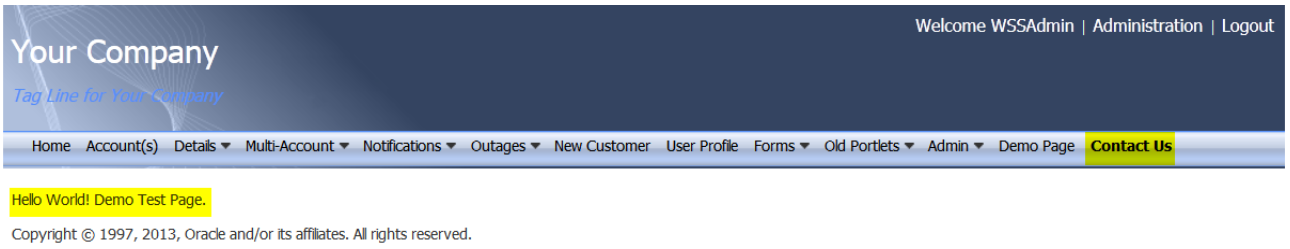
3 In the **Edit Navigation Item** dialog, choose the **Target** tab. Enter the link details, then click **OK**.



4 Add links to other pages per your site requirements. Click **OK** in the **Edit** dialog when you have finished adding links.



5 Click the **Back to Portal** link and verify that the navigation model is updated.



Exporting an Existing Navigation Model

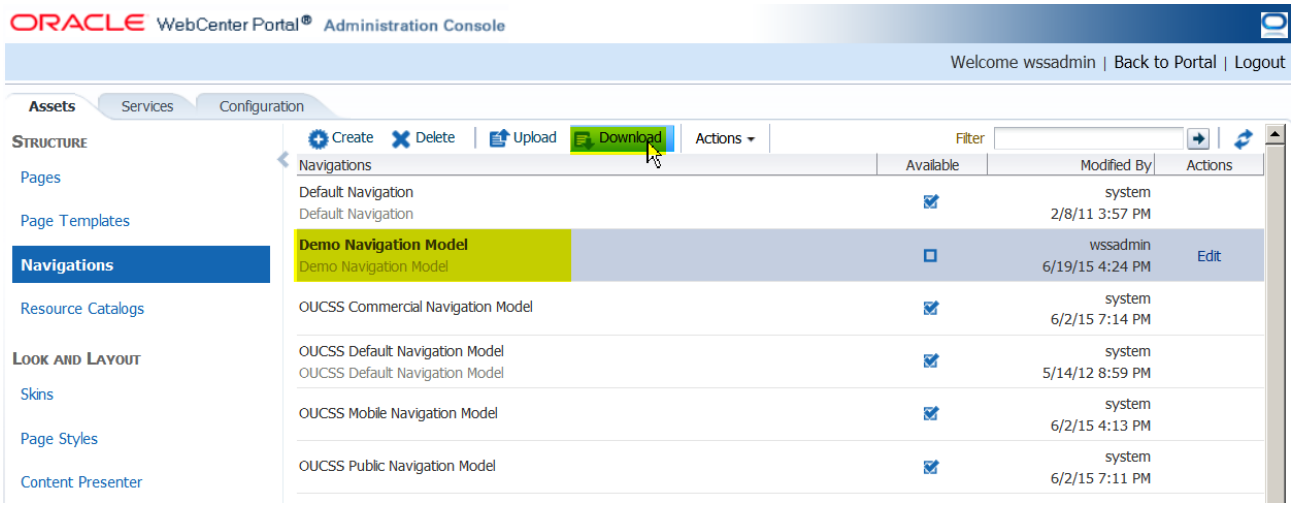
Navigation models from other Portal instances can be exported for migration and backup.

To export a navigation model:

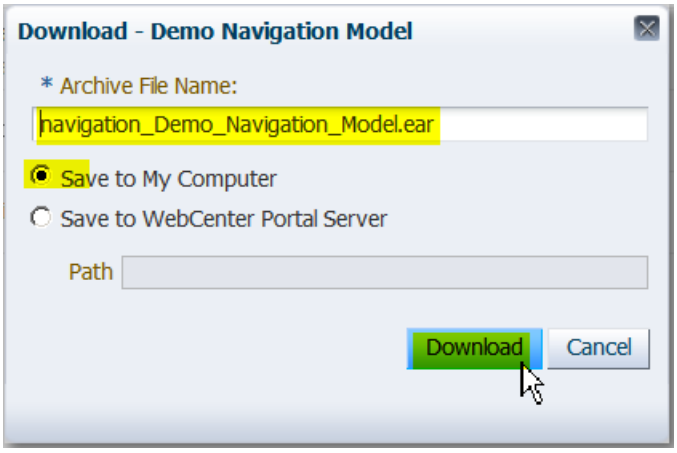
1 Log in to the OUCSS Portal with Administrator privileges and go to the Administration Console.

2 Choose the **Assets** tab and click on the **Navigations** link.

3 Select the navigation model to export/download and click **Download**.



4 In the **Download** dialog, enter the archive name and choose the save location, then click the **Download** button. You can also choose to save the file on the Portal Server by providing the path to the server.



5 Save the file to the target location.

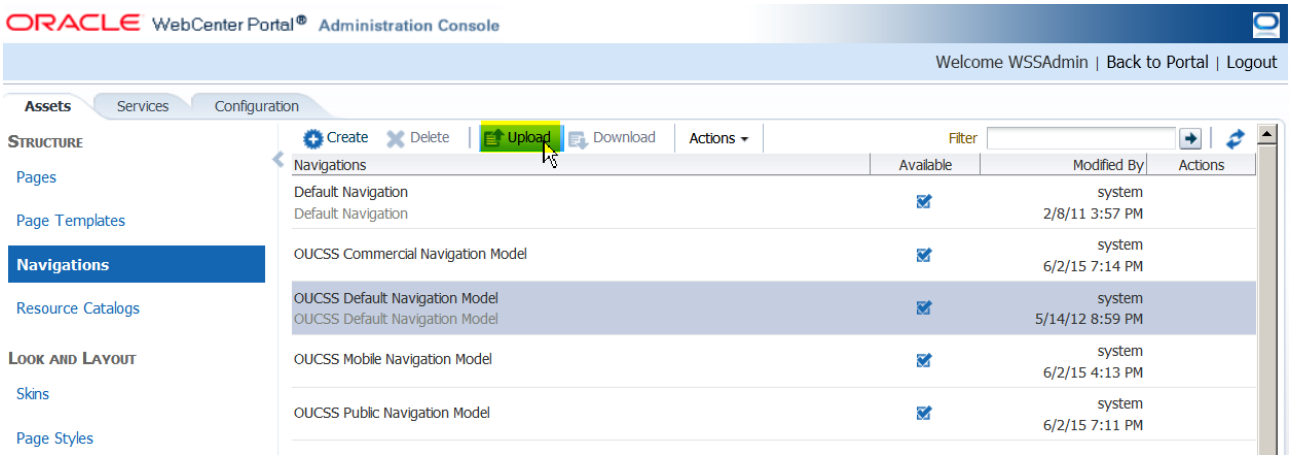
Importing a Navigation Model

To import navigation models from other instances or from backup archives:

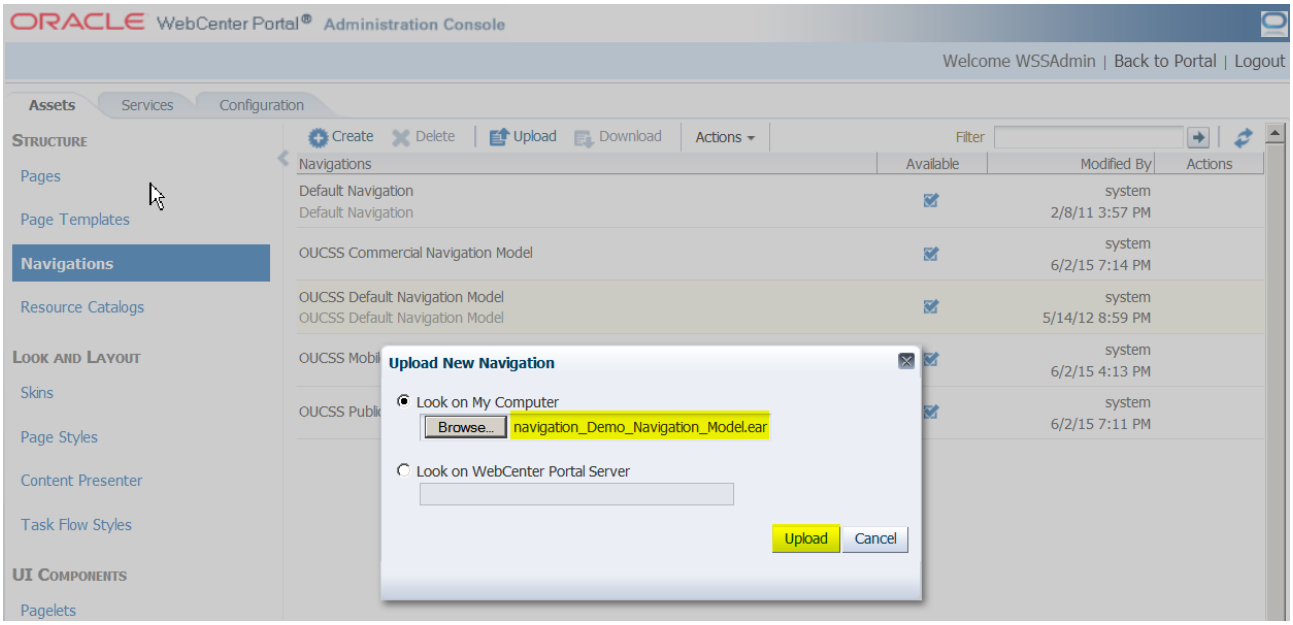
1 Log in to the OUCSS Portal with Administrator privileges and go to Administration Console.

2 Choose the **Assets** tab and click on the **Navigations** link.

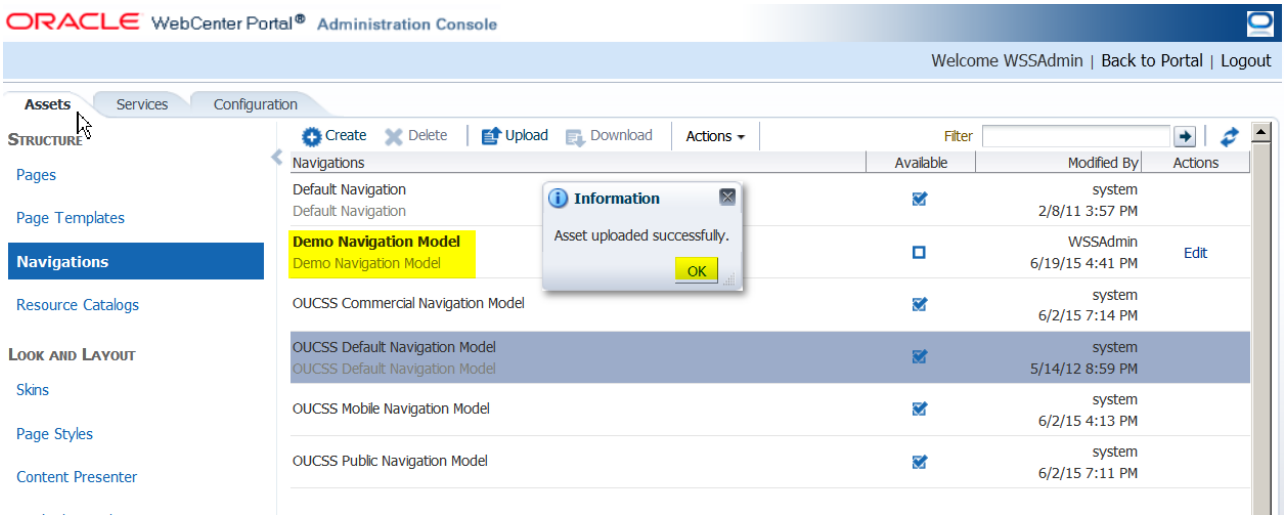
3 Click **Upload**.



4 In the **Upload New Navigation** dialog, browse the exported navigation model archive to locate the model for import, then click **Upload**.



5 Click **OK** in the confirmation dialog, and verify that the imported model appears in the **Navigations** list.



6 To customize and configure the imported navigation model, see the topic, Customizing Provided/Creating New Navigation Models.

Customizing the Resource Catalog

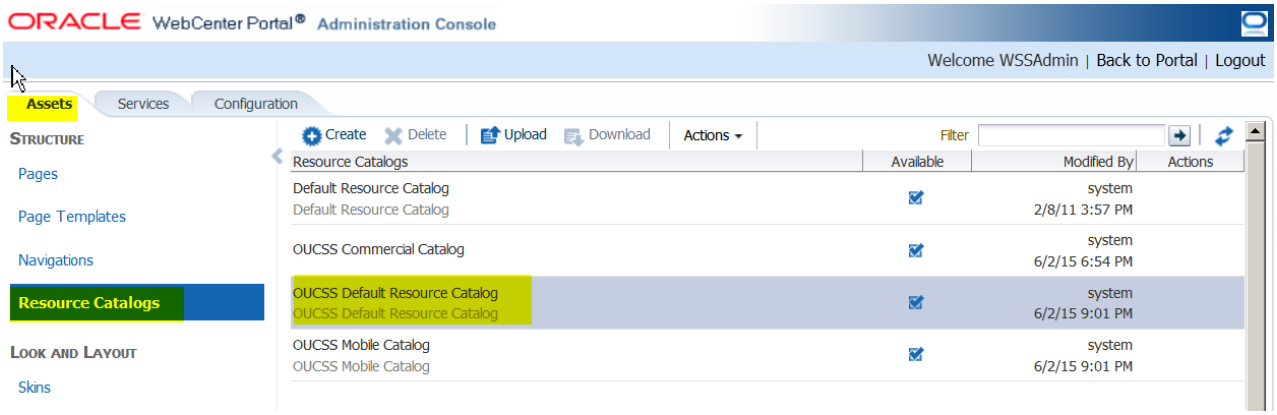
An OUCSS Portal application is configured to use separate catalogs for Residential and Commercial contexts. The following example describes customizing the OUCSS Default Resource Catalog. The same steps can be followed to customize the OUCSS Commercial Catalog.

For additional information on customizing the resource catalog, see http://docs.oracle.com/cd/E29542_01/webcenter.1111/e63259/jpsdg_catalog.htm

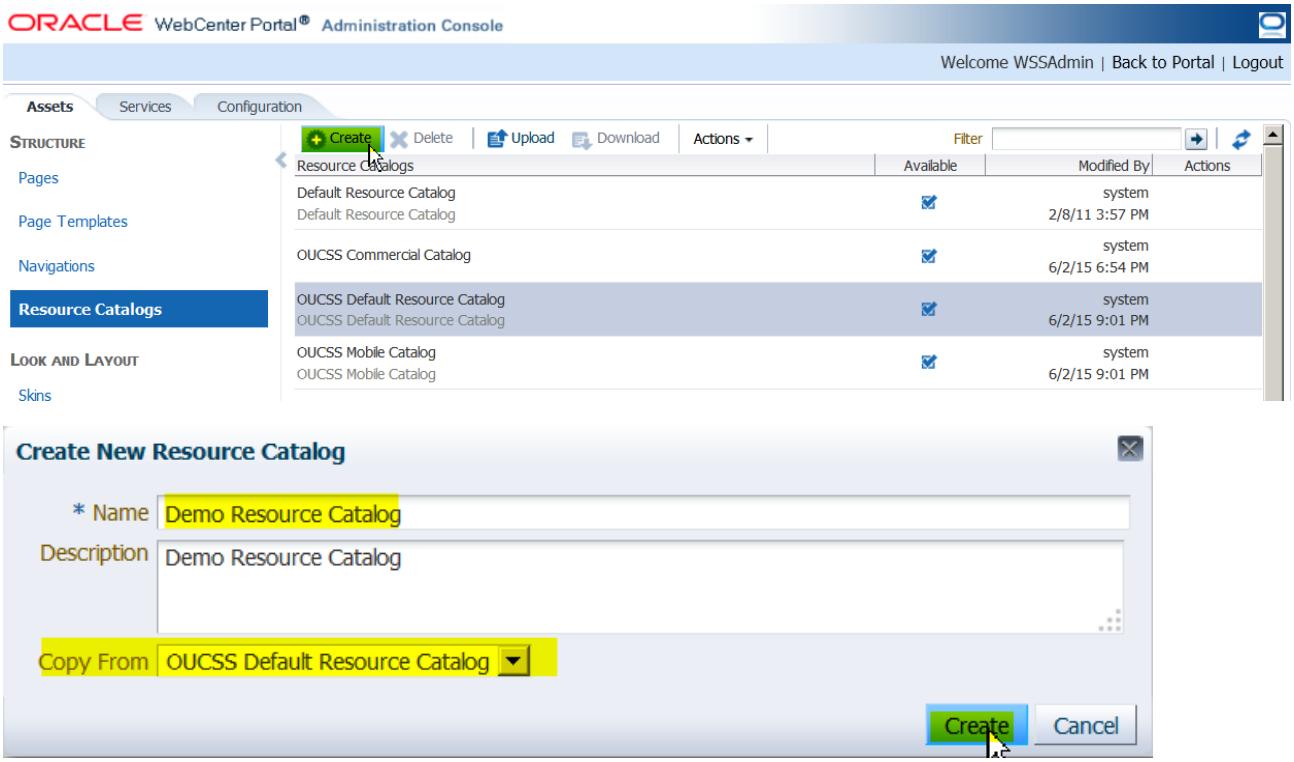
To consume the taskflow from the `extend.oucss.portal` library, the provided (out-of-the-box) resource catalog must be customized as follows:

1 Log in to the OUCSS Portal with Administrator privileges and go to the Administration Console.

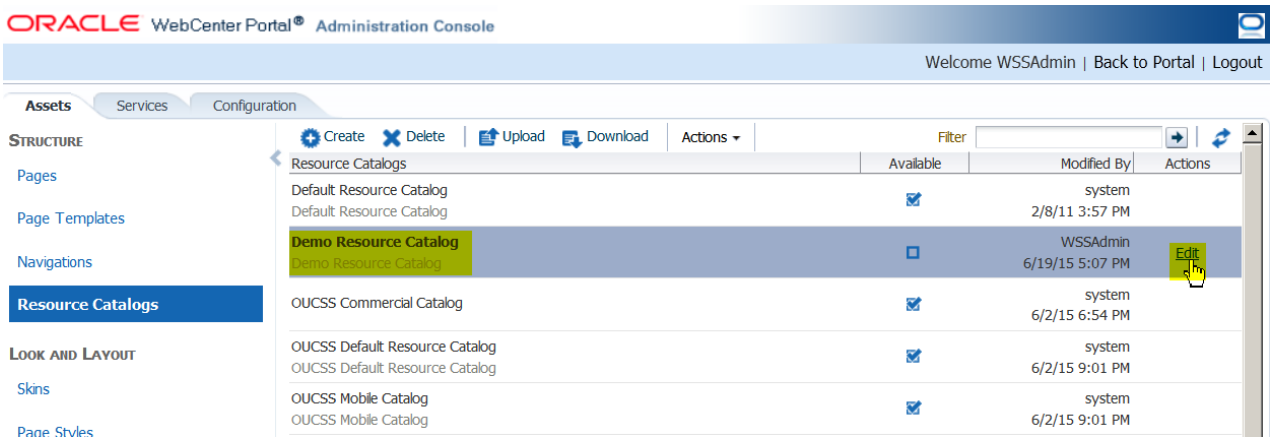
2 Choose the **Assets** tab and click on the **Resource Catalogs** link in the **Structure** section.



3 Create a new catalog and choose **OUCSS Default Resource Catalog** from the **Copy from** list box.

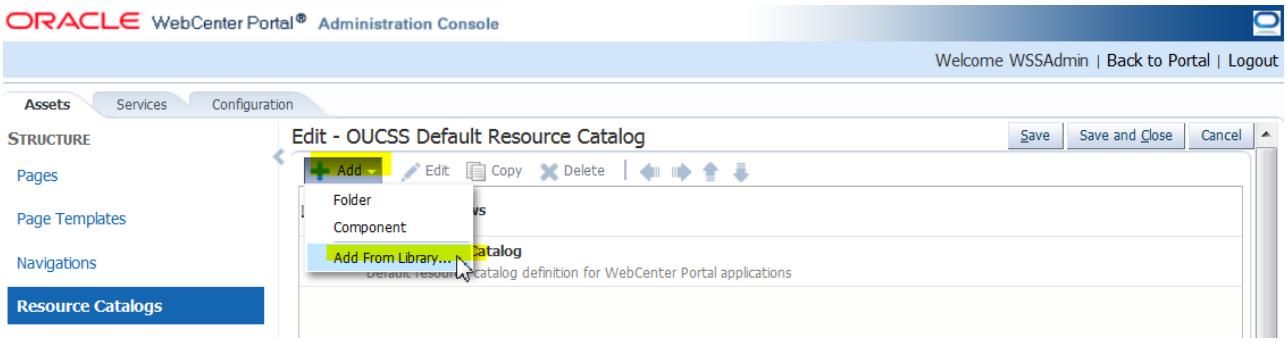


4 Click **Edit** to edit the catalog.



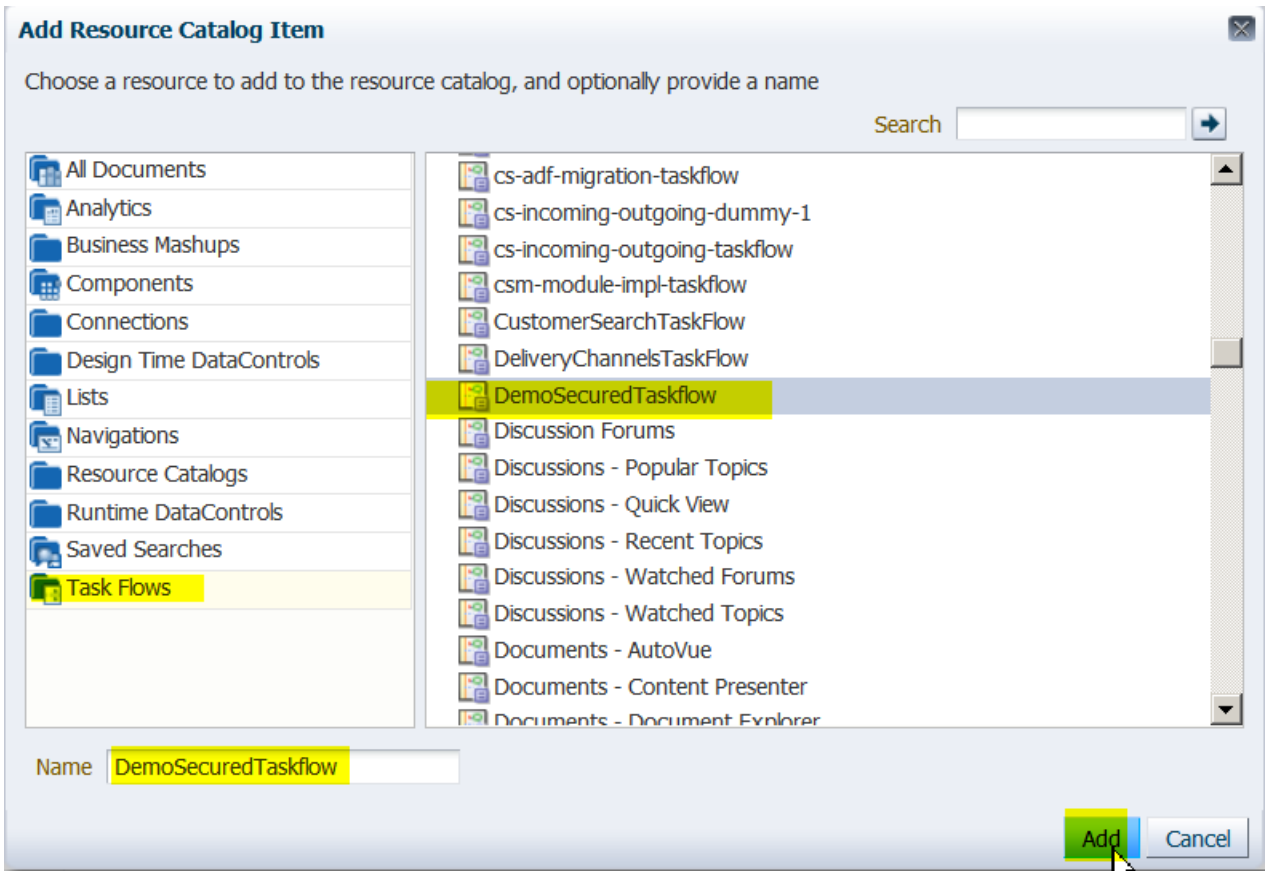
5 In the **Edit** dialog, click **Add > Add From Library…** to add the new taskflow to the catalog.

Note: You can also add folders or add the taskflow to another location in the catalog. For the purposes of this demo, the new taskflow (TestTaskflow) will be added at the catalog's root level.

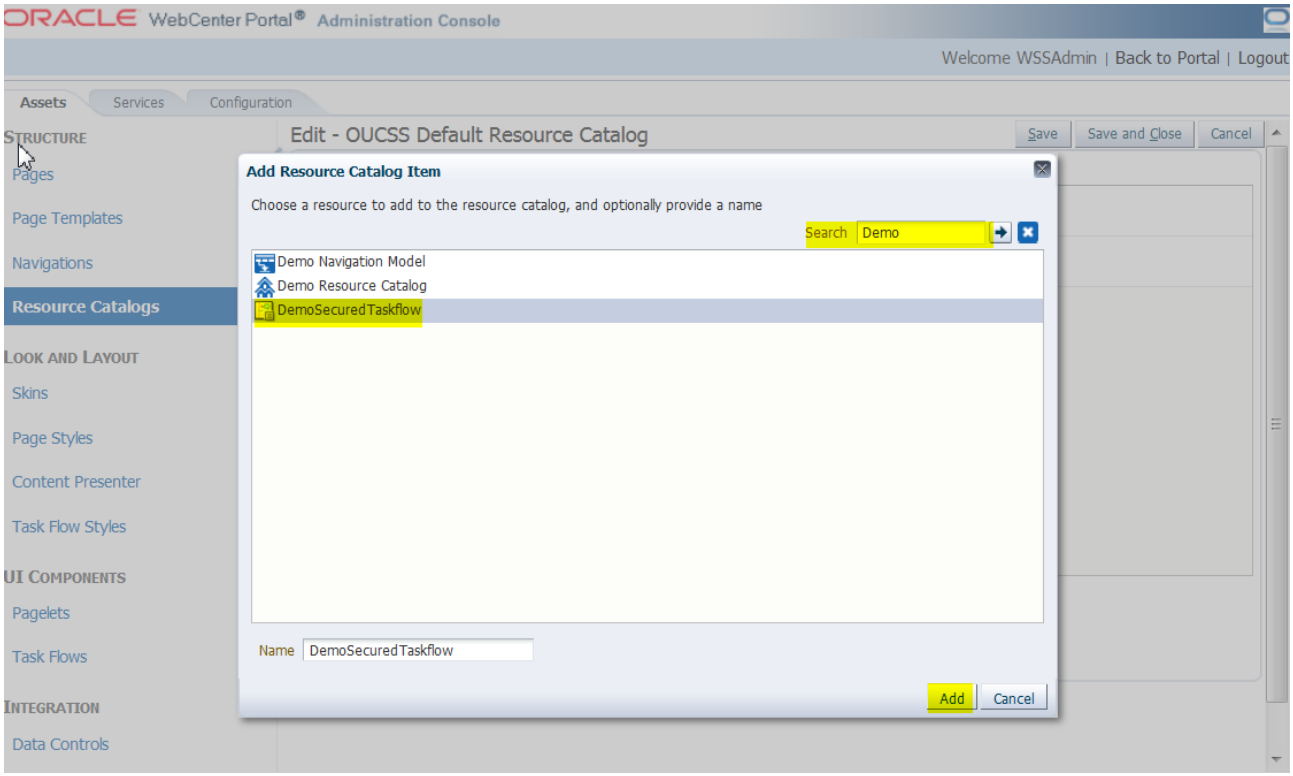


6 In the **Add Resource Catalog Item** dialog, select **Taskflows** from the menu on the left, then scroll down and choose the new taskflow (in this demo, choose the **DemoSecuredTaskflow** that was created earlier). Note that you can also search for the taskflow using the **Search** field in the dialog.

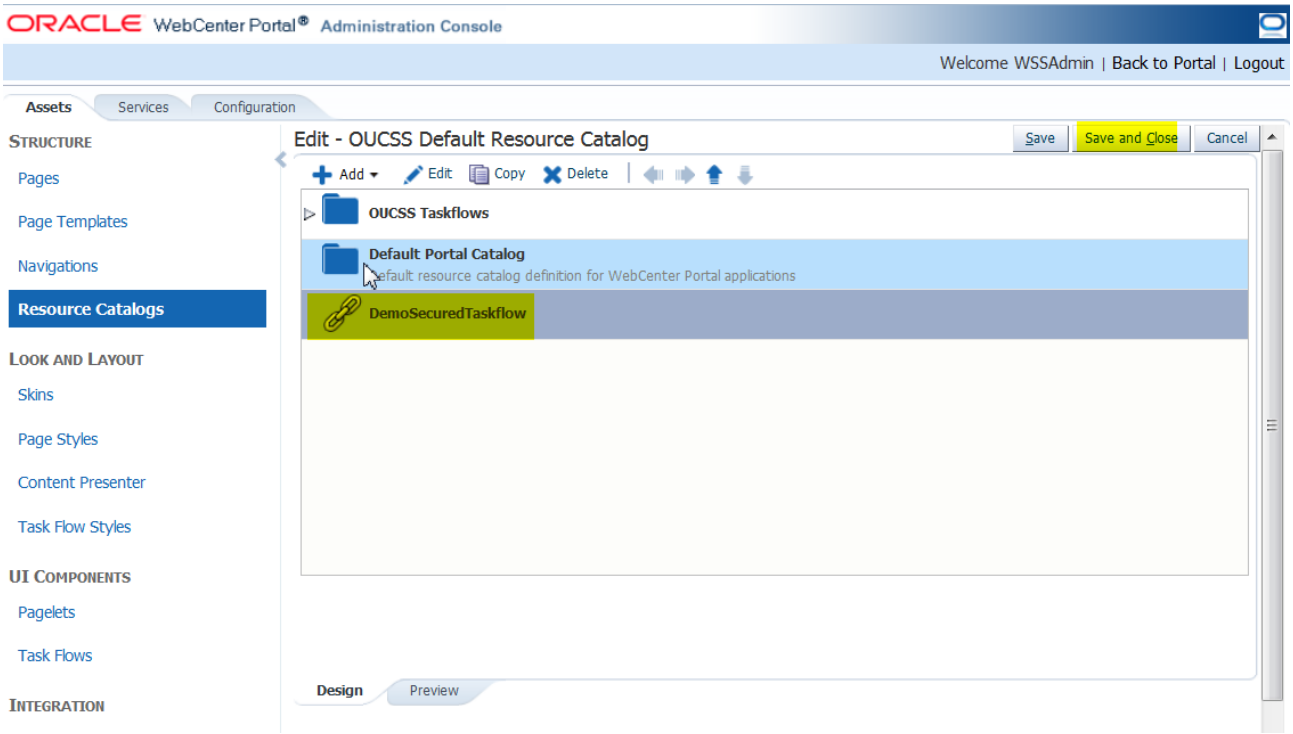
When the taskflow is selected, click the **Add** button.



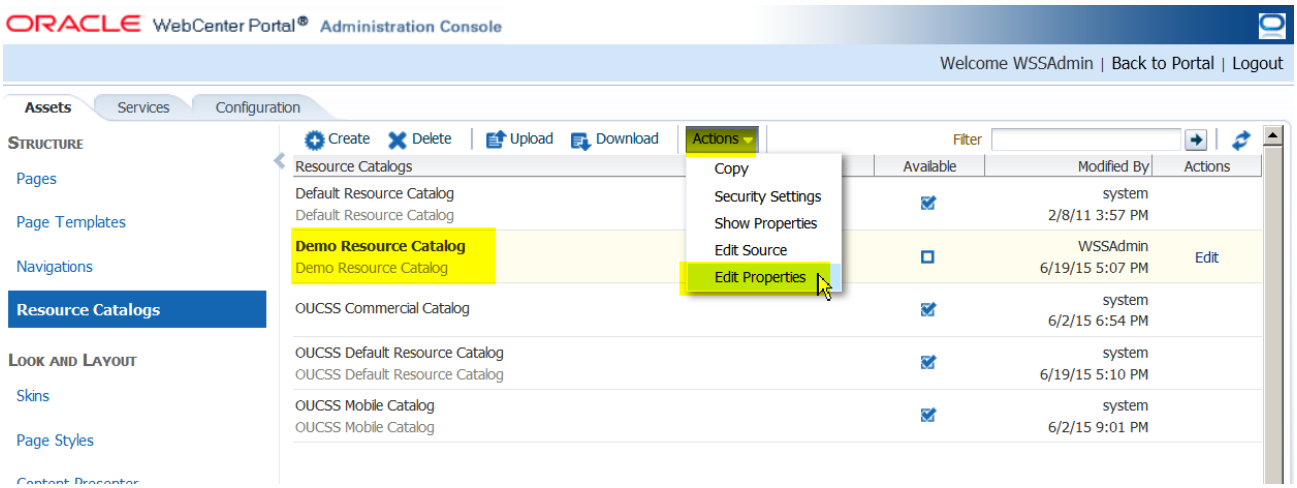
Alternatively, you can search a taskflow and add it to the catalog.



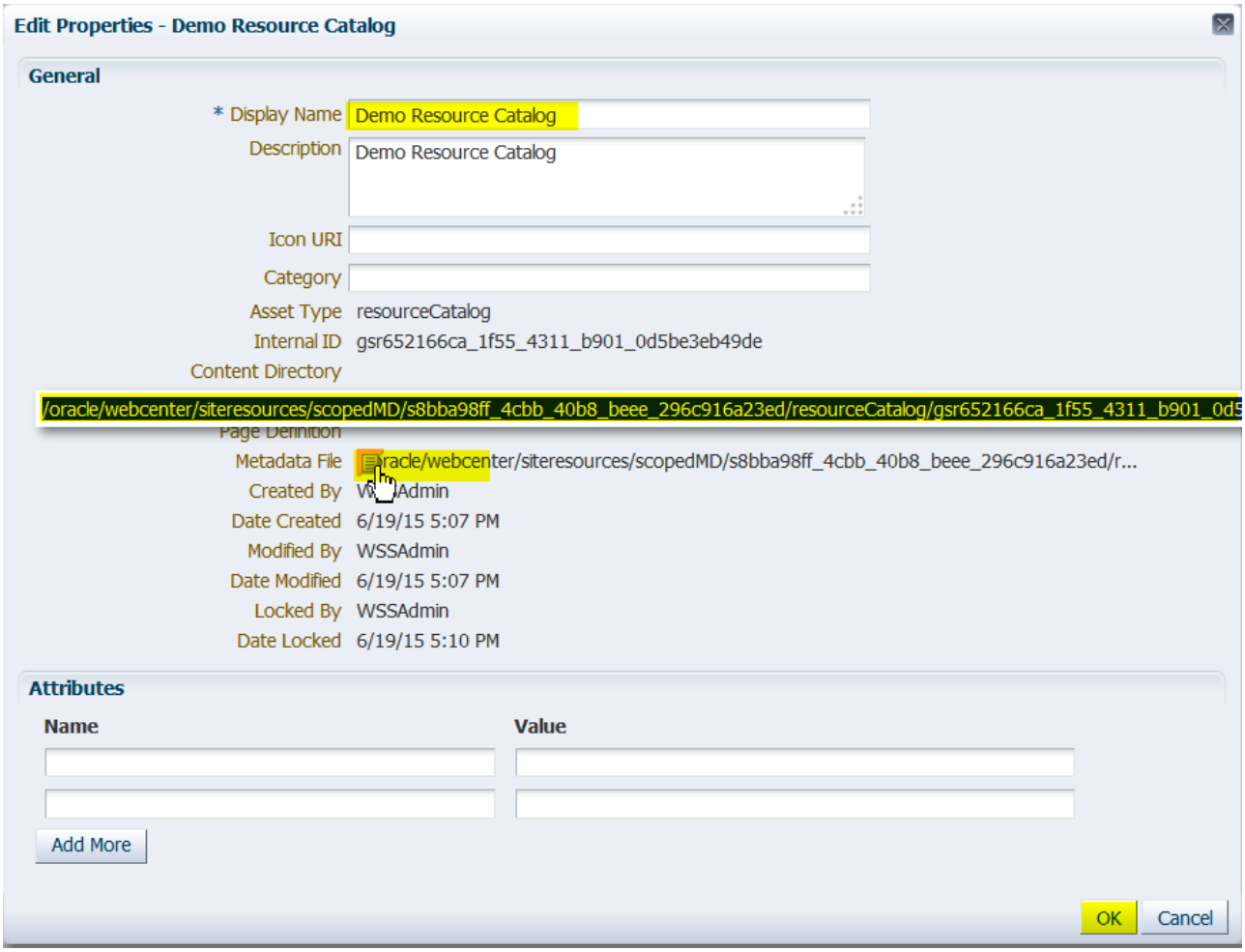
7 The new taskflow now appears in the catalog list. Click **OK** to save the changes.



8 To activate and use the new **Custom Portal Catalog** for one or more contexts, first copy the page path of the new catalog by choosing **Actions** > **Edit Properties**.

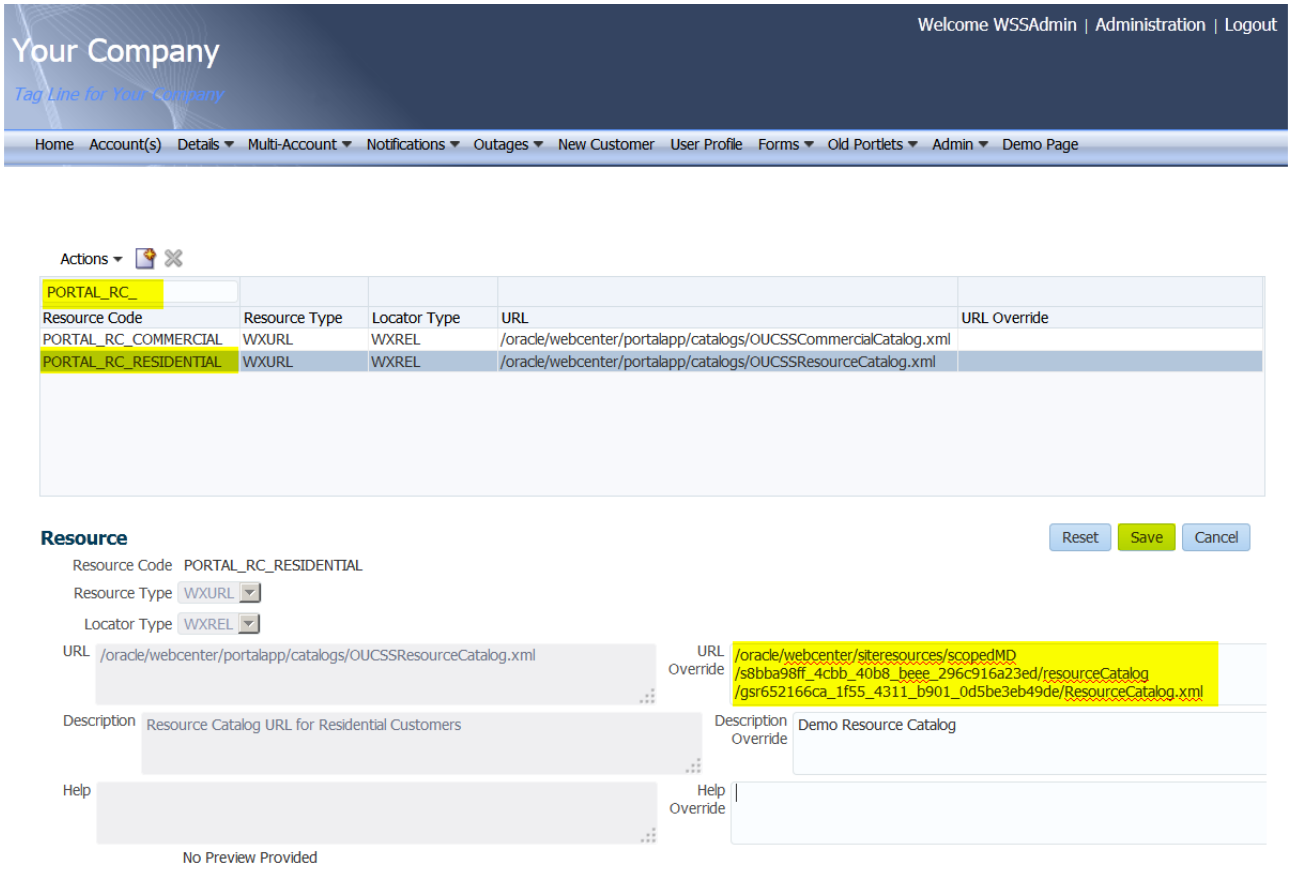


9 Select and copy the **Metadata File** path from the **Edit Properties** dialog.



10 Click the **Back to Portal** link, then select the **Admin > Resources** page.

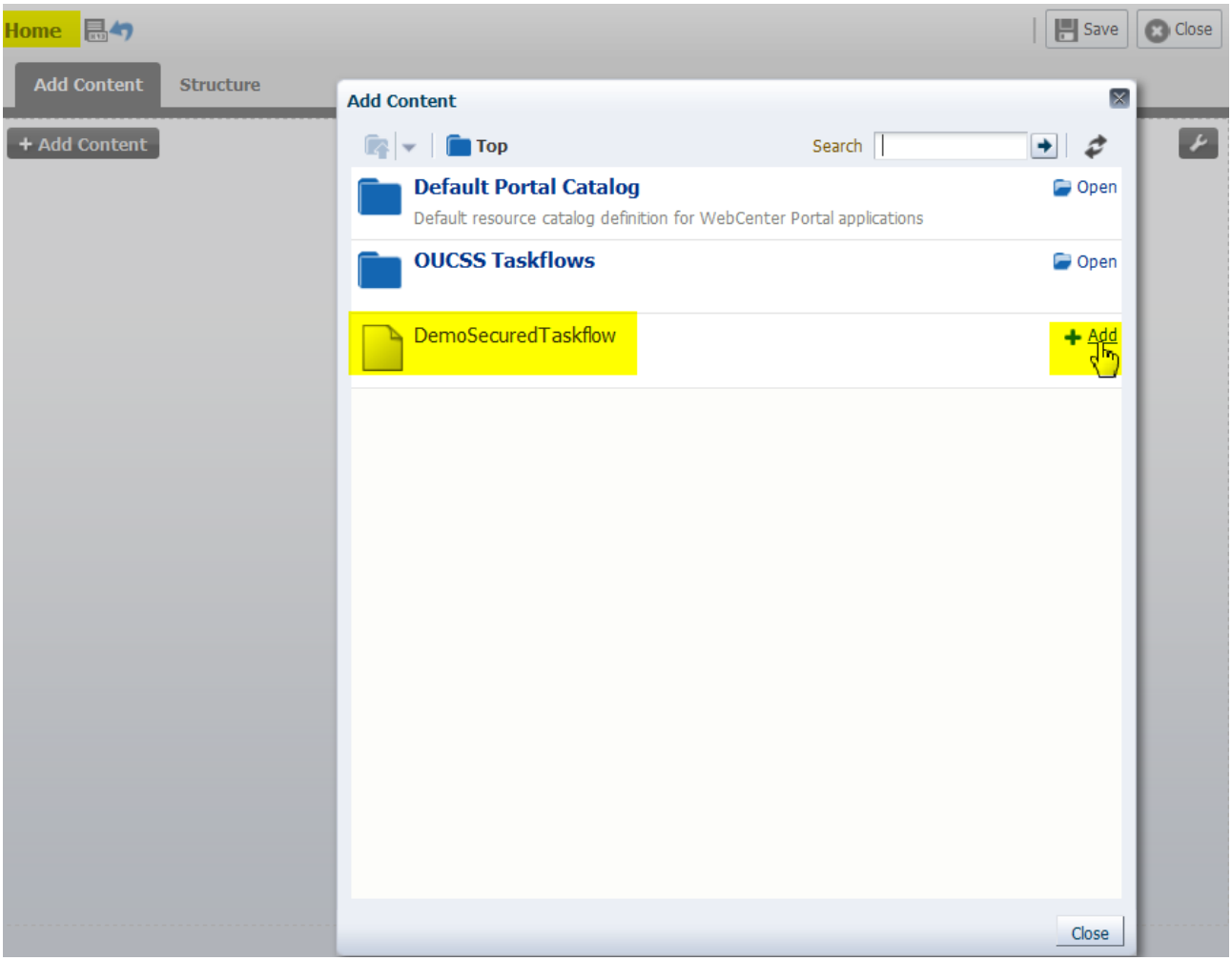
11 Filter the resources with resource codes **PORTAL_RC_**. Update the **URL Override** property by selecting the **PORTAL_RC_<<Context>>** resource with the **Metadata File** path.



12 Save your changes and repeat the same process to update other contexts with the custom **Site Template**.

13 Restart the OUCSS Portal managed server and verify that your changes are reflected in the portal.

14 Edit a page and add the **Demo** taskflow from the **Resource Catalog**. Click **Save**, then **Close**.



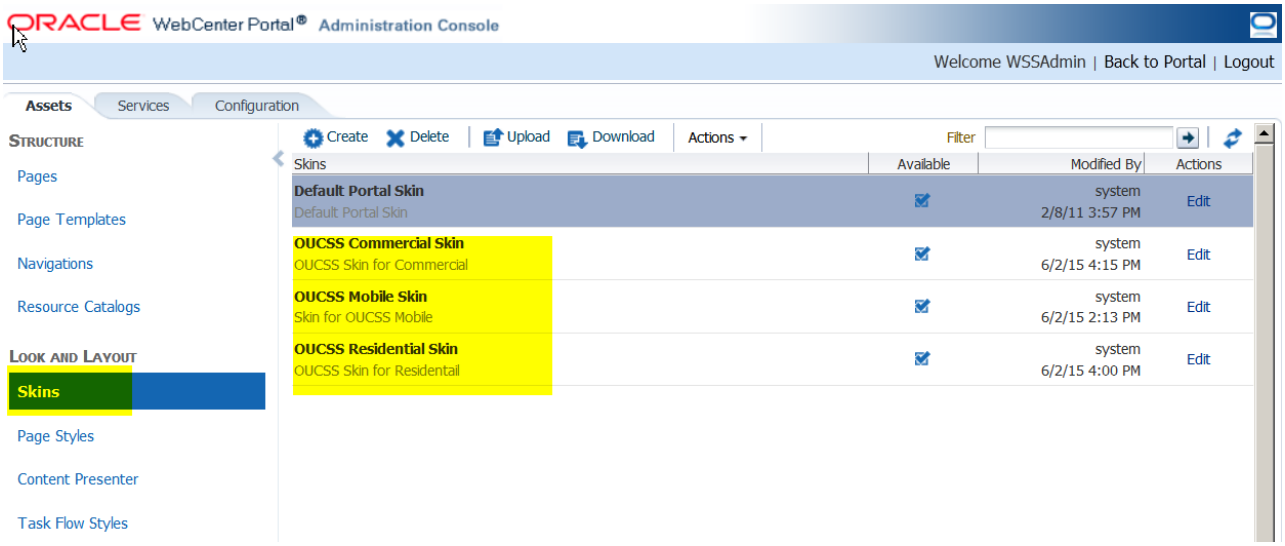
Customizing Skins

An OUCSS Portal application is configured to use separate skins for Residential and Commercial contexts as well as for Mobile. The following example describes customizing the OUCSS Residential Skin. The same steps can be followed to customize other skins.

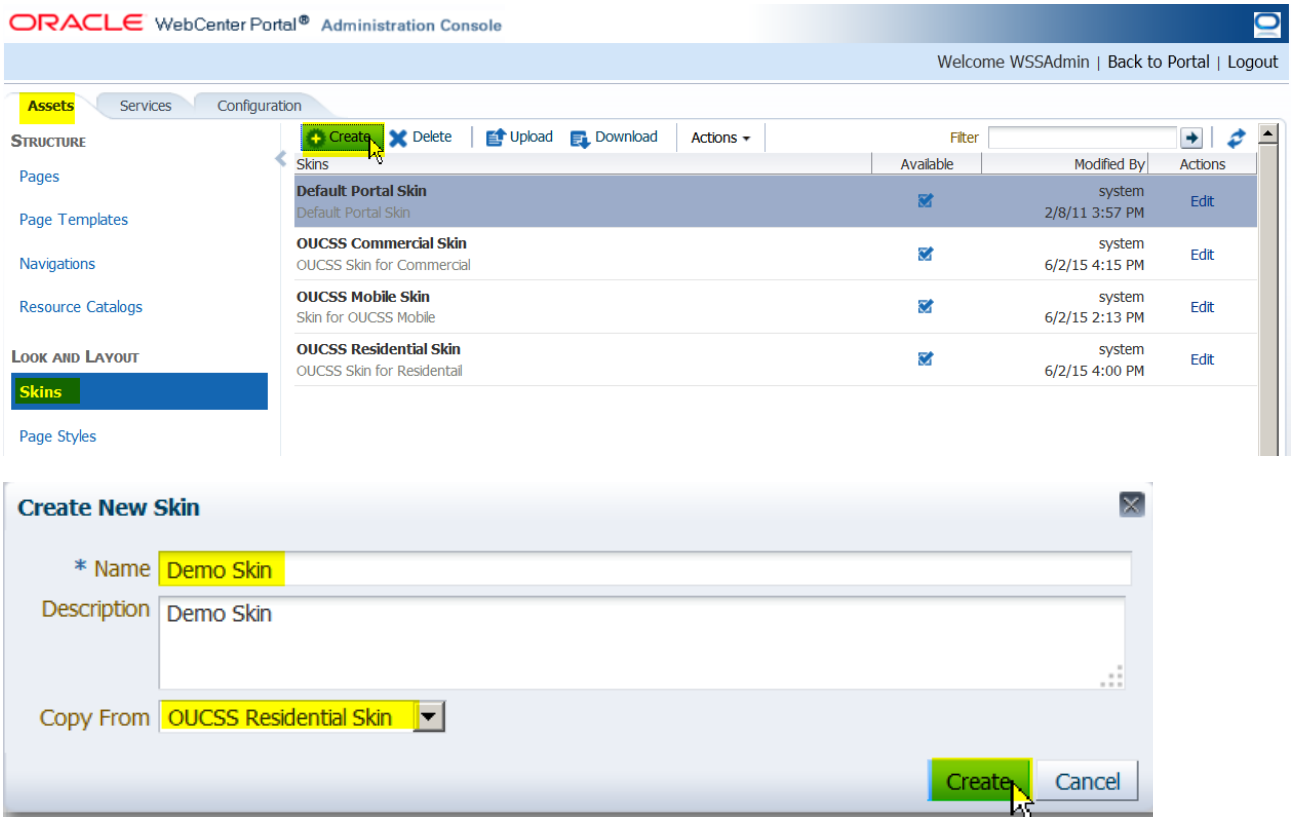
For additional information on customizing the resource catalog, see
http://docs.oracle.com/cd/E29542_01/webcenter.1111/e63259/jpsdg_skin.htm

OUCSS Residential Skin customization example:

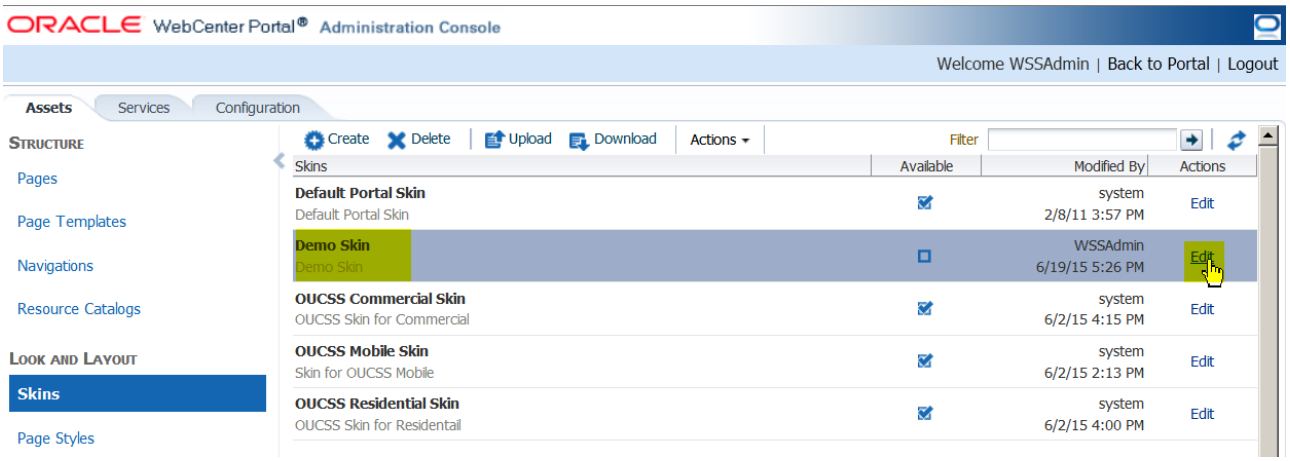
1. Log in to the OUCSS Portal with Administrator privileges and go to the Administration Console.
2. Choose the **Assets** tab and click on the **Skins** link in the **Structure** section.



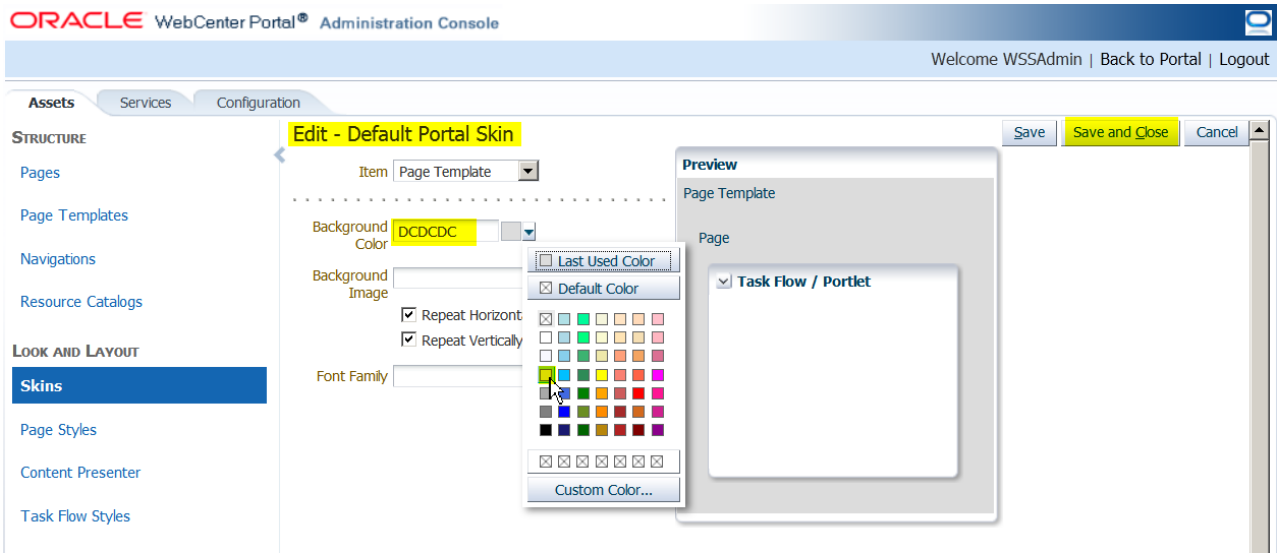
3 Click **Create** to create a new Skin and choose **OUCSS Residential Skin** from the **Copy from** list box.



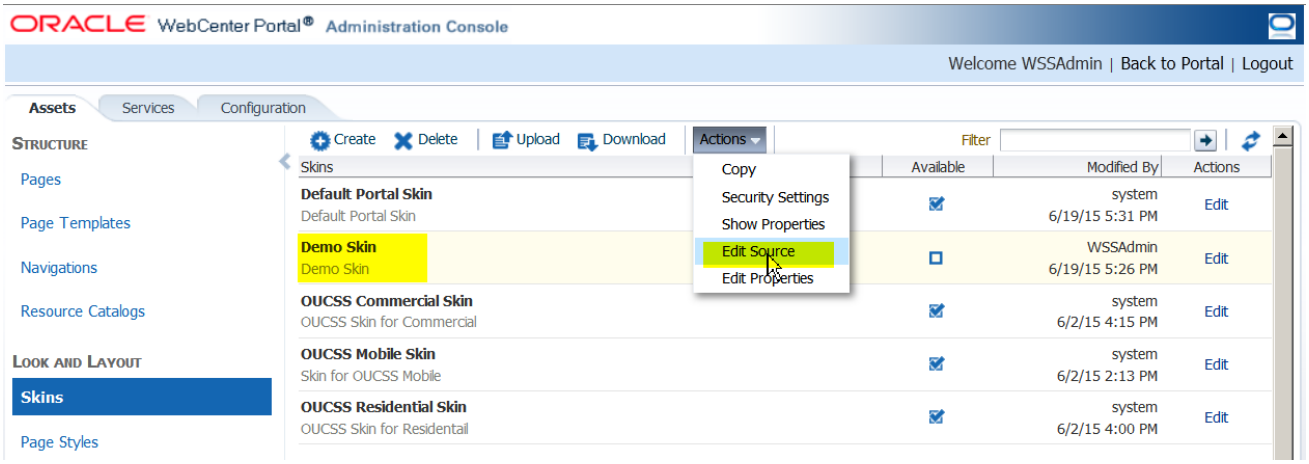
4 Click **Edit** to edit the skin.



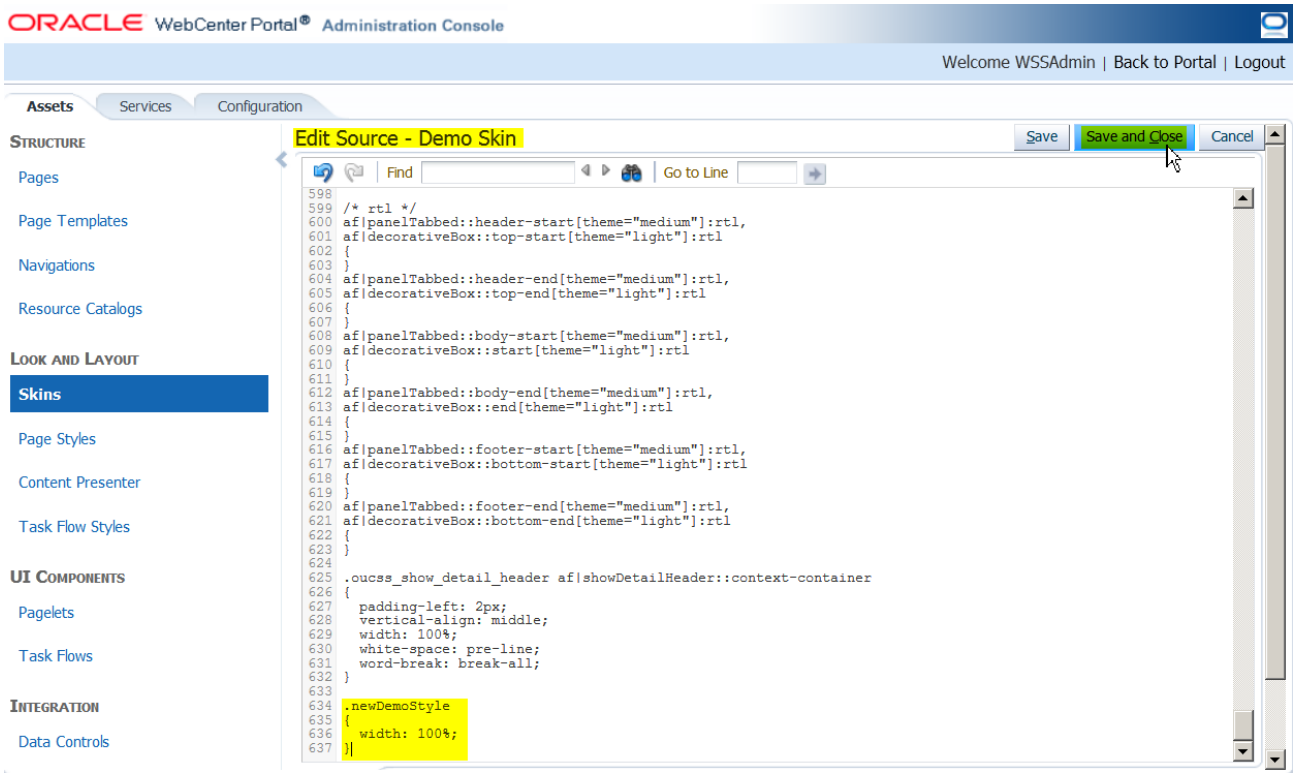
5 In **Edit** mode, customize the skin. When editing is complete, click **Save and Close**.



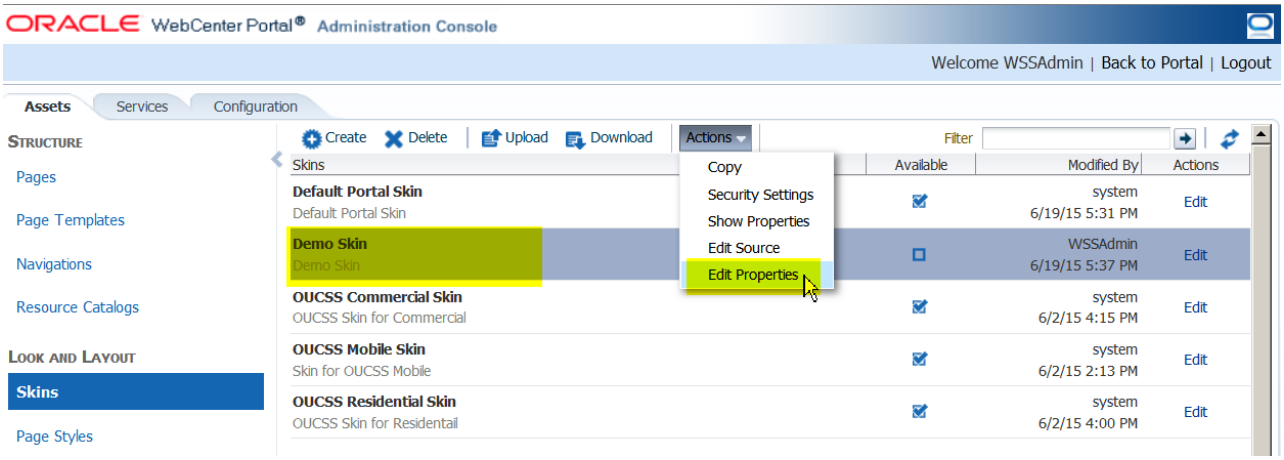
6 Skin styles can also be modified by editing the source. To edit the source, Select the **Skin** and click **Actions** > **Edit Source**.



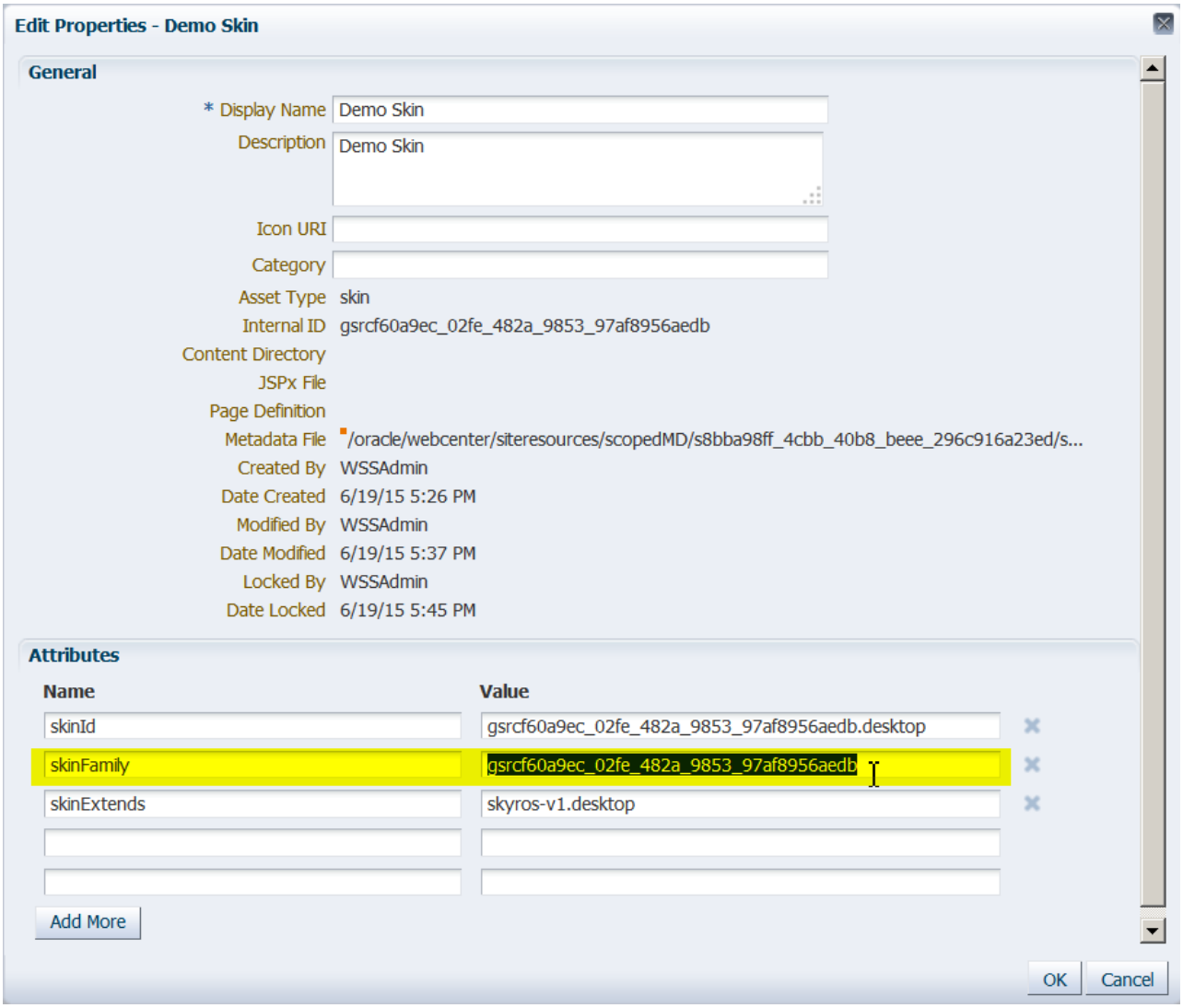
7 Edit the source. When the modification is complete, click **Save**, then **Close**.



8 To activate and use the **Demo Skin**, copy the **skinFamily** name of the new skin, the click **Actions** -> **Edit Properties**.

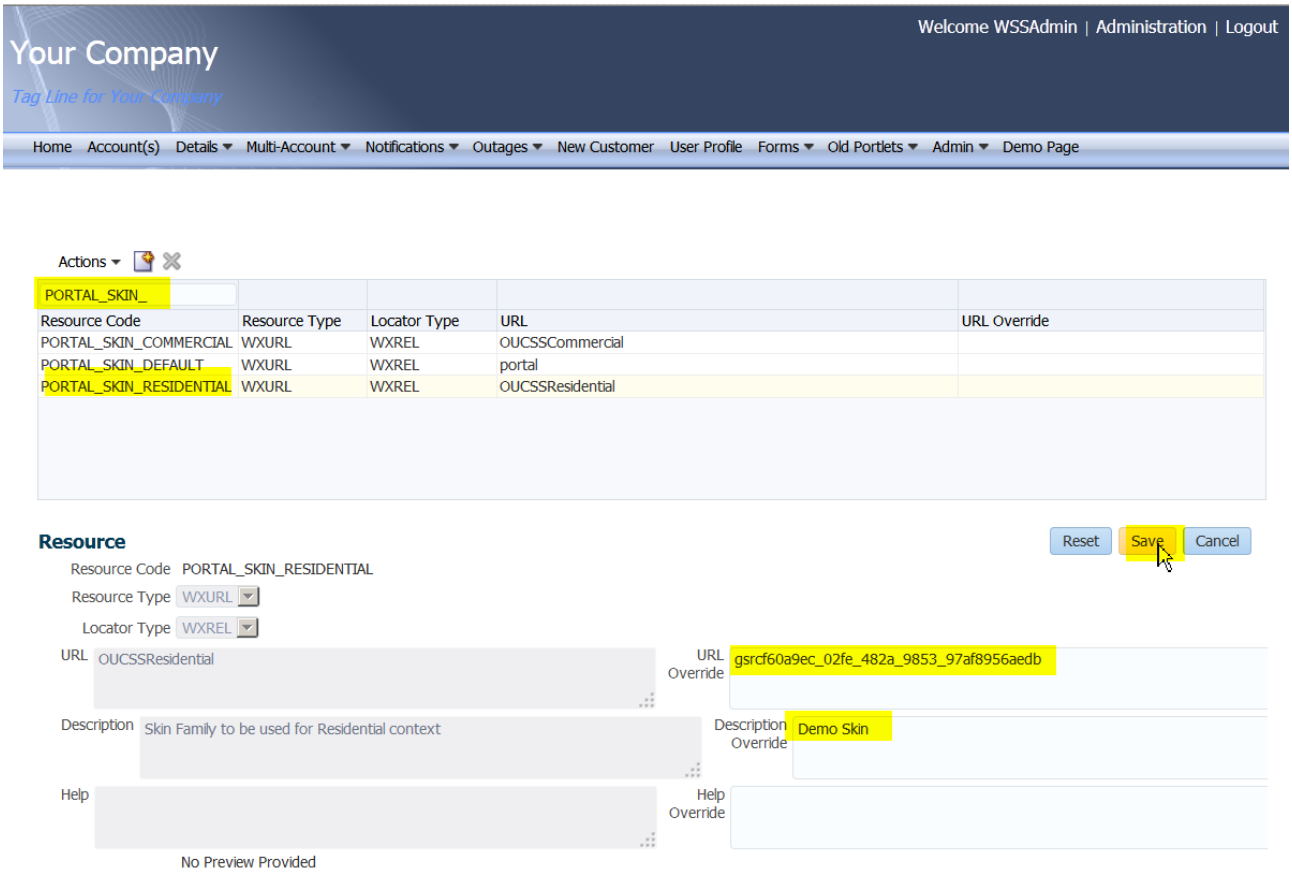


9 Select and copy the **skinFamily** attribute value from the **Edit Properties** dialog.



10 Click the **Back to Portal** link and then select the **Admin > Resources** page.

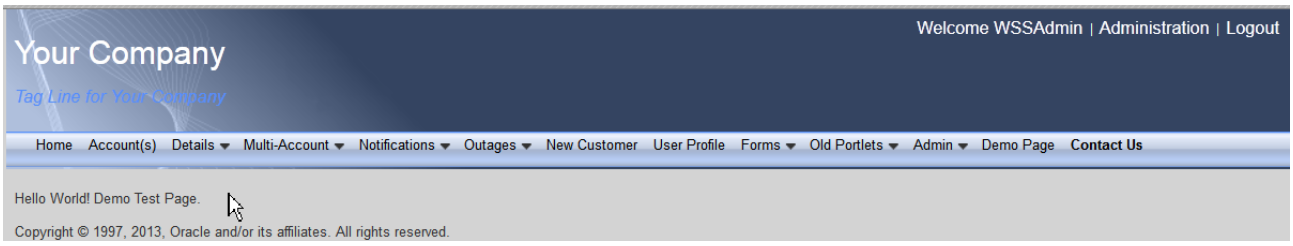
11 Filter the resources with resource codes **PORTAL_SKIN_** . Select a row and update the **URL Override** property with the **skinFamily** name.



12 Save your changes and repeat the same process to update another Context with the custom Site Template.

13 Restart the OUCSS Portal managed server and verify that you changes are reflected in the portal.

Sample: Body background color style changed to light gray:



Appendix A

Portal Labels

Portal labels and their descriptions are mentioned in the table below.

LABEL	DESCRIPTION
APP_TITLE	Application Title
DASHBOARD_TITLE	Dashboard Title
ACCOUNTS_TITLE	Accounts Title
ACCOUNT_LIST_TITLE	Accounts List Title
ACCOUNT_SEARCH_TITLE	Account Search Title
BILLING_HISTORY_TITLE	Billing History Title
FINANCIAL_HISTORY_TITLE	Financial History Title
BUDGET_BILLING_TITLE	Budget Billing Title
VIEW_BILL_TITLE	View Bill Title
HOME_TITLE	Home Title
COMPARE_RATES_TITLE	Compare Rates Title
USAGE_DETAILS_TITLE	Usage Details Title
PERSONAL_INFO_TITLE	Personal Info Title
OUTAGE_DETAIL_TITLE	Outage Detail Title
REPORT_OUTAGE_TITLE	Report Outage Title
SERVICE_MGMT_TITLE	Service Management Title
START_SERVICE_TITLE	Start Service Title

STOP_SERVICE_TITLE	Stop Service Title
TRANSFER_SERVICE_TITLE	Transfer Service Title
OUTAGES_TITLE	Outage Title
OUTAGE_MAP_TITLE	Outage Map Title
OUTAGE_TABLE_TITLE	Outage Table Title
REPORT_PUBLIC_OUTAGE_TITLE	Report Public Outage Title
NEW_CUSTOMER_TITLE	New Customer Title
SCALAR_READ_TITLE	Scalar Read Title
CHANGE_PASSWORD_TITLE	Change Password Title
CHANGE_USER_INFO_TITLE	Change User Info Title
MANAGE_ADDRESS_TITLE	Manage Address Title
MANAGE_PHONE_TITLE	Manage Phone Title
MANAGE_EBILL_TITLE	Manage EBill Title
MANAGE_AUTOPAY_TITLE	Manage Autopay Title
MANAGE_BILLNOTIFY_TITLE	Manage Bill Notify Title
ONETIME_PAY_TITLE	One-time Payment Title
PAY_ARRANGEMENT_TITLE	Pay Arrangement Title
RESET_PAGE_TITLE	Forgot User Id/Password Title
PASSWORD_RESET_TITLE	Password Reset Title
RETREIVE_USERID_TITLE	Retrieve User ID Title
REGISTER_TITLE	Register Title
EDUCATION_TITLE	Education Title
ENVIRONMENT_TITLE	Environment Title
EDU_SERVICE_TITLE	Service Title
EDU_SAFETY_TITLE	Safety Title
EDU_CONSERVATION_TITLE	Conservation Title
ABOUT_TITLE	About Title
ABOUT_ORACLE_TITLE	About Oracle Title
ABOUT_UGBU_TITLE	About UGBU Title
ABOUT_OUCSS_TITLE	About OUCSS Title
HOME_BANNER_DESC	Home Banner
HOME_SECTION_DESC	Home Section Links
LOGIN_TITLE	Login Title
USER_PROFILE_TITLE	User Profile Title

WEBCENTER_RC_ITEMS	WebCenter Items
ENROLL_WRAPPER	Enroll Wrapper
ENROLL	Enroll
ADMIN_USER_TITLE	Admin User Title
ADMIN_LABEL_TITLE	Label Title
ADMIN_LOOKUP_TITLE	Lookup Title
ADMIN_SECURITY_TITLE	Security Title
ADMIN_MESSAGE_TITLE	Message Title
ADMIN_TRAIN_TITLE	Train Title
ADMIN_RESOURCE_TITLE	Resource Title
ADMIN_ACCESS_TITLE	Access Title
ADMIN_EDGEAPP_TITLE	Edge Application Title
ADMIN_TITLE	Admin Title
ADMIN_SYSTEM_TITLE	Admin Configuration Options
ADMIN_LOB_TITLE	Line of Business Title
ADMIN_PORTLET_TITLE	Portlets Title
ADMIN_LANGUAGE_TITLE	Language Title
ADMIN_OFFERS_TITLE	Offers Title
OUCSS_ADMIN_TASKFLOWS	OUCSS Admin Taskflows
VIEW_ACCOUNT_TITLE	View Account Title
DETAILS_MENU	Details Menu Title
BANNER_PROMOTION_TITLE	Promotional Banner Title
ACCOUNT_SUMMARY_TITLE	Account Summary Title
OUCSS_BASE_TASKFLOWS	OUCSS Base Taskflows Title
BASE_RC_TITLE	Base Title
BILLING_AND_PAYMENT_RC_TITLE	Billing and Payment Title
BILLING_AND_PAYMENT_RC_DESC	OUCSS Billing and Payment Taskflows Title
CUSTOMER_SERVICE_RC_TITLE	Customer Service Management Title
TASKFLOWS_RELATED_TO_OUCSS_SER	Taskflows related to OUCSS Service Management
ADMIN_RC_TITLE	Admin Title
ADMIN_RC_DESC	Taskflows related to OUCSS Administration.
OFFERS_RC_TITLE	Offers Title
OFFERS_RC_DESC	Taskflows related to OUCSS Offers
BASE_RC_DESC	Taskflows related to OUCSS Base/Core functionality

OUCSS_TASKFLOWS_TITLE	OUCSS Taskflows Title
OUCSS_TASKFLOWS_RC_TITLE	OUCSS Taskflows Title
OUCSS_COMM_TASKFLOWS_RC_TITLE	OUCSS Commercial Taskflows Title
ERR_UNAUTH_TITLE	Un-Authorized Title
ERR_UNAUTH_ERROR_DESC	User Un-Authorized Error Description
ERR_PAGENOTFOUND_TITLE	Page Not Found Error Title
ERR_PAGENOTFOUND_DESC	Page Not Found Error Description
ERR_TITLE	Error Title
ERR_DESC	Description for Unknown Exception
GOTO_HOME_LINK	Go to Home Link
LOGIN_TASKFLOW_RC_TITLE	Login Taskflow Title
LOGIN_TASKFLOW_RC_DESC	Description for Login Taskflow with Context enabled
USAGE_COMPARE_TITLE	Usage Compare Title
USAGE_AGGREGATE_TITLE	Usage Aggregate Title
COMMERCIAL_BCONTEXT_TITLE	Commercial Business Context Title
SET_ACCOUNTS_TITLE	Set Accounts Title
SET_MANAGEMENT_TF_TITLE	Set Management Title
CREATE_SET_TITLE	Create Set Title
NOTIFICATIONS	Notifications Title
NOTIFICATION_LIST	Notification Inbox Title
NOTIFICATION_CHANNELS	Notification Profile Title
NOTIFICATION_PREFERENCES	Notification Preferences Title
CUSTOMER_SEARCH_TITLE	Customer Search Title
ACCOUNT_SEARCH	Account Search Title
NAME_AND_ADDRESS_SEARCH	Name and Address Search
SEARCH_RESULTS_GRID	Search Results
ENTER_FORM	Enter Form
FORMS_MANAGEMENT	Forms Management Title
LOG_AN_ISSUE	Log an Issue Description
FORM_LIST	Form List Title
UPDATE_FORM	Update Form Title
SCALAR_USAGE_DETAILS	Scalar Usage Details Description
ACCOUNT_DOCUMENTS_TITLE	Account Documents Title
PREPAID_BALANCE_AND_CHARGES	Prepaid Balance And Charges Title

PREPAID_ESTIMATES_AND_COST	Prepaid Estimates And Cost Title
SERVICE_CHARGES_TITLE	Service Charge To Date Title
USAGE_OVERVIEW_TITLE	Usage Overview Title
CONSUMPTION_SUMMARY_TITLE	Consumption Summary Title
ALERTS_TITLE	Alerts Title
OLD_SUPPORTED_PORTLETS	Old (Supported) Portlets Title
MULTIACCOUNT	Multi-Account Title
VALIDATE_EMAIL_TITLE	Validate Email
CSR_VIEW_TITLE	CSR View
MANAGE_AUTOMATIC_PAYMENT	Manage Automatic Payment Title
AUTOMATIC_PAYMENT	Automatic Payment Title
STANDARD_MOBILE_PROMOTION	Standard Mobile Promotion
ACCOUNT_ACCESS_TITLE	Account Access Title
ACCOUNTS	Accounts Page Title
NAVIGATION	Navigation
ACCESSIBILITY	Accessibility Link
ACCESSIBILITY_DESC	Accessibility Link Description
ACCESSIBILITY_TITLE	Accessibility Popup Title
SCREEN_READER	Accessibility Screen Reader Option
HIGH_CONTRAST	Accessibility High Contrast Option
LARGE_FONTS	Accessibility Large Fonts Option
OK_BUTTON	Accessibility popup Ok button label
CANCEL_BUTTON	Accessibility popup Cancel button label
LOGOUT	Logout link label
CHANGE_ACCOUNT_ACCESS	Change Account Access
LOGOUT_CONFIRM_MSG	Message in popup before logout
ORACLE_LOGO	Oracle logo description
GOBACK_LINK	Go Back button label
MENU_LBL	Menu Label
PROMOTION__OFFERS	Promotion Offers Title
ENTER_METER_READING	Enter Meter Reading title
DEPRECATED_TASKFLOWS	Deprecated taskflows folder title in resource catalog
REMOVE_ACCOUNT_TASKFLOW	Remove Account taskflow label in resource catalog
ACCOUNT_NICK_NAME_TASKFLOW	Account Nickname taskflow label in resource catalog

INVITE_TASKFLOW	Invite taskflow label in resource catalog
ACCOUNT_ACCESS_TASKFLOW	Account Access taskflow label in resource catalog
DEPRECATED_TASKFLOWS_DESC	Description for deprecated taskflows folder in resource catalog.