

# **Oracle Utilities Customer Self Service**

Whitepaper:  
Customizing and Extending  
OUCSS Custom Portals

Release 2.0.1

**E36186-02**

September 2012

---

Oracle Utilities Customer Self Service

Whitepaper: Customizing and Extending OUCSS Custom Portals

E36186-02

Copyright © 2012, 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 software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

#### U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007).  
Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

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 which 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.

This software or hardware and documentation may provide access to or information on 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. 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.

---

---

## Table of Contents

Overview .....	4
Additional Resources.....	4
Abbreviations .....	4
Preparation and Procedures .....	4
Software Requirements.....	4
Pre-Installation Tasks .....	4
Assumptions .....	4
Customizing Page Template .....	5
Customize and Configure Existing Page Template .....	5
Exporting an Existing Page Template .....	12
Importing a Page Template.....	14
Customizing the Navigation Model.....	15
Customizing and Configuring an Existing Navigation Model .....	16
Exporting an Existing Navigation Model .....	21
Importing a Navigation Model .....	22
Creating a New Navigation Model .....	23
Extending the OUCSS Portal.....	24
About Portal Security .....	25
Adding New Taskflows and Pages.....	25
Deploying the extend.oucss.portal Shared Library.....	29
Customizing and Configuring an Existing Resource Catalog .....	32
Accessing Custom Pages.....	38
Accessing User Account Details .....	41

## Overview

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

## Additional Resources

For additional details on functionality and procedures described in this document, see the *Oracle Utilities Customer Self Service Installation Guide* and the *Oracle Utilities Customer Self Service Implementation Guide*. Both documents are available on the Oracle Technology Network (OTN) in the Oracle Utilities Documentation section (<http://www.oracle.com/technetwork/documentation/utilities-155272.html>, expand **Oracle Utilities Customer Self Service**, then choose **View Library** or **Download** part number E36170-01).

Further information is available in the *Developer's Guide for Oracle WebCenter Portal (11.1.1.6.0)*, available on the OTN at [http://docs.oracle.com/cd/E23943\\_01/webcenter.1111/e10148/toc.htm](http://docs.oracle.com/cd/E23943_01/webcenter.1111/e10148/toc.htm).

## 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

## 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 should be completed before you can customize a OUCSS custom portal.

- Ensure that your Oracle JDeveloper version is 11gR1 PS5 (11.1.1.6.0).
  - Ensure that the WebCenter Extension bundle is enabled in JDeveloper.
- Ensure that your OUCSS (custom) Portal has been deployed and running.

### Assumptions

- The OUCSS Portal is built using the “WebCenter Portal Framework” application template.
- The Custom Portal is deployed onto an Oracle WebLogic (10.1.6) server.

- The OUCSS Schema is created and populated with seed data. For running the taskflows, a data source with jndi name “jdbc/OUCCSSDS” 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/OUCCSS” 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.
- 

## Customizing Page Template

The default OUCSS Portal contains the out-of-box page templates Global and Swooshy, as provided by the Oracle WebCenter Framework Application template. The default page templates can be replaced by customizing the existing page templates or importing new page templates exported from another instance.

For more information on default and page template customization, see [http://docs.oracle.com/cd/E23943\\_01/webcenter.1111/e10148/jpsdg\\_id\\_resources.htm#BABJCCAC](http://docs.oracle.com/cd/E23943_01/webcenter.1111/e10148/jpsdg_id_resources.htm#BABJCCAC)

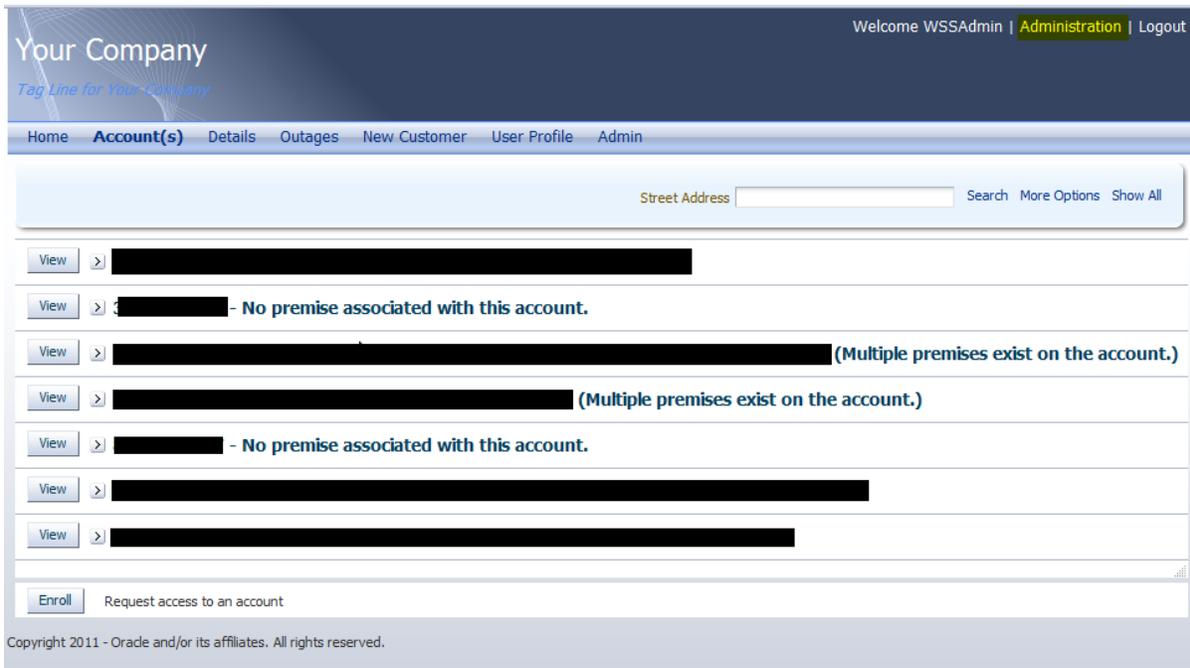
## Customize and Configure Existing Page Template

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

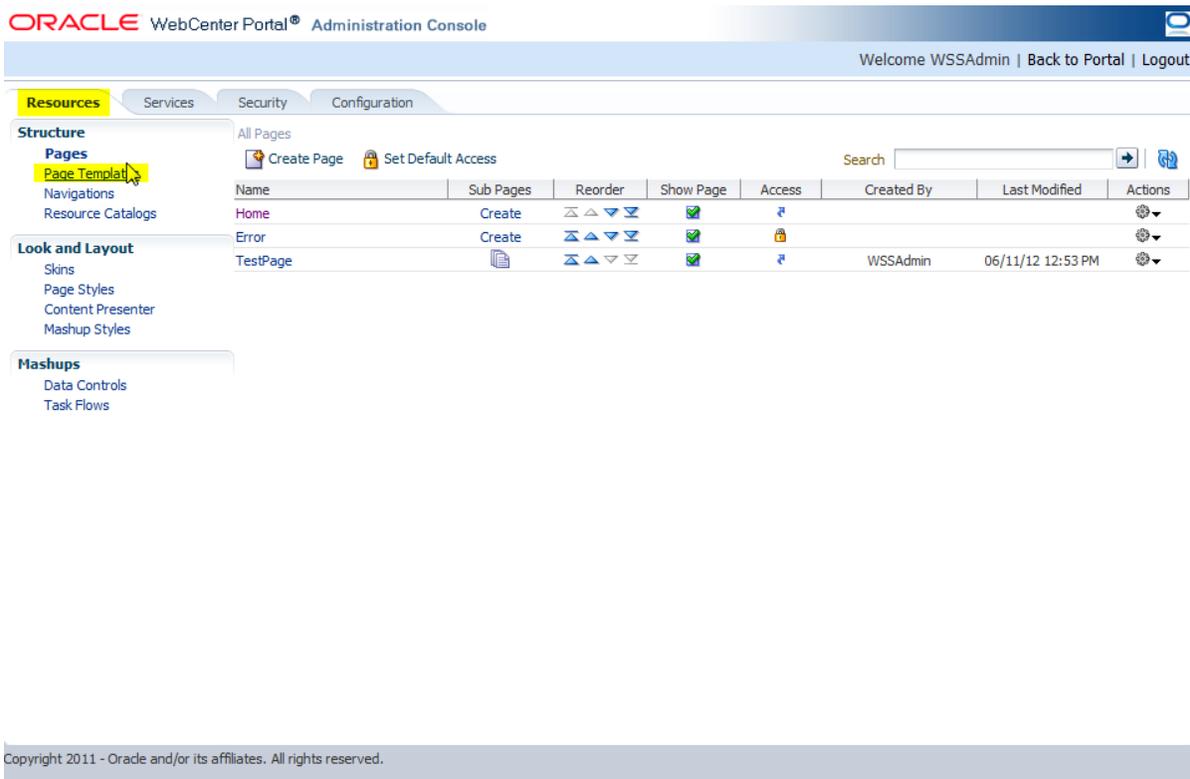
- 1 Log in to the OUCSS Portal with Administrator permission (WSSAdmin use is seeded with the required permission).



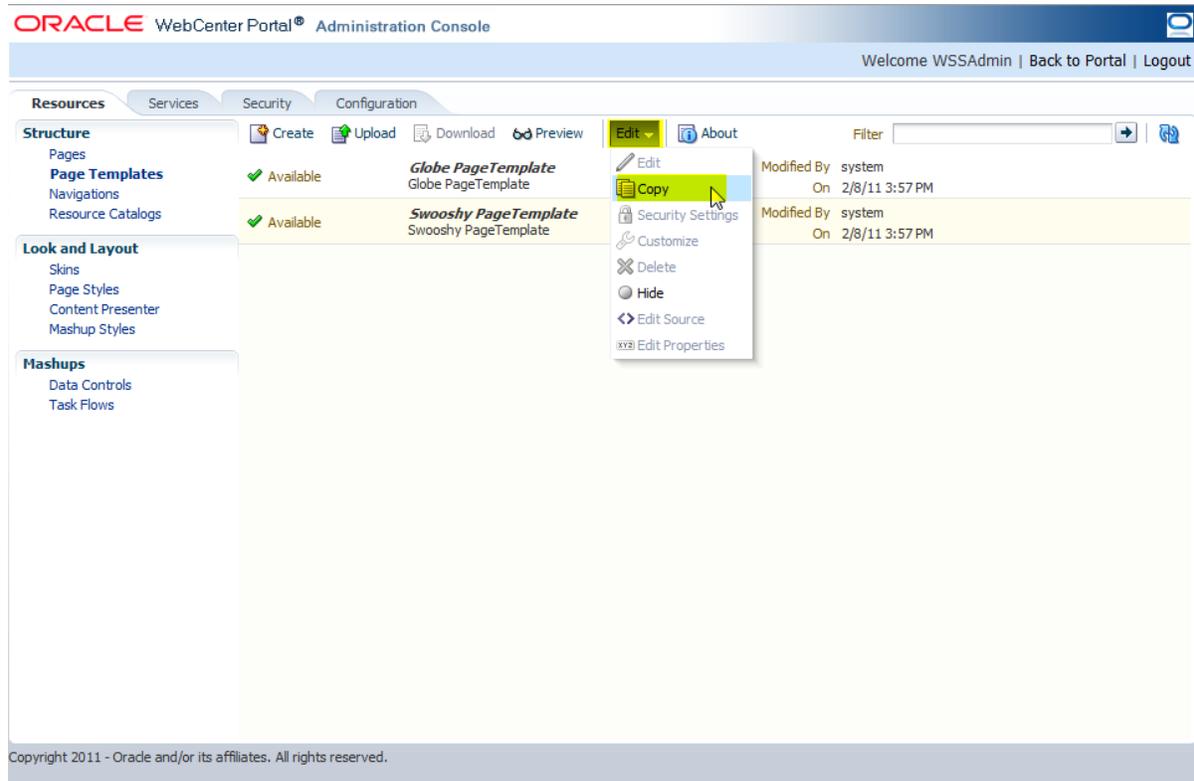
2 Click on the **Administration** link on the top-right section of the screen.



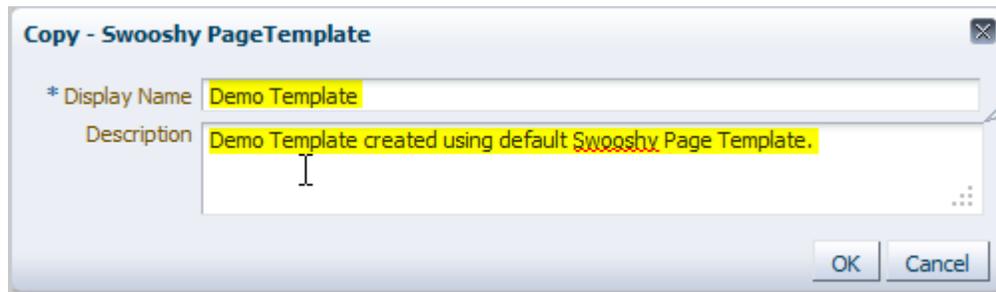
3 In the Administration console, click the **Resources** tab, then choose **Page Templates** in the **Structure** section:



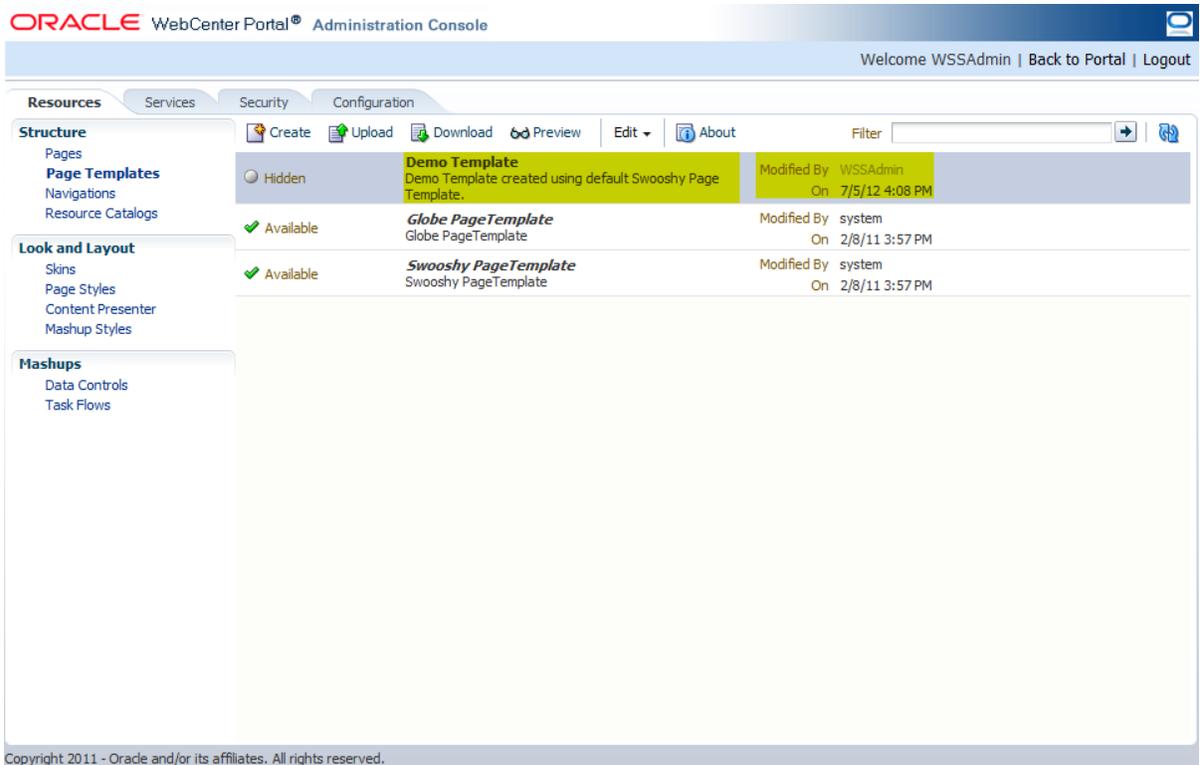
- 4 The Swooshy Page Template is read-only and already seeded. To customize it, make a copy of the Existing Swooshy Page Template. To make a copy:
  - A. Select the Swooshy Page Template from the list of Page Templates.
  - B. Click the **Edit** menu and select **Copy**.



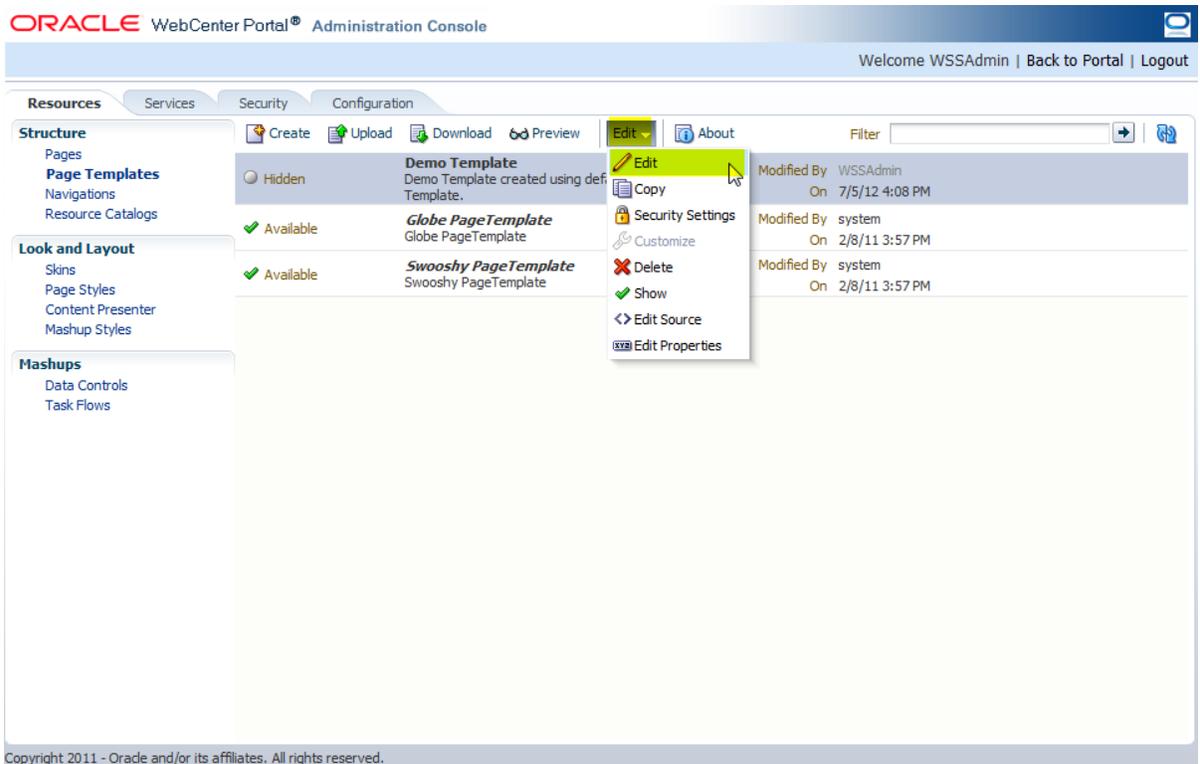
- C. Enter a name and description for the new template:



- D. Click **OK** to create the new template.

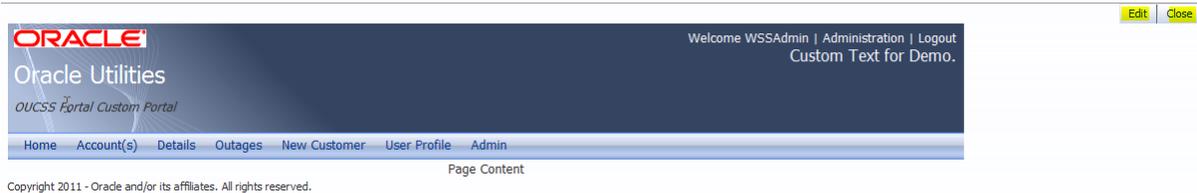
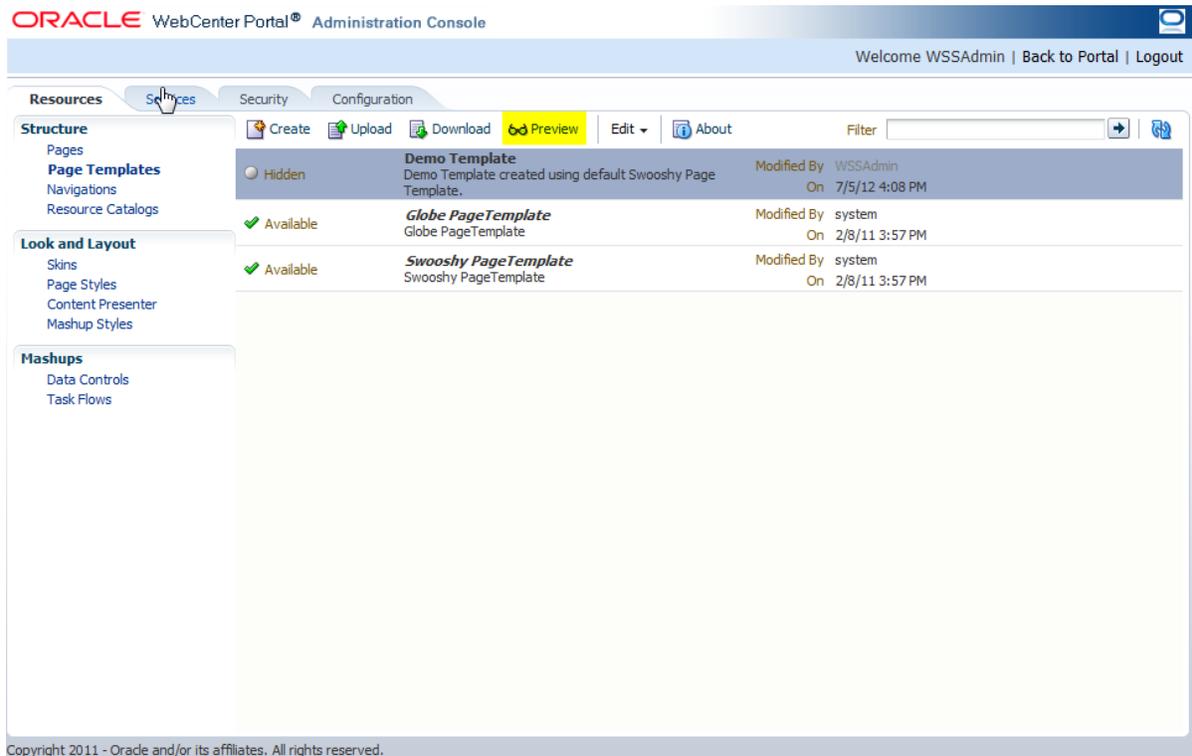


- 5 Select the new template, then click the **Edit** menu and choose **Edit** to open the template for customization.

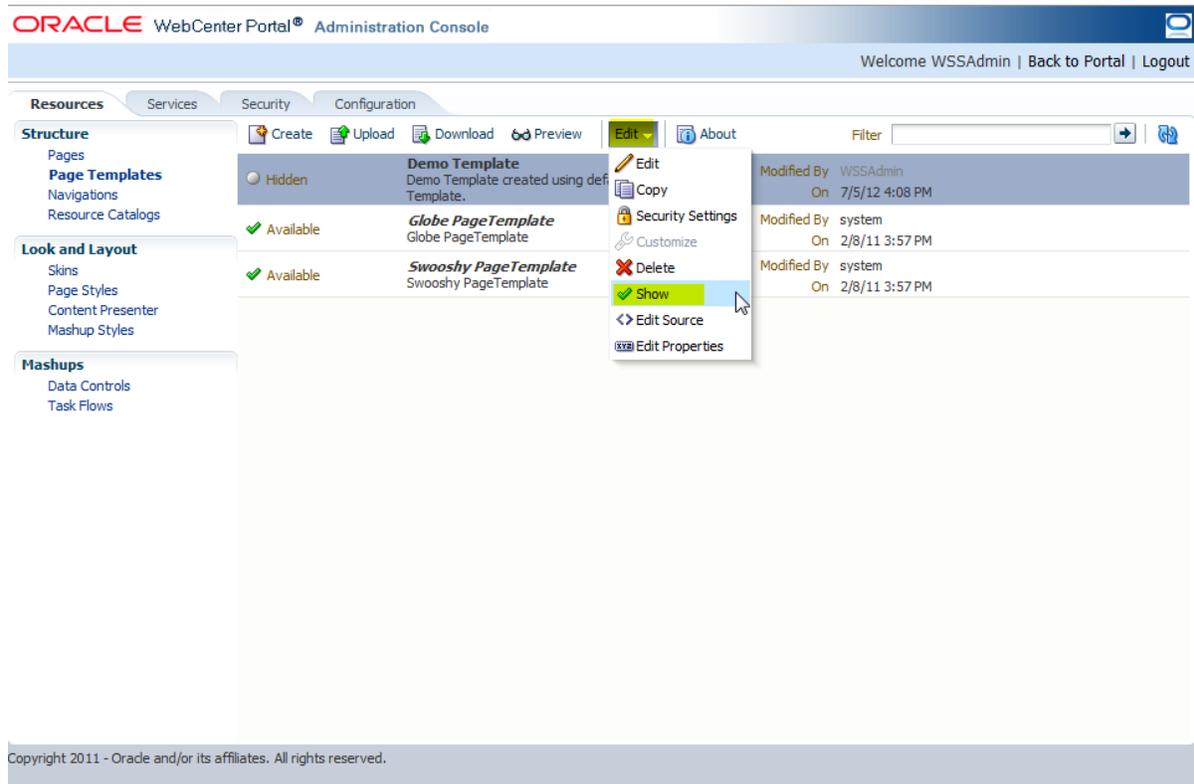




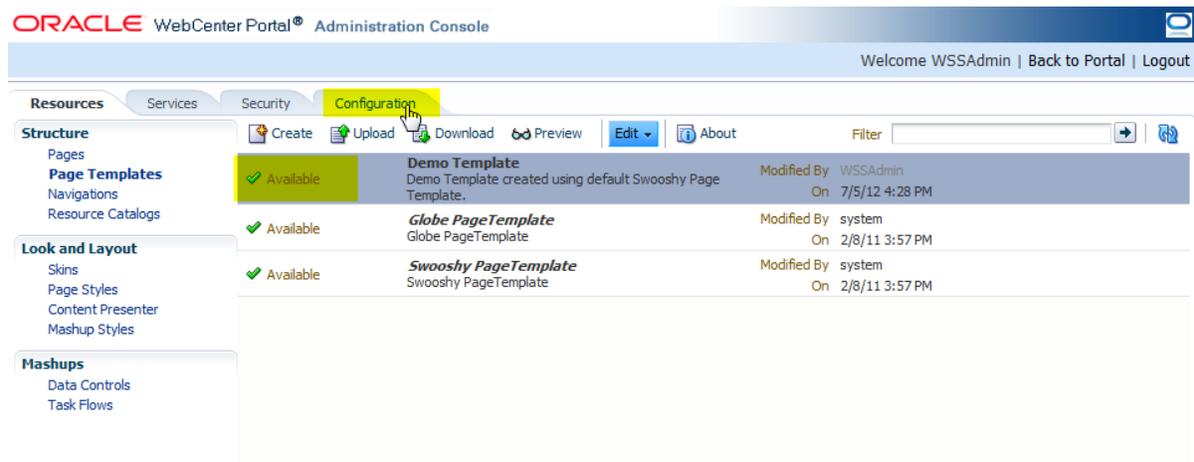
- 6 Customize as required, then click **Save**, then **Close**.
- 7 Click **Preview** to preview the changes. If not satisfied, click **Edit** on the **Preview** screen and modify the customization. When complete, click **Close**.



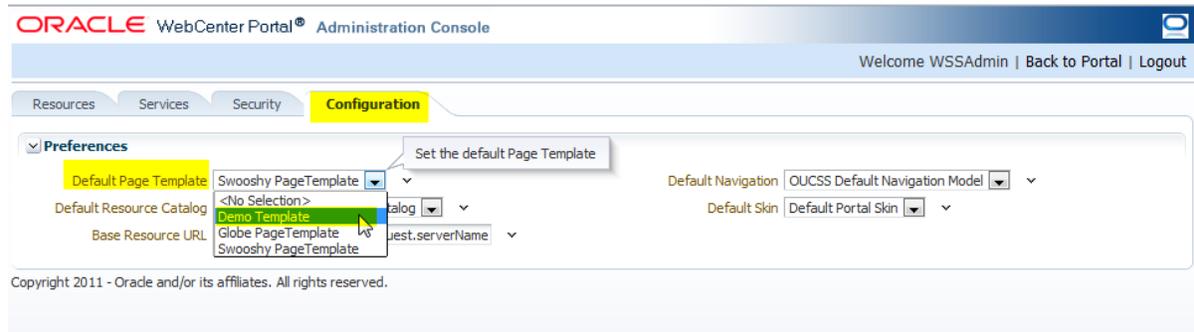
8 All newly-created templates are hidden by default. Unhide the template by clicking **Edit > Show**.



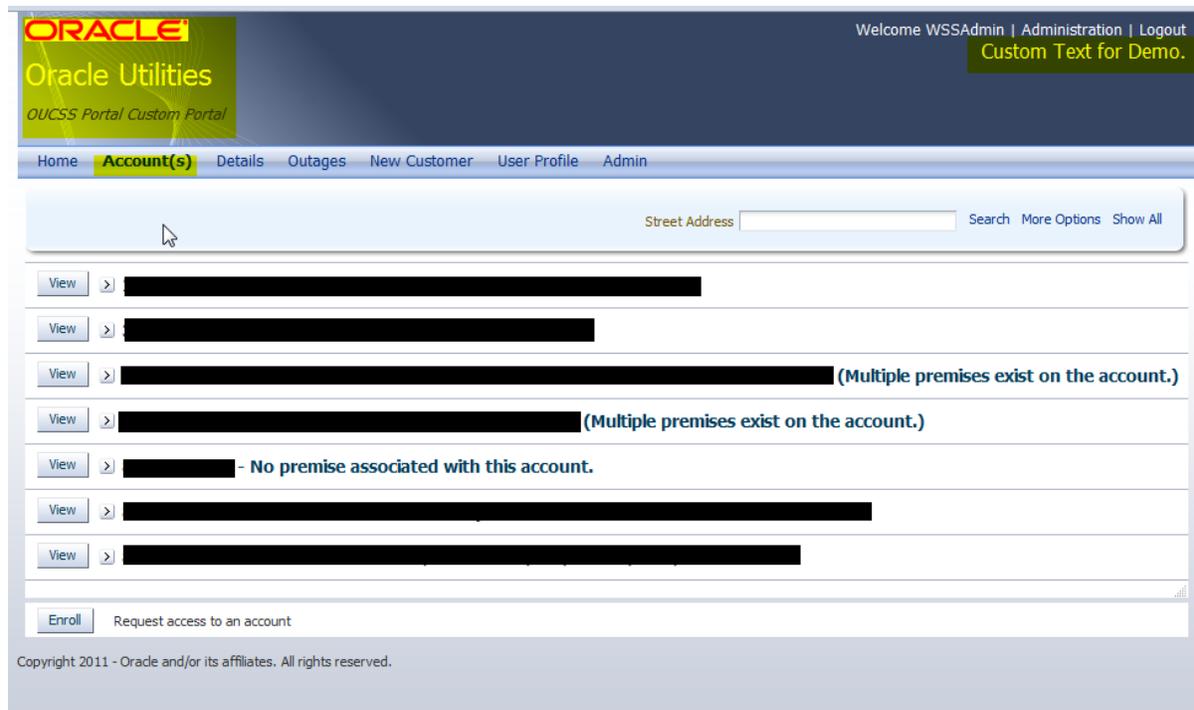
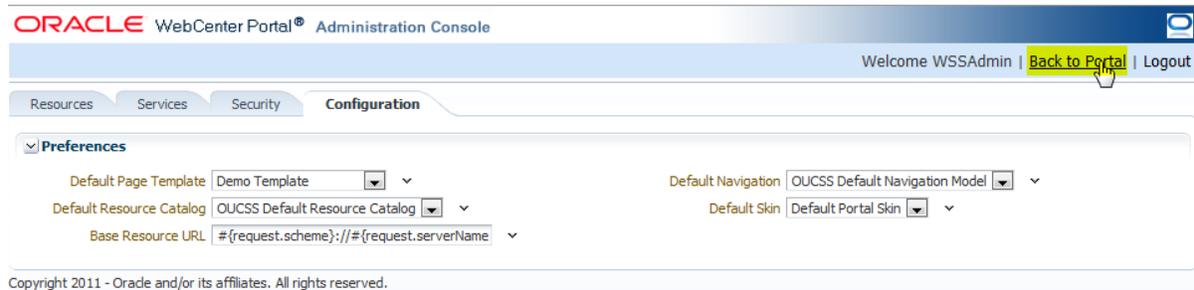
9 To use your new template as the site template, click the **Configuration** tab and select the new template in the **Default Page Template** list.



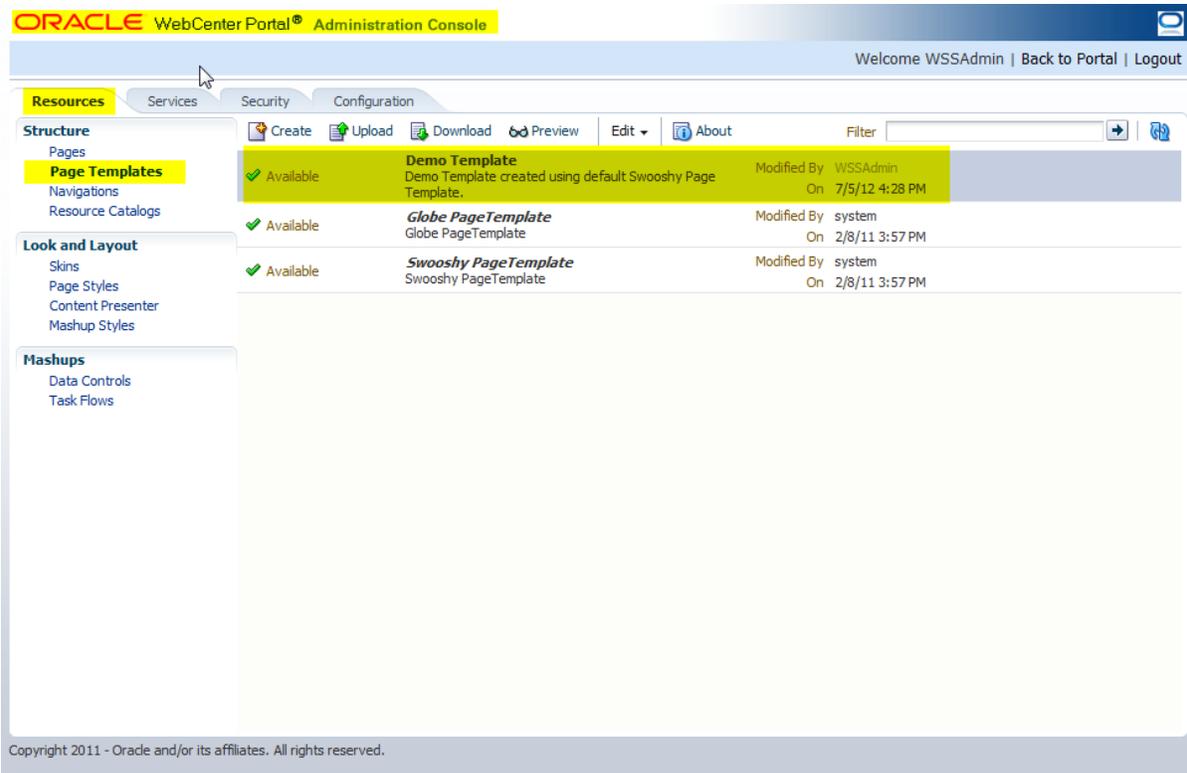
**IMPORTANT:** Selections made in the **Configuration** screen are automatically saved. Make sure that all entries are correct before clicking **Back to Portal**.



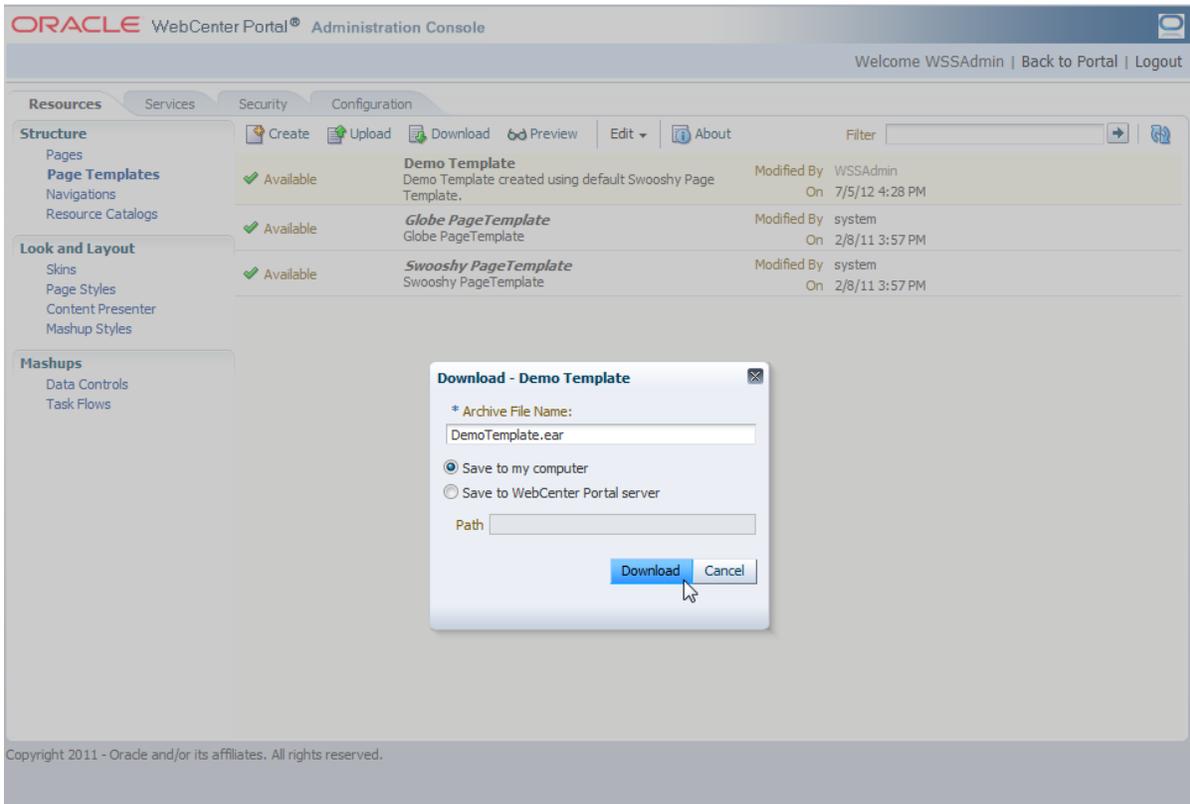
10 Click **Back to Portal** to return to the portal pages and check your new template.



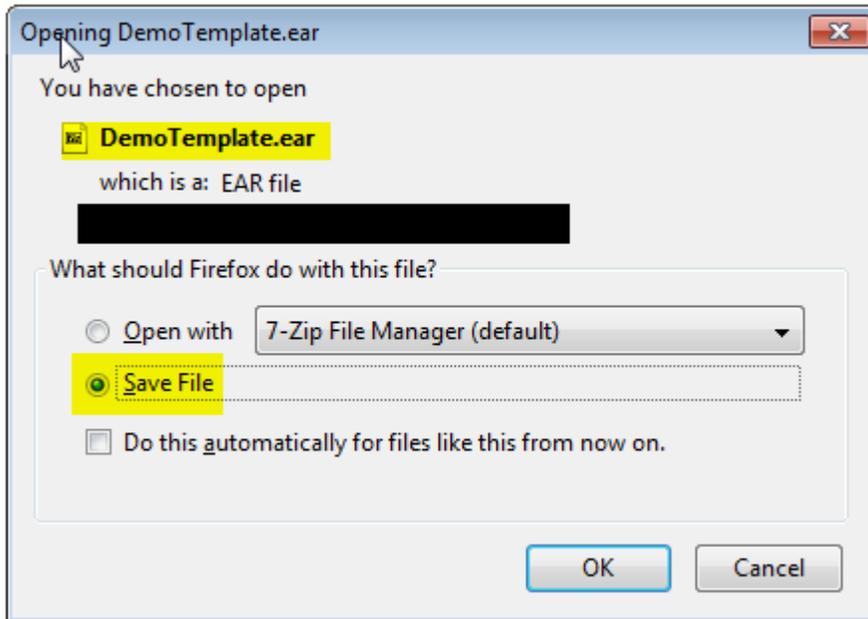




- 3 Click **Download** on the toolbar to open the Download dialog. Enter the Archive File Name, choose the location for the exported archive on your local machine, then click **Download**.



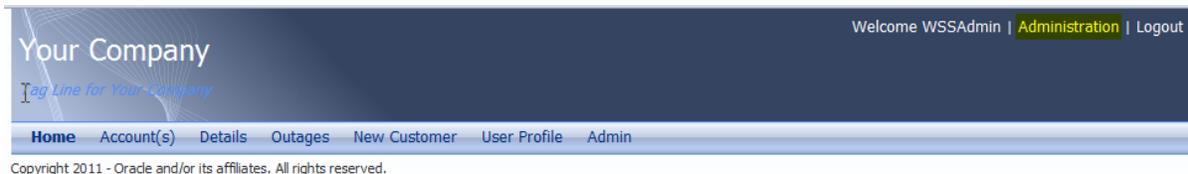
- In the **Opening** dialog, choose **Save File** to save the template, then click **OK**.



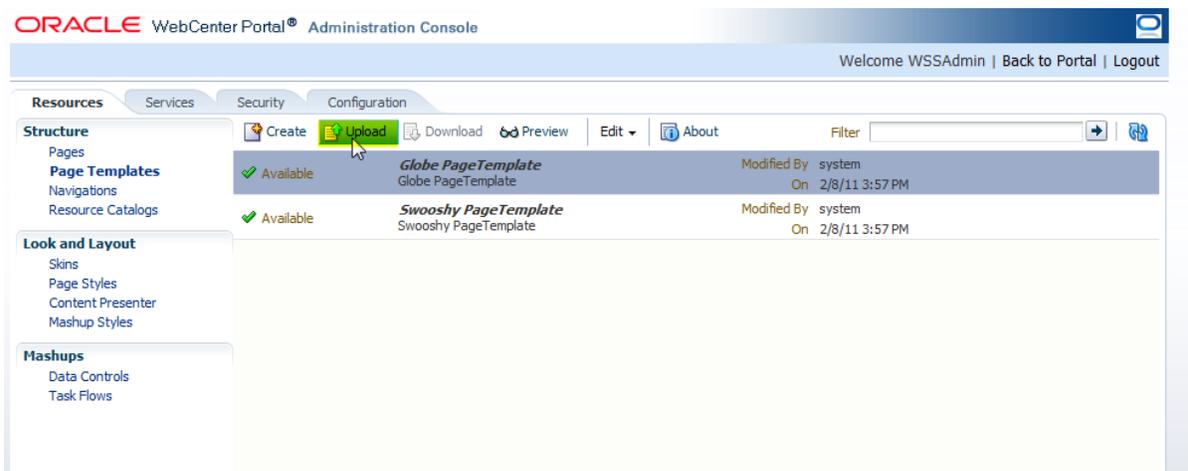
- Log out of the Administration Console.

## Importing a Page Template

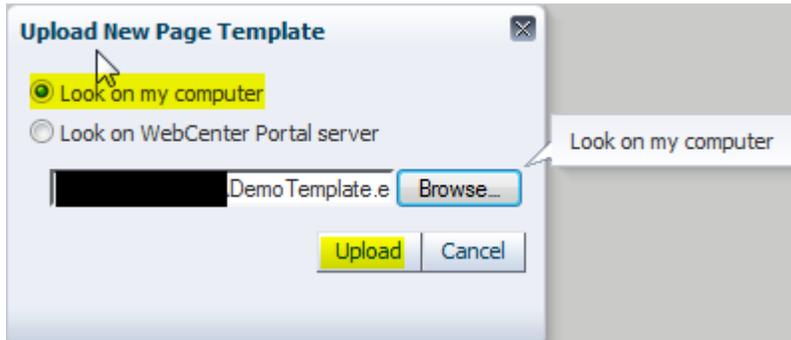
- Log in to the portal instance with Administrator privileges in the location in which the template is to be migrated/imported.



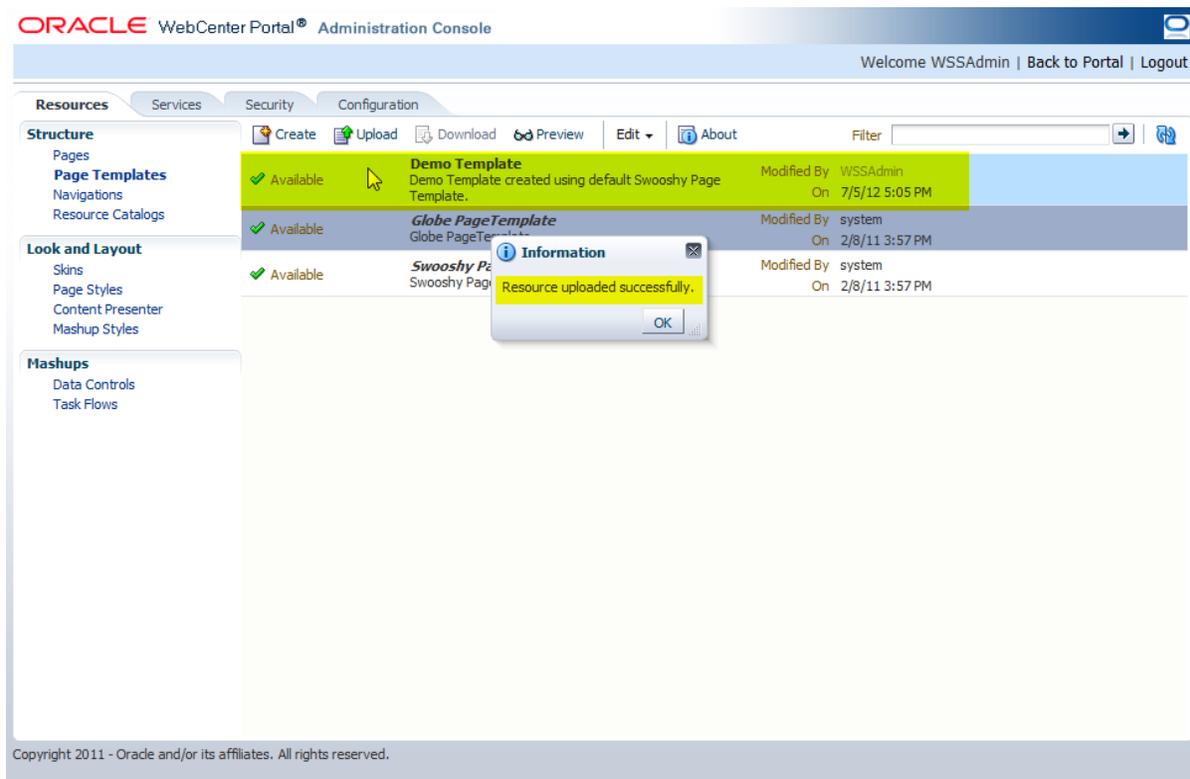
- Click **Administration** and go to the **Resources > Page Templates** page.
- Click **Upload**.



- 4 Select the **DemoTemplate.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 Click the **Configuration** tab and set the new page template as the default page template. For more on this step, see the topic, "[Customizing and Configuring 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 numerous changes are required, a new navigation model can also be created from scratch.

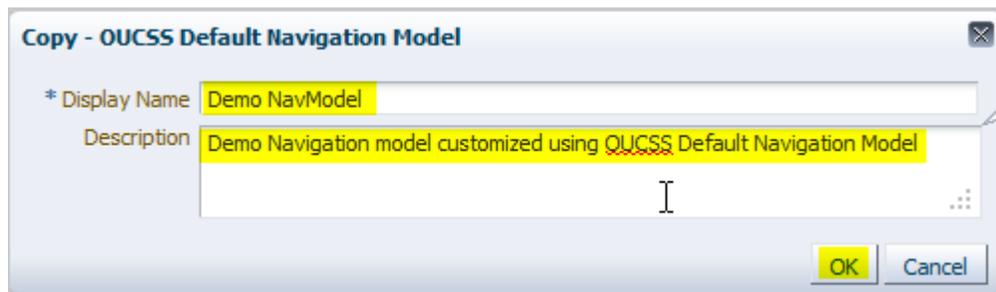
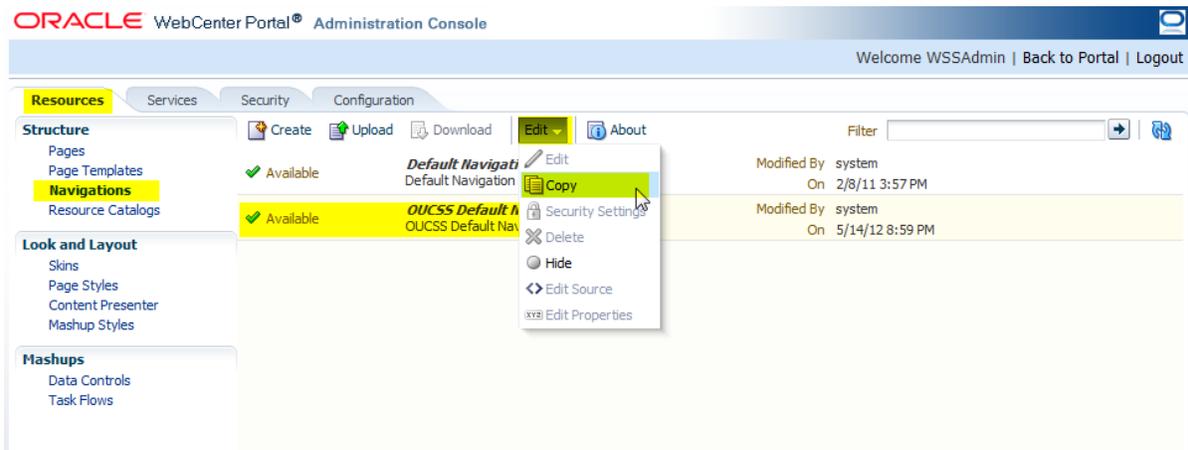
For additional information on customizing the navigation model, see [http://docs.oracle.com/cd/E23943\\_01/webcenter.1111/e10148/jpsdg\\_navigation.htm#BABJHFCE](http://docs.oracle.com/cd/E23943_01/webcenter.1111/e10148/jpsdg_navigation.htm#BABJHFCE)

## Customizing and Configuring an Existing Navigation Model

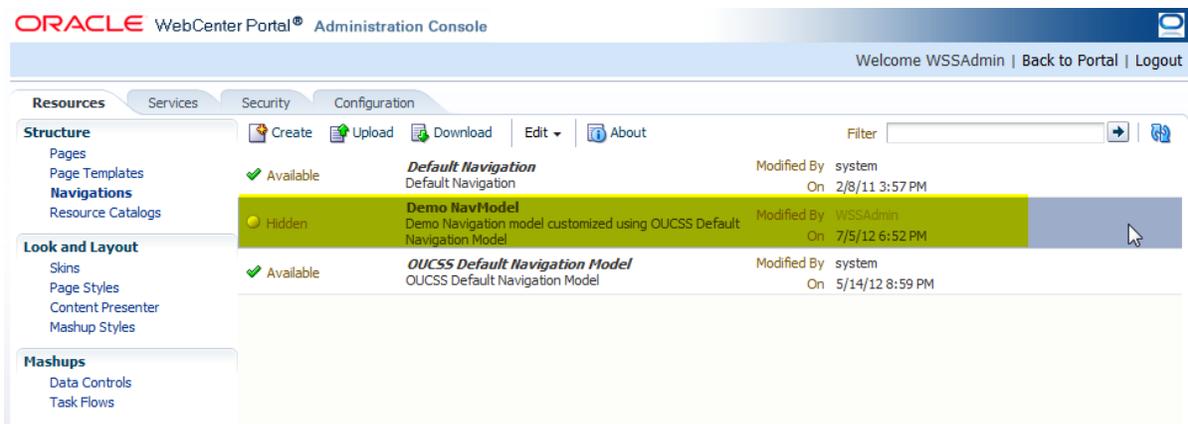
The default OUCSS Portal is configured to use the OUCSS Default Navigation Model. This navigation model is read-only and already seeded. To customize it, a copy of this navigation model must be created.

To customize the default navigation model:

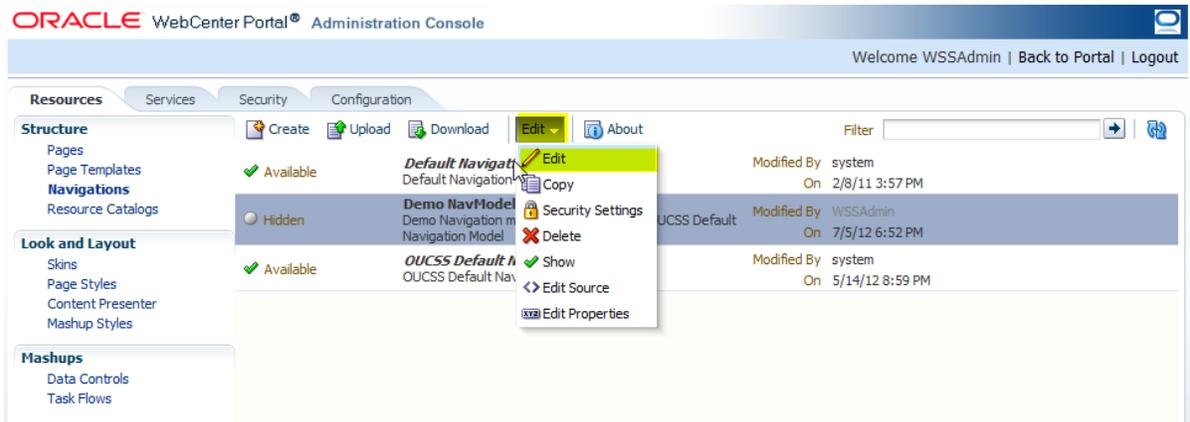
- 1 Log in to the OUCSS Portal with Administrator privileges and go to the Administration Console.
- 2 Choose the Resources tab, then click the **Navigations** link in the **Structure** section.
- 3 Select **OUCSS Default Navigation Model**, then click **Edit > Copy**.



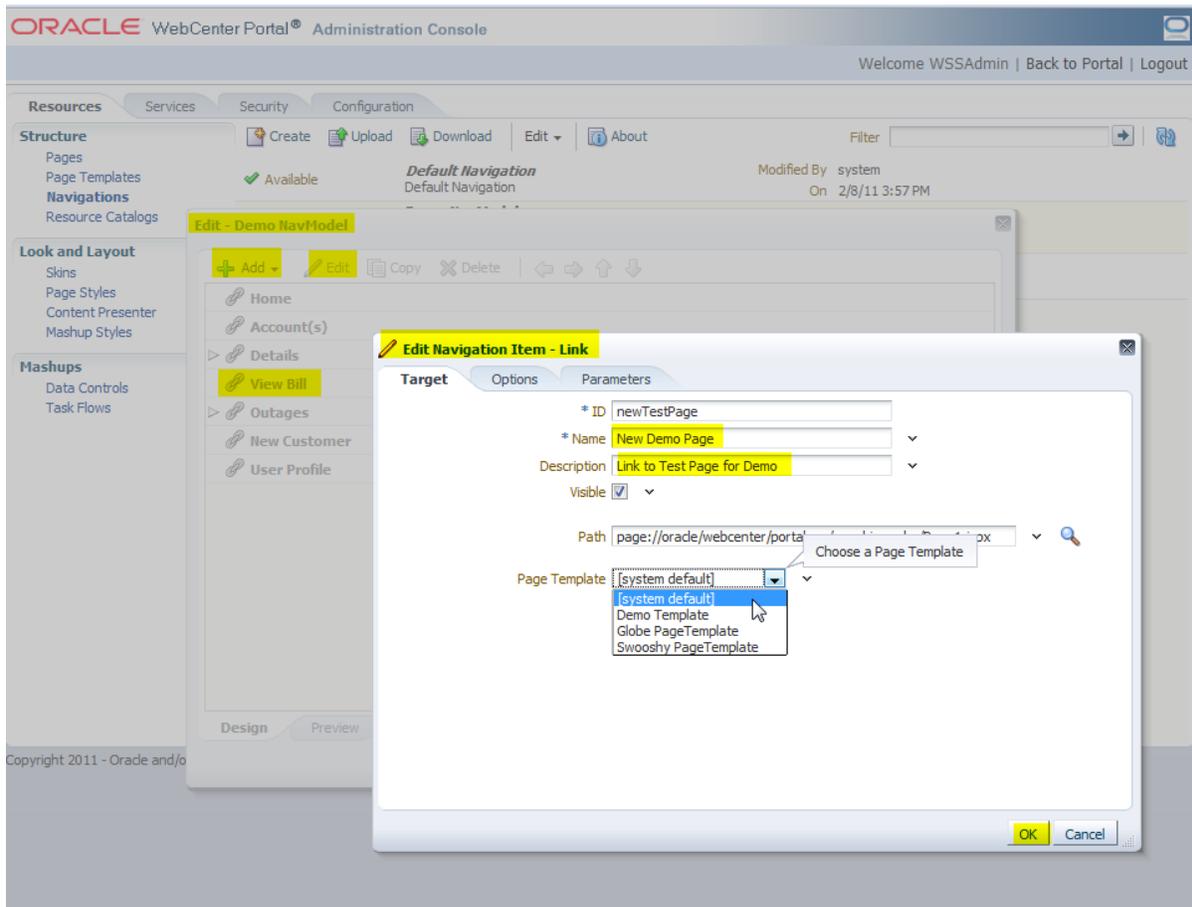
- 4 In the **Copy** dialog, enter a name and description for the new navigation model, then click **OK** to create the model.

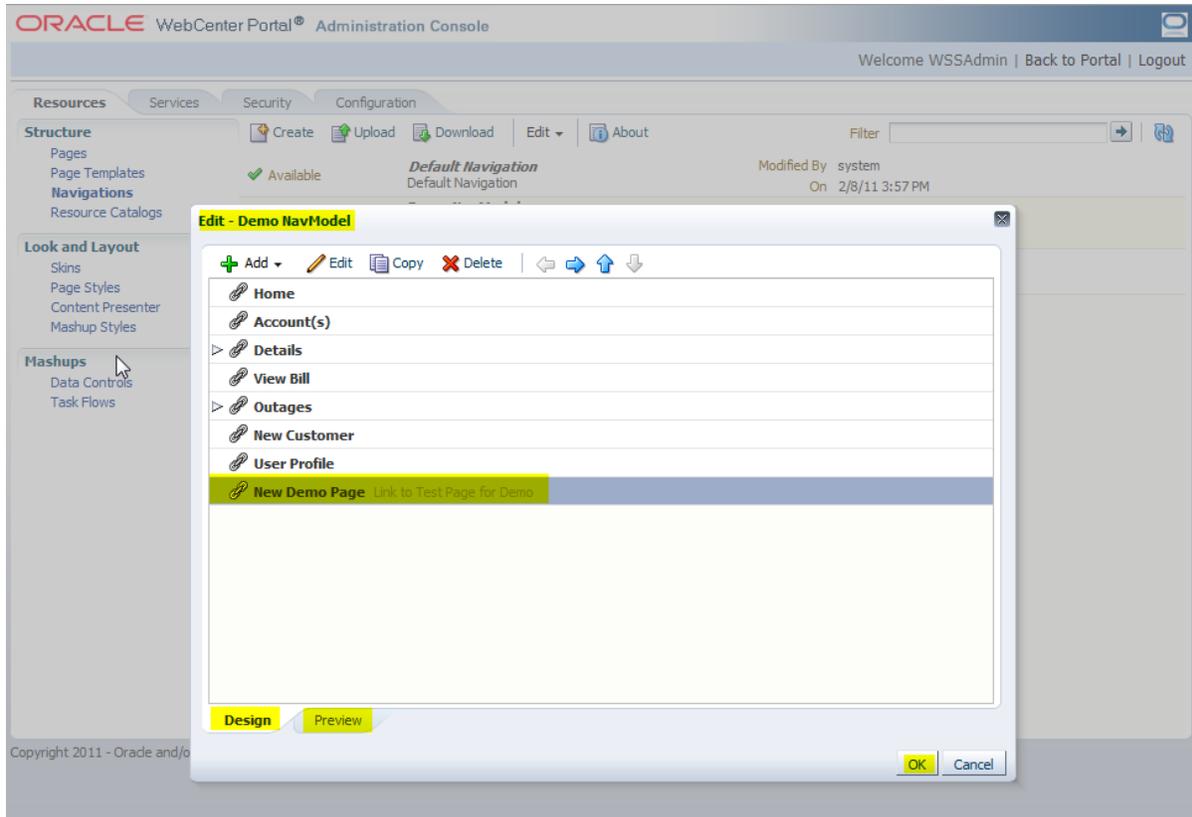


5 To begin customization, choose the new navigation model and click **Edit > Edit**.



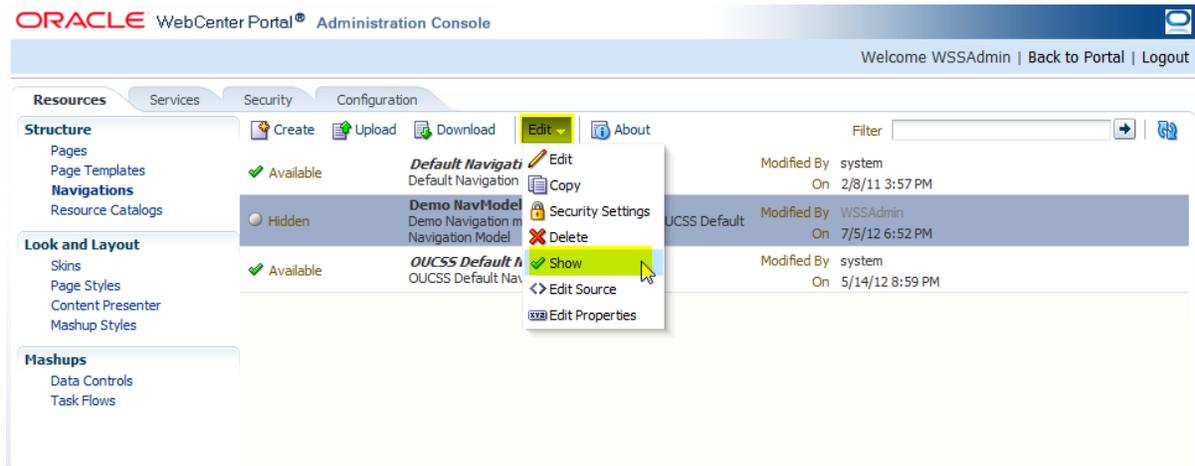
6 Customize the navigation model per your requirements, then choose the **Preview** tab at the bottom of the Edit dialog to verify the customized model. If satisfied with the customizations, click **OK**.



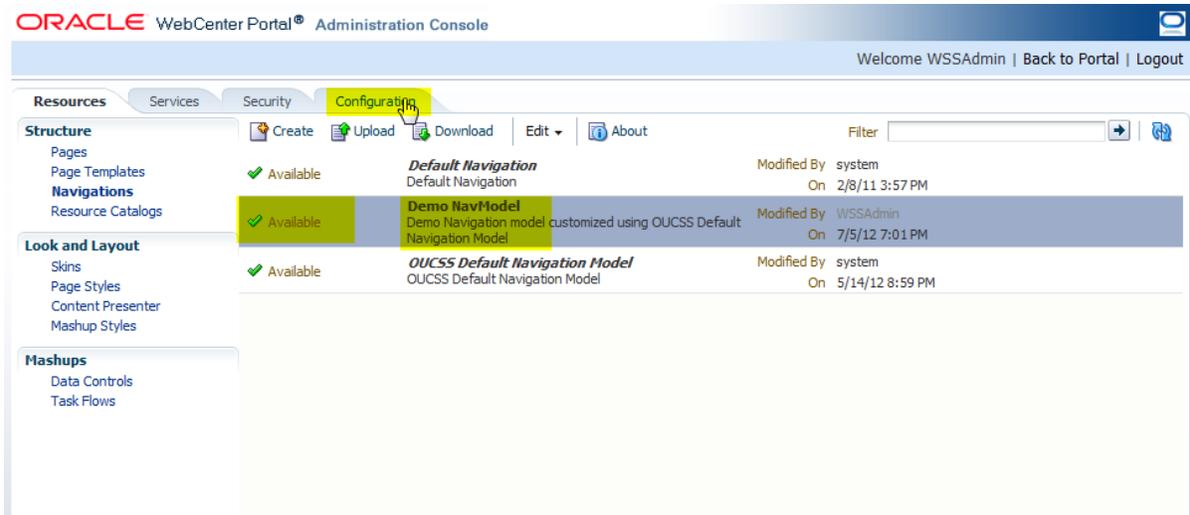




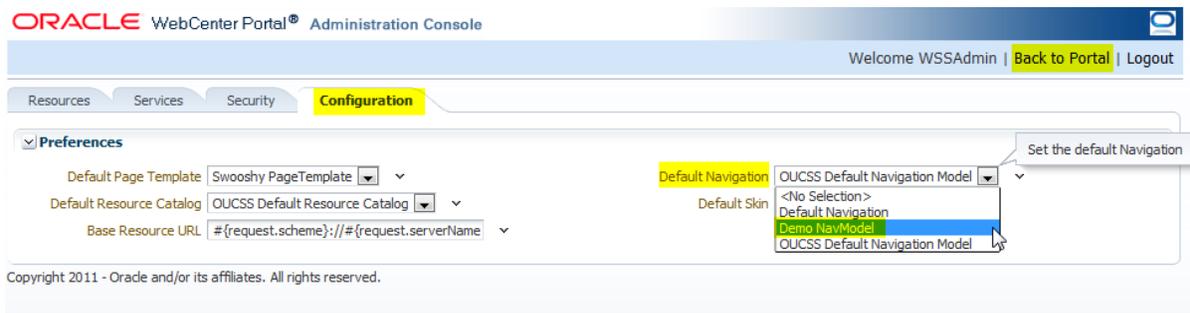
7 Newly-created navigation models are hidden by default. To unhide your new model, click **Edit > Show**.



8 Click the **Configuration** tab to set the new navigation model as the default.

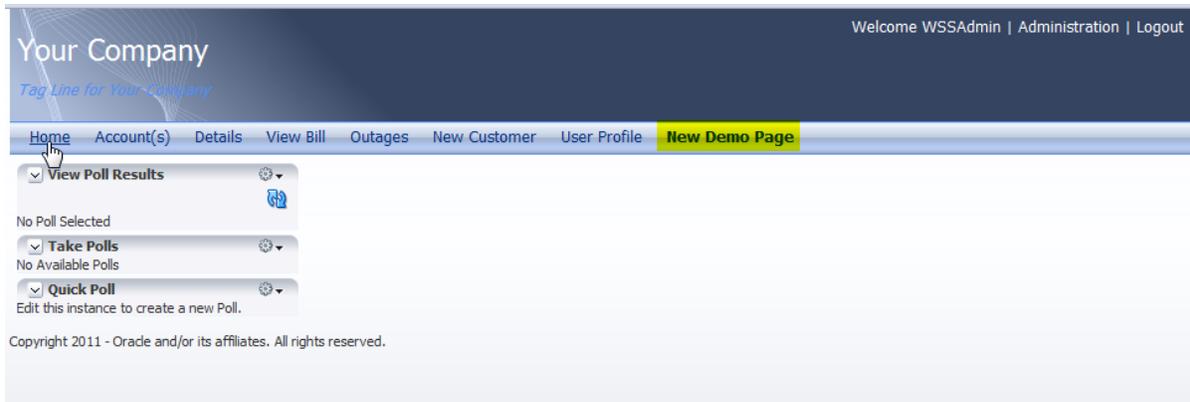


- Select your new navigation model (in this example, Demo NavModel) from the Default Navigation dropdown list.



- Click the **Back to Portal** link to exit the Administration Console and return to the portal application and verify that the navigation model has been updated.



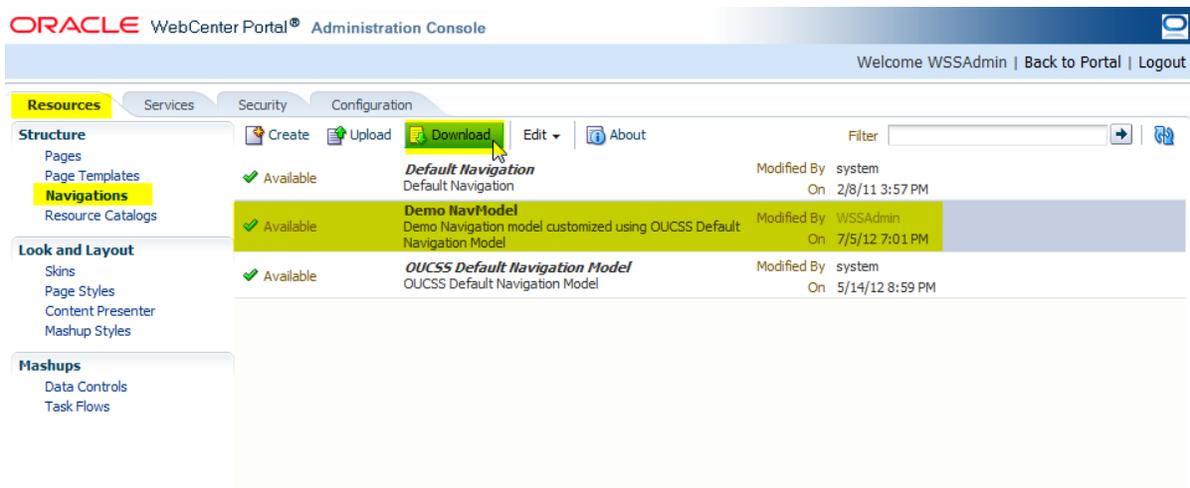


## Exporting an Existing Navigation Model

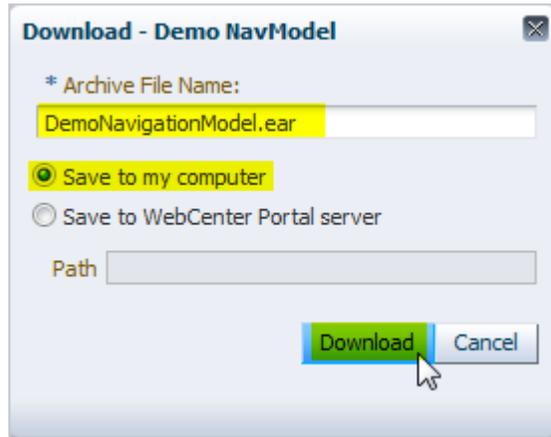
Navigation models from other WebCenter instances can be exported for migration and back up.

To export a navigation model:

- 1 Log in to the OUCSS Portal with Administrator privileges and go to the Administration Console.
- 2 Choose the **Resources** tab and click on the **Navigations** link.
- 3 Select the navigation model to migrate and click **Download**.



- 4 In the **Download** dialog, enter the archive name and choose the save location. Then click the **Download** button.

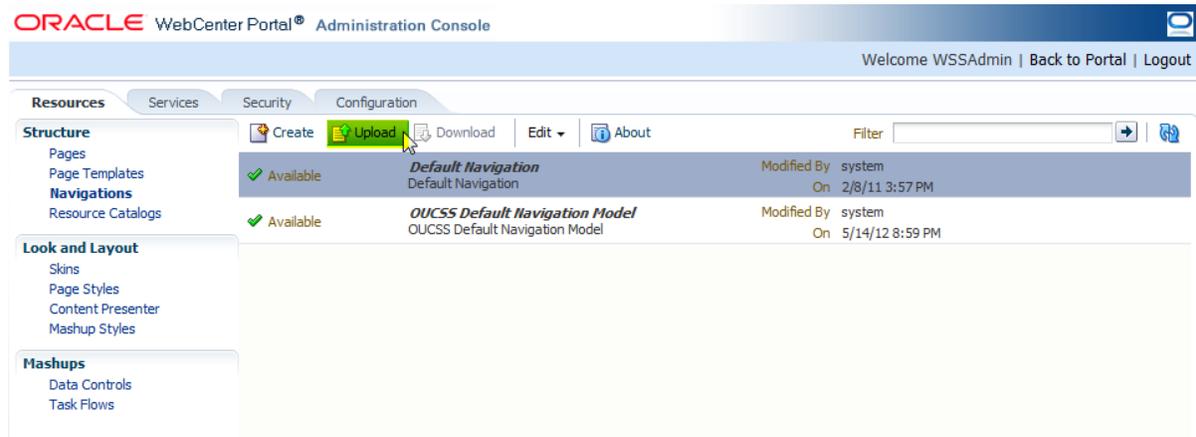


- 5 Save the file on the target location.

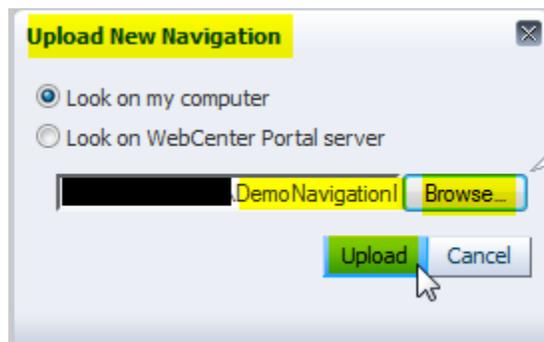
### Importing a Navigation Model

To import navigation models from other instances:

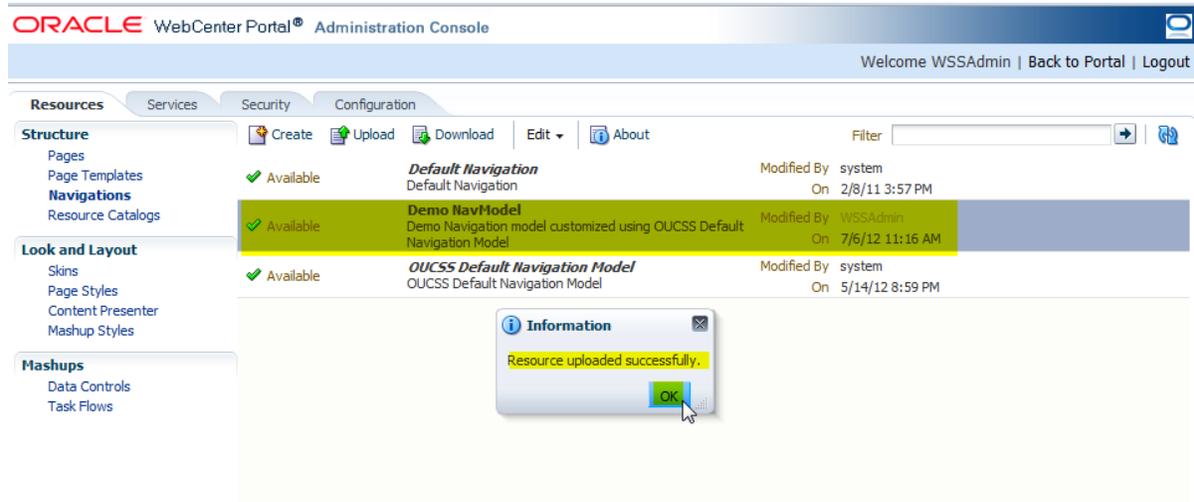
- 1 Log in to the OUCSS Portal with Administrator privileges and go to Administration Console.
- 2 Choose the **Resources** tab and click on the **Navigations** link.
- 3 Click **Upload**.



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



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



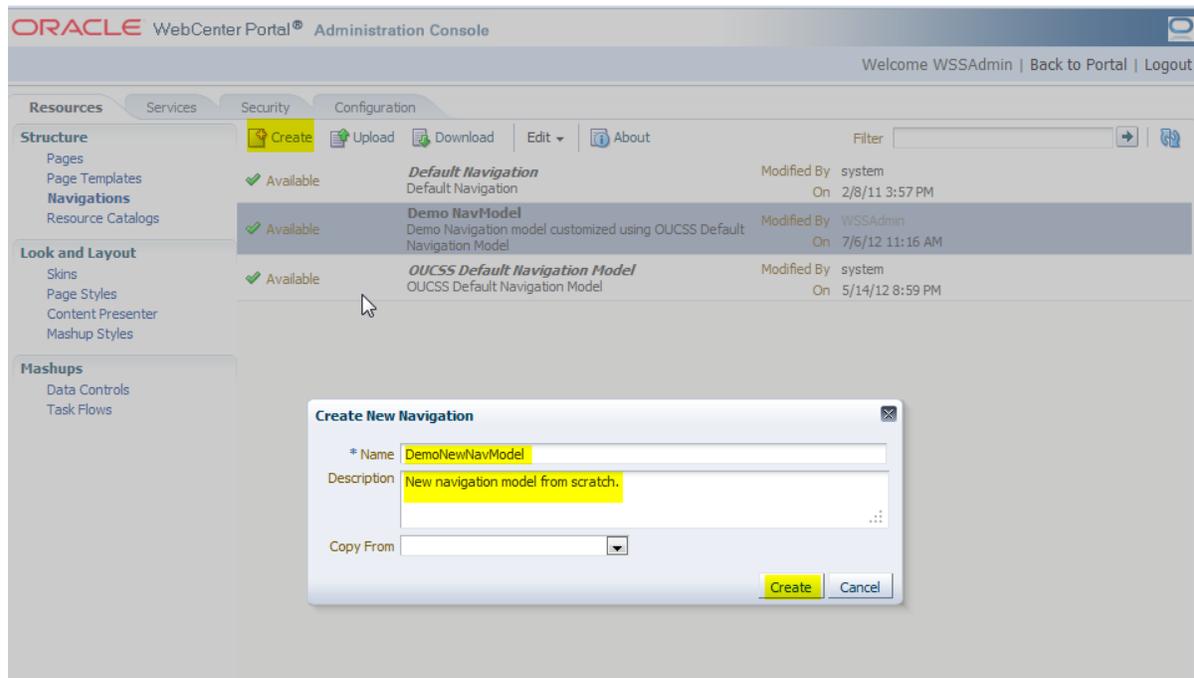
- To customize and configure the imported navigation model, see the topic, “[Customizing and Configuring Existing Navigation Model](#)”.

## Creating a New Navigation Model

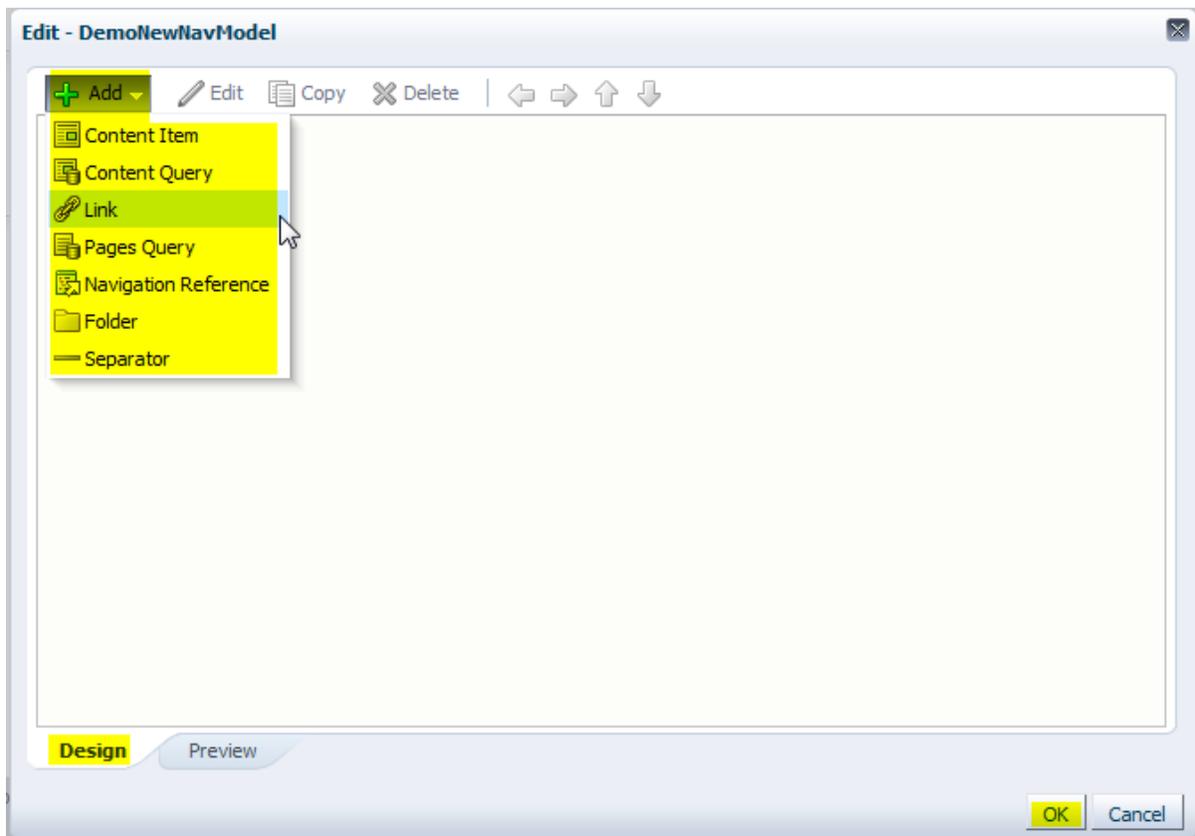
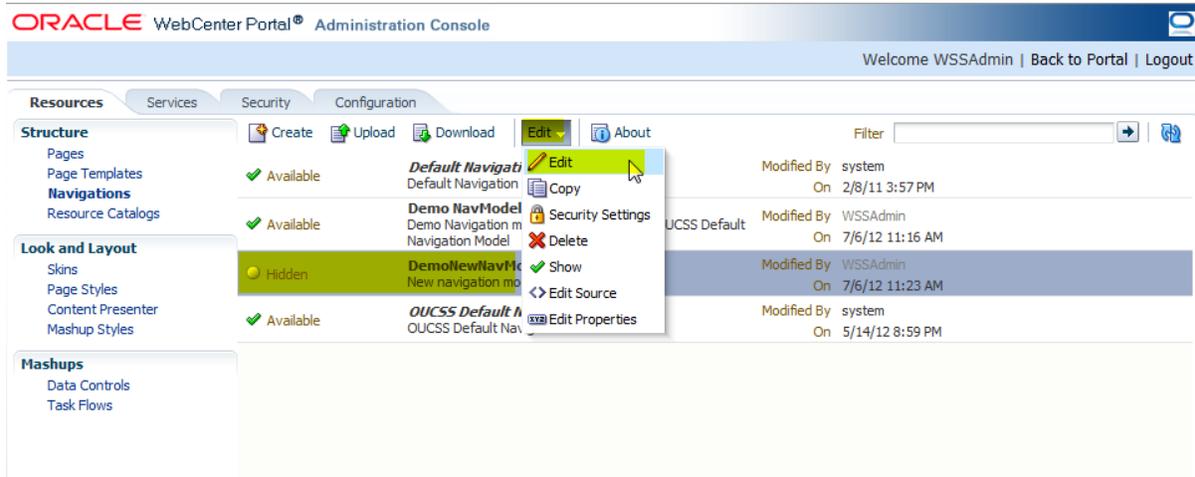
A new navigation model can be created from scratch to suit your portal needs.

To create a new navigation model:

- Log in to yjr OUCSS Portal with Administrator privileges and go to the Administration Console.
- Choose the **Resources** tab and click on the **Navigations** link.
- Click **Create**, on the top menu, enter the new model name and description in the **Create New Navigation** dialog, then click the **Create** button in the dialog.



- To configure the new navigation model, select the new model from the list and click **Edit > Edit**.



- For details on customizing and configuring the new navigation model, see the topic, “[Customizing and Configuring an Existing Navigation Model](#)”.

## Extending the OUCSS Portal

An OUCSS Portal can be extended with custom taskflows and pages by using **ExtendOUCSSPortal** workspace included in `ExtendOUCSSPortal.zip`. The `ExtendOUCSSPortal` workspace includes two projects: `OUCSSPortalADFLibrary` and `OUCSSPortalSharedLibrary`.

The OUCSSPortalADFLibrary project is used to add portal artifacts (taskflows and pages) and the OUCSSPortalSharedLibrary project is used for redeploying the **extend.oucss.portal** shared library.

## About Portal Security

An 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 security to work, however, the taskflow definition should be created in the <<ExtensionWorkSpace>>/OUCSSPortalADFLibrary/public\_html/WEB-INF/oracle/ugbu/ss/custom/secure and/or <<ExtensionWorkSpace>>/OUCSSPortalADFLibrary/public\_html/WEB-INF/oracle/ugbu/ss/custom/public folders.

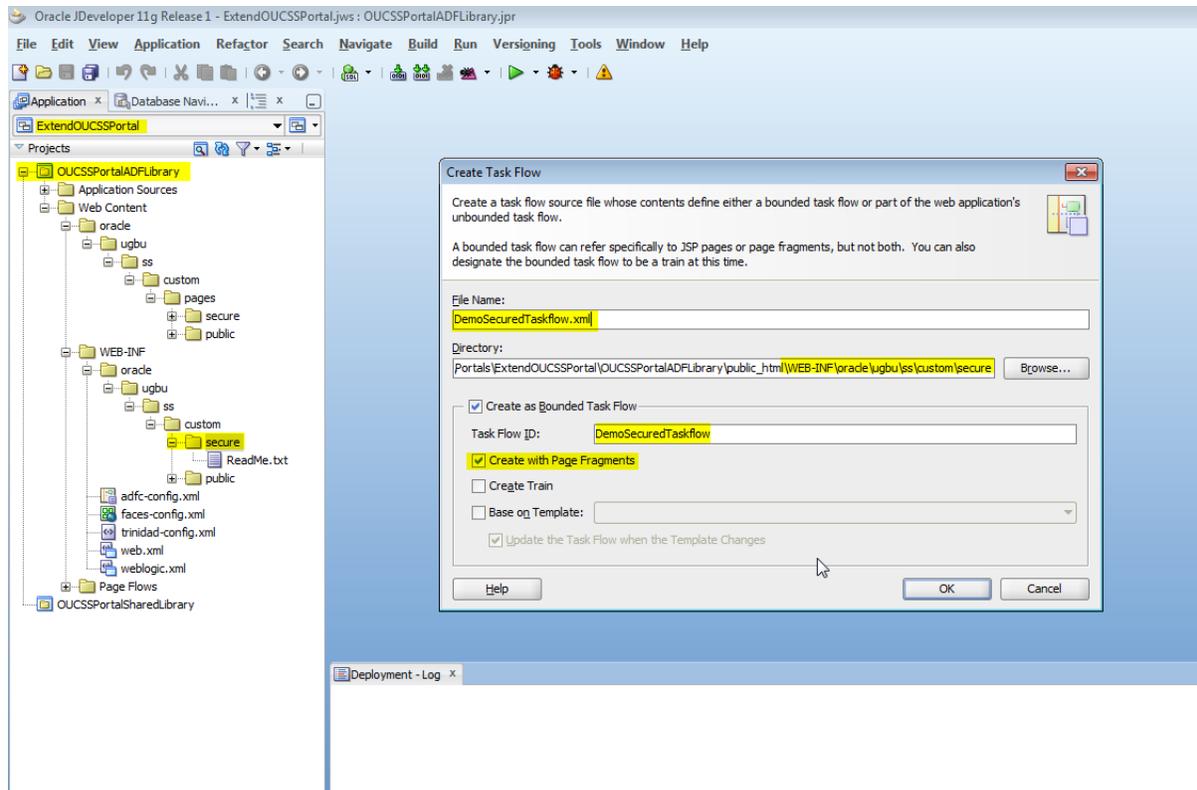
Custom Pages should also be created in the <<ExtensionWorkSpace>>/OUCSSPortalADFLibrary/public\_html/oracle/ugbu/ss/custom/pages/secure and/or <<ExtensionWorkSpace>>/OUCSSPortalADFLibrary/public\_html/oracle/ugbu/ss/custom/pages/public folders.

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 users (e.g, 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 under \oracle\ugbu\ss\custom\pages\secure if the pages should be accessible only to authenticated users (e.g, logged-in users).
- Create pages 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.

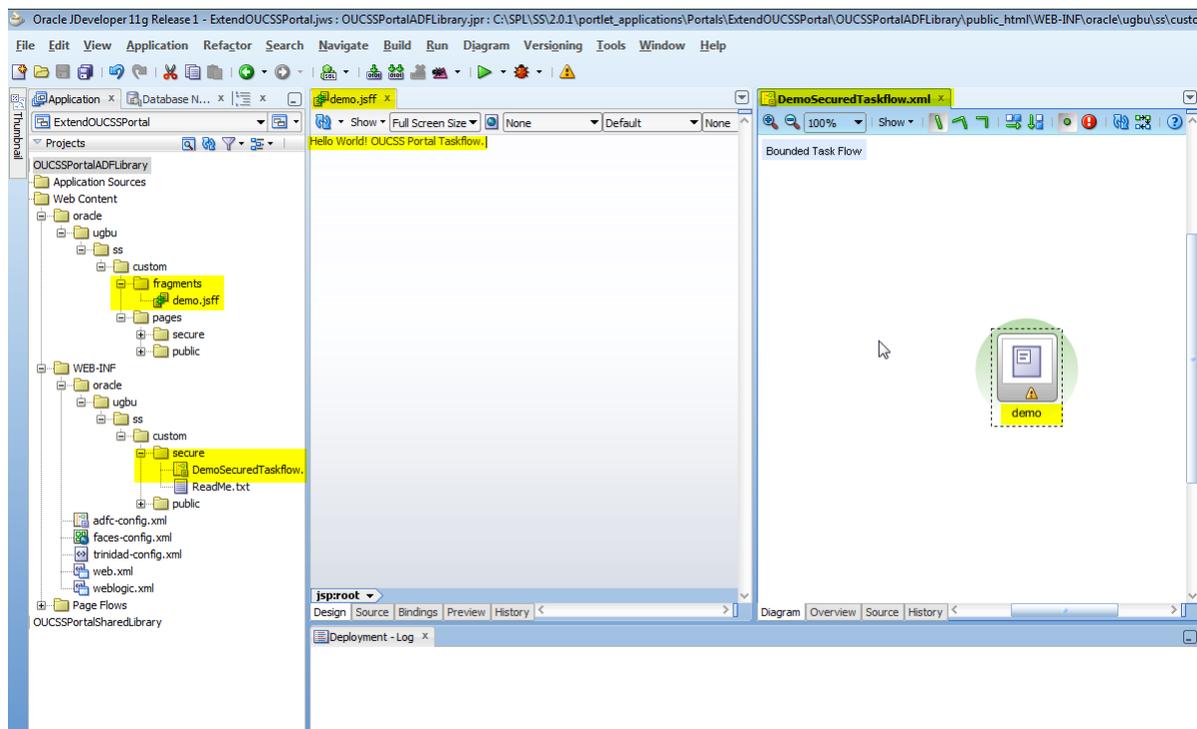
## Adding New Taskflows and Pages

- 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.



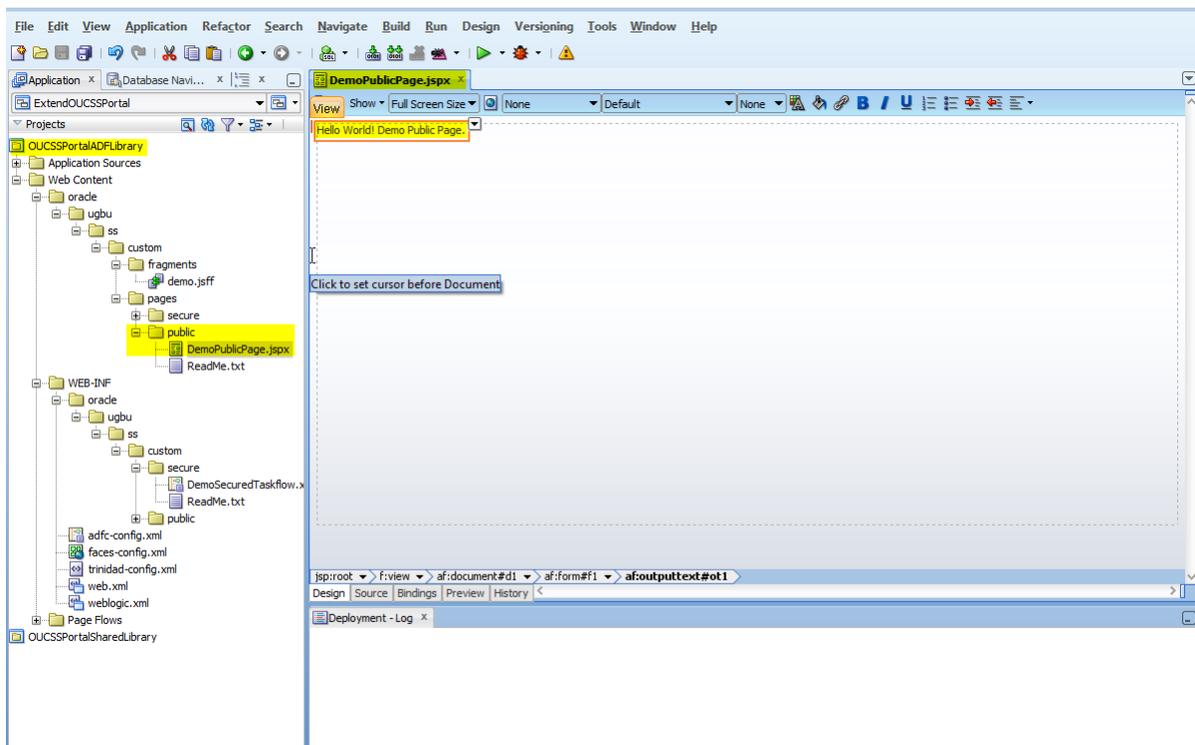
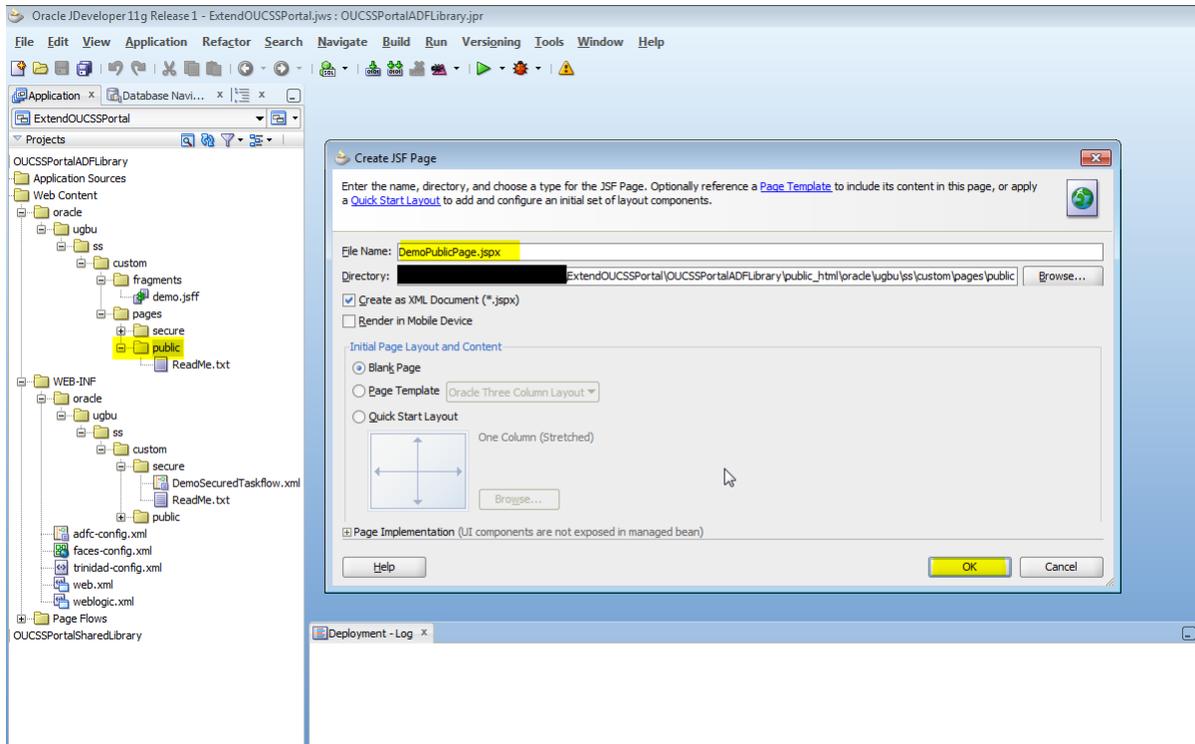
**3** Implement the taskflow per the security requirements.

For the purposes of this demo, we'll create a secure taskflow containing one view that displays, "Hello World! OUCSS Portal Taskflow".



- 4 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.jsx is created that contains the text, “Hello World! Demo Public Page.”



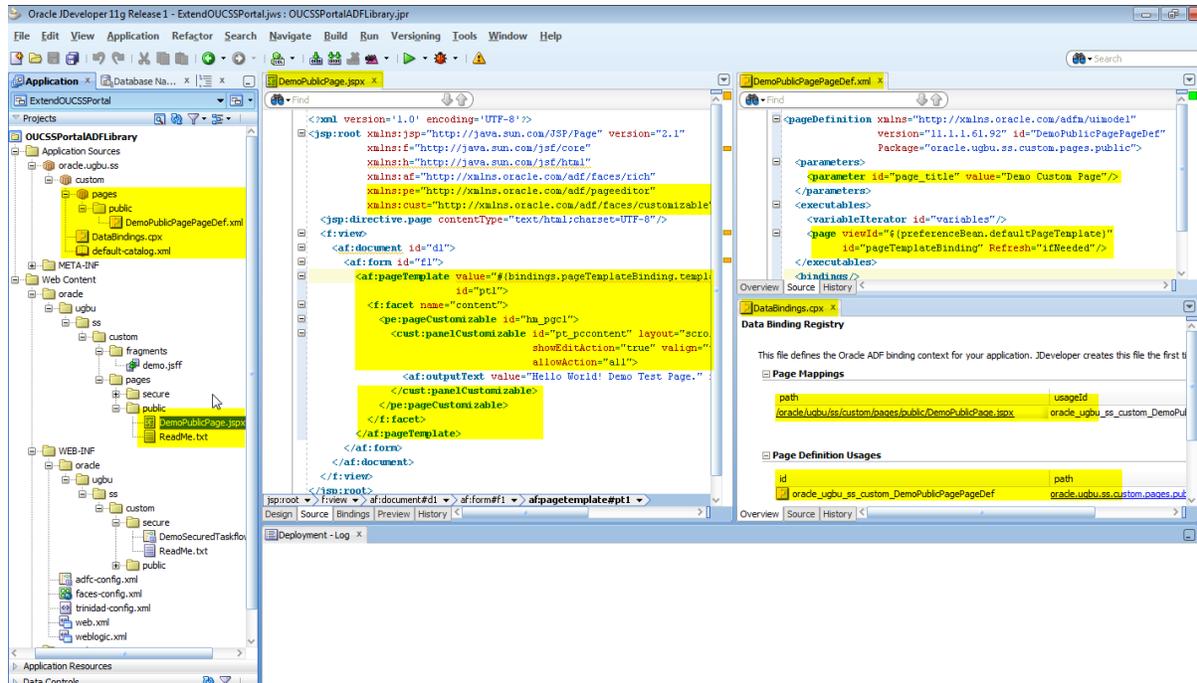
- 5 The newly-created page should reference a page template and page editor to enable runtime customization. To create the references:

A. Modify JSPX to:

```
<?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_pgcl">
              <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>
```

B. Modify Page Def to:

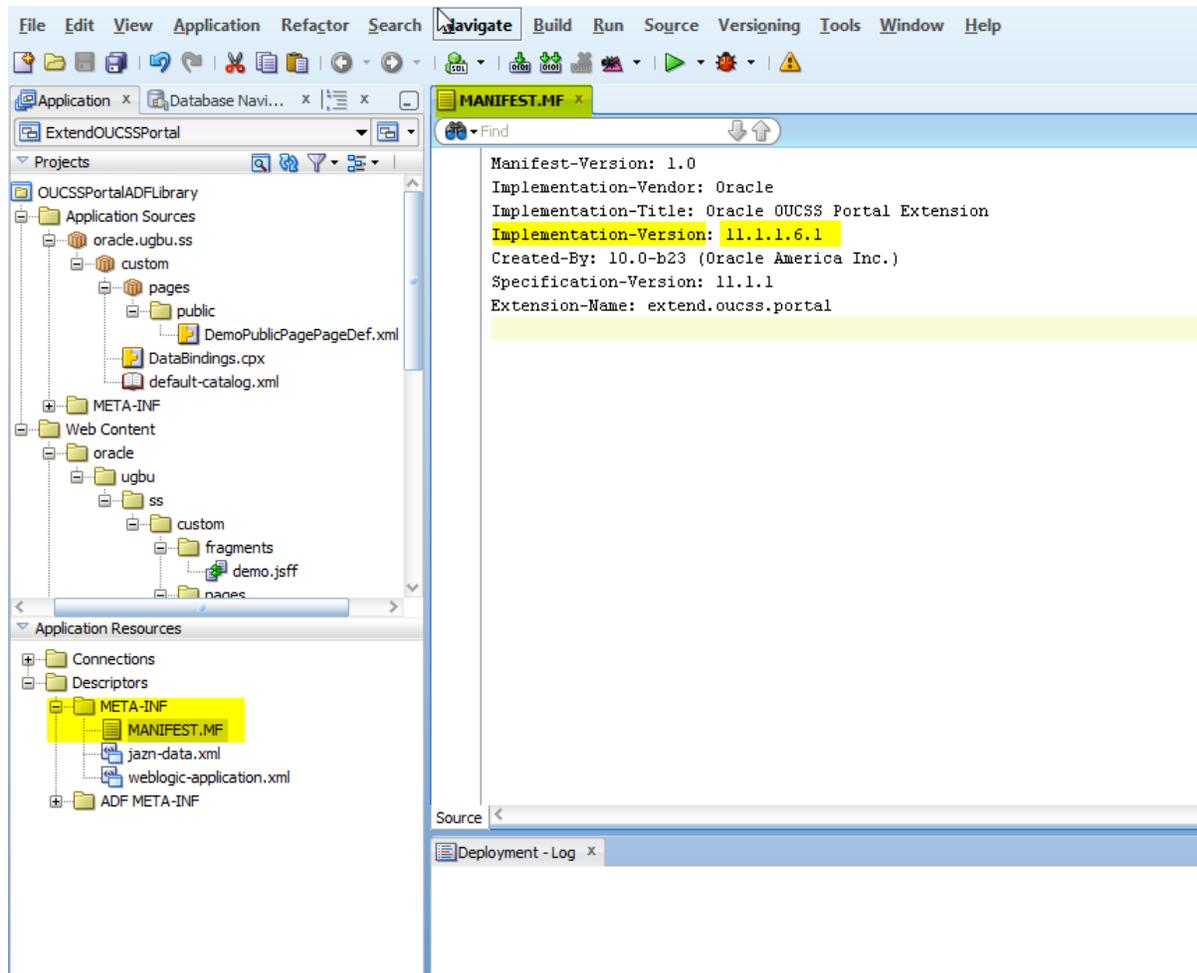
```
<?xml version="1.0" encoding="UTF-8" ?>
<pageDefinition xmlns="http://xmlns.oracle.com/adfm/uimodel"
  version="11.1.1.61.92" id="DemoPublicPagePageDef"
  Package="oracle.ugbu.ss.custom.pages.public"> <parameters>
  <parameter id="page_title" value="Demo Custom Page"/>
</parameters>
<executables>
  <variableIterator id="variables"/>
  <page viewId="{preferenceBean.defaultPageTemplate}"
    id="pageTemplateBinding" Refresh="ifNeeded"/>
</executables>
<bindings/>
</pageDefinition>
```



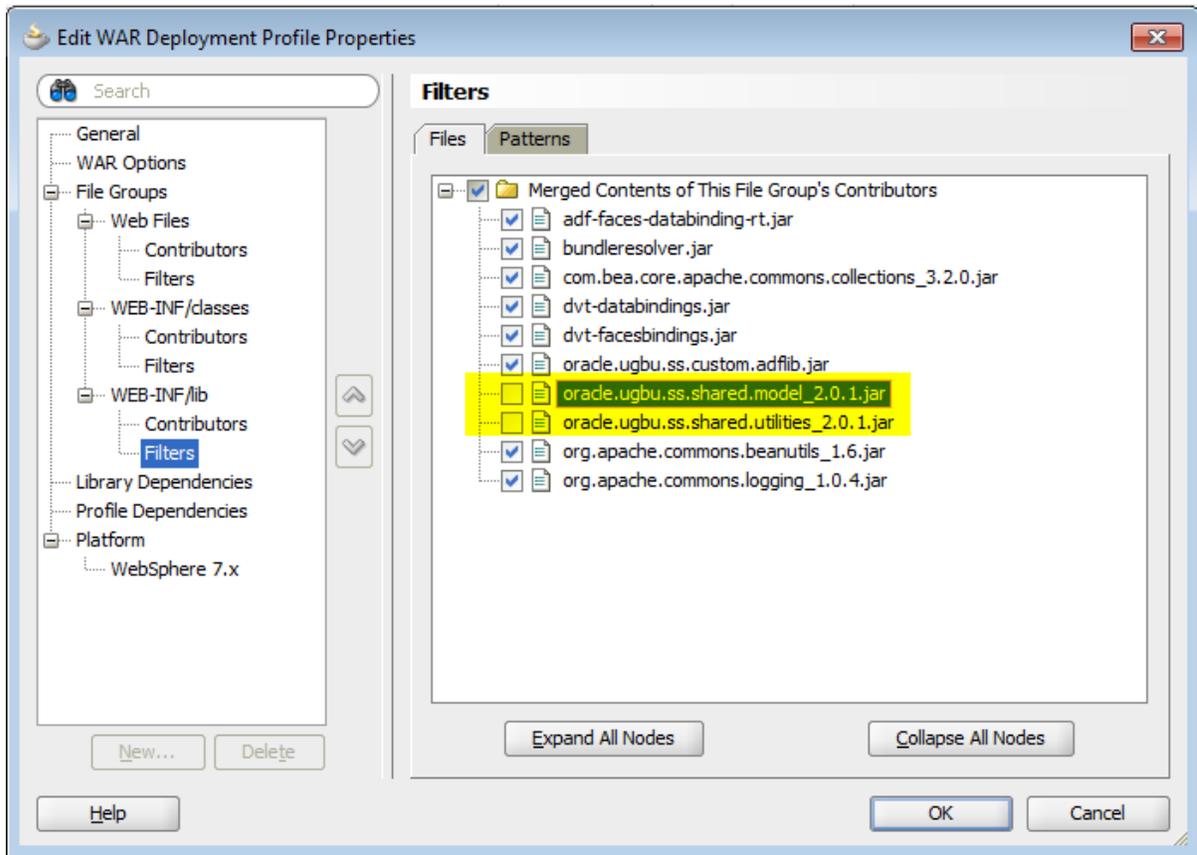
## Deploying the extend.oucss.portal Shared Library

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

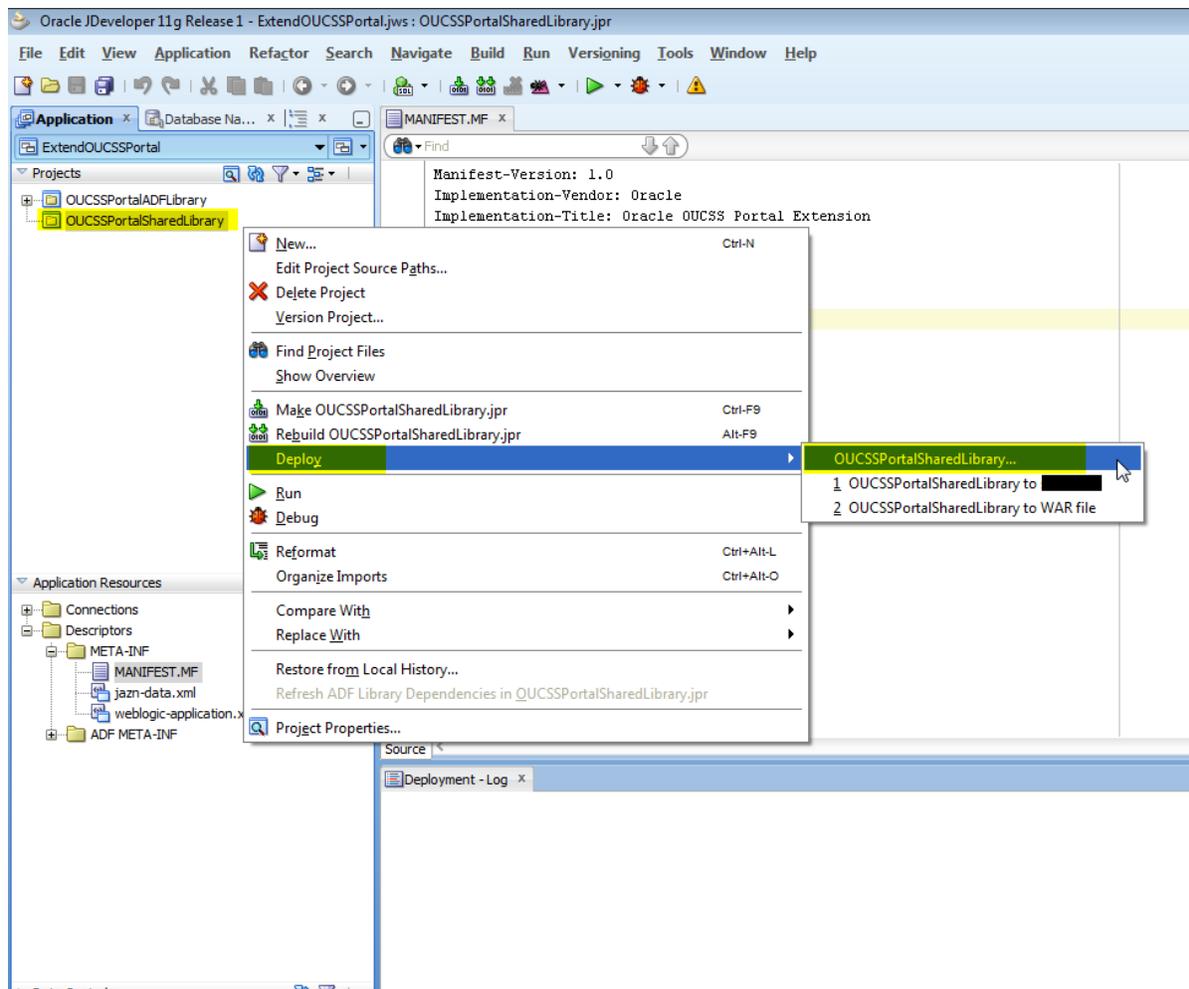
The default implementation version number is 11.1.1.6.0. You can increment the major or minor version. For the purposes of this demo, the new implementation version number will be incremented to 11.1.1.6.1.



- 2 Edit the OUCSSPortalSharedLibrary deployment profile and uncheck references to jars from the OUCSS Shared library, as shown in the following image.



- 3 Deploy the OUCSS Portal Extension Shared Library to consume the taskflow in the portal. To deploy, right-click on the **OUCSSPortalSharedLibrary** project and select the **OUCSSPortalSharedLibrary** deployment profile to redeploy the extension to the same managed server as the OUCSS Portal application.



4 Check **Deployments** in the WebLogic Console to verify that the deployment was successful.

	extend.oucss.portal(11.1.1,11.1.1.6.0)	Active		Library	100
	extend.oucss.portal(11.1.1,11.1.1.6.1)	Active		Library	100

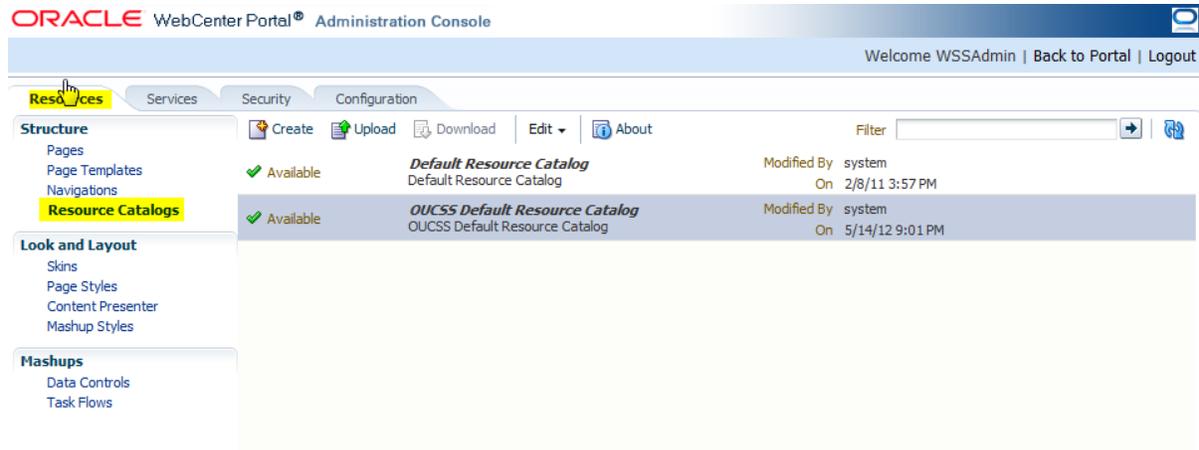
5 The custom managed server must be restarted for the newly-created taskflows and pages to be consumed in the OUCSS Portal.

## Customizing and Configuring an Existing Resource Catalog

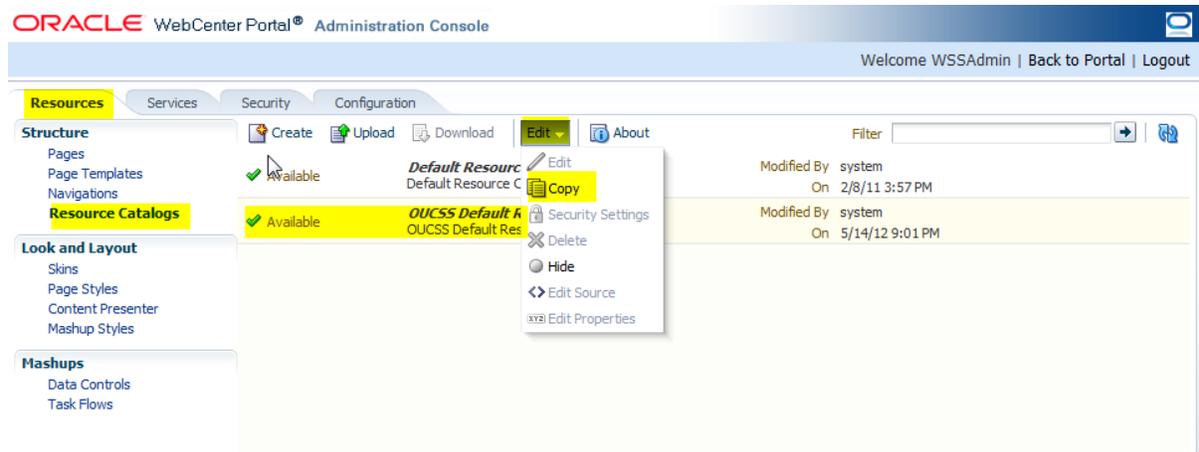
An OUCSS Portal application is deployed with a default OUCSS Default Resource Catalog.

To consume the newly-created taskflow, a customized resource catalog must be created.

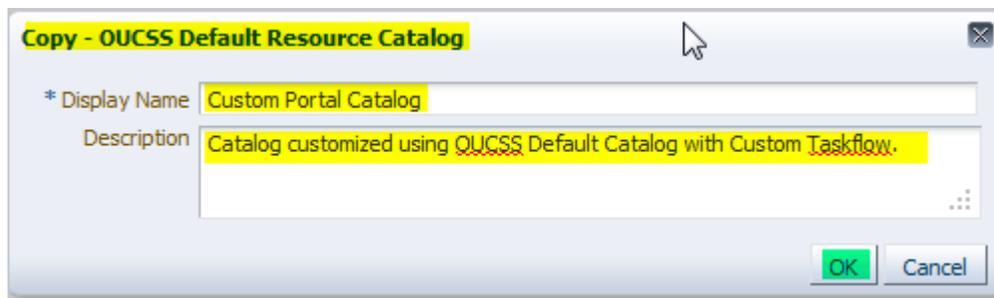
- 1 Log in to the OUCSS Portal with Administrator privileges and go to the Administration Console.
- 2 Choose the **Resources** tab and click on the **Resource Catalogs** link in the **Structure** section.



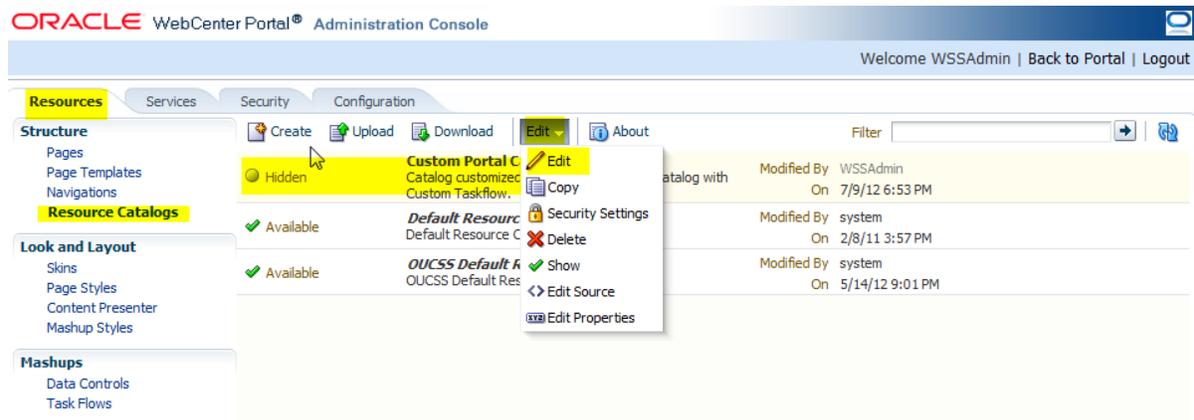
- To copy the default catalog (which is read-only and seeded), choose **OUCSS Default Resource Catalog** and click **Edit > Copy**.



- In the **Copy** dialog, enter the display name and description for the new catalog, then click OK.

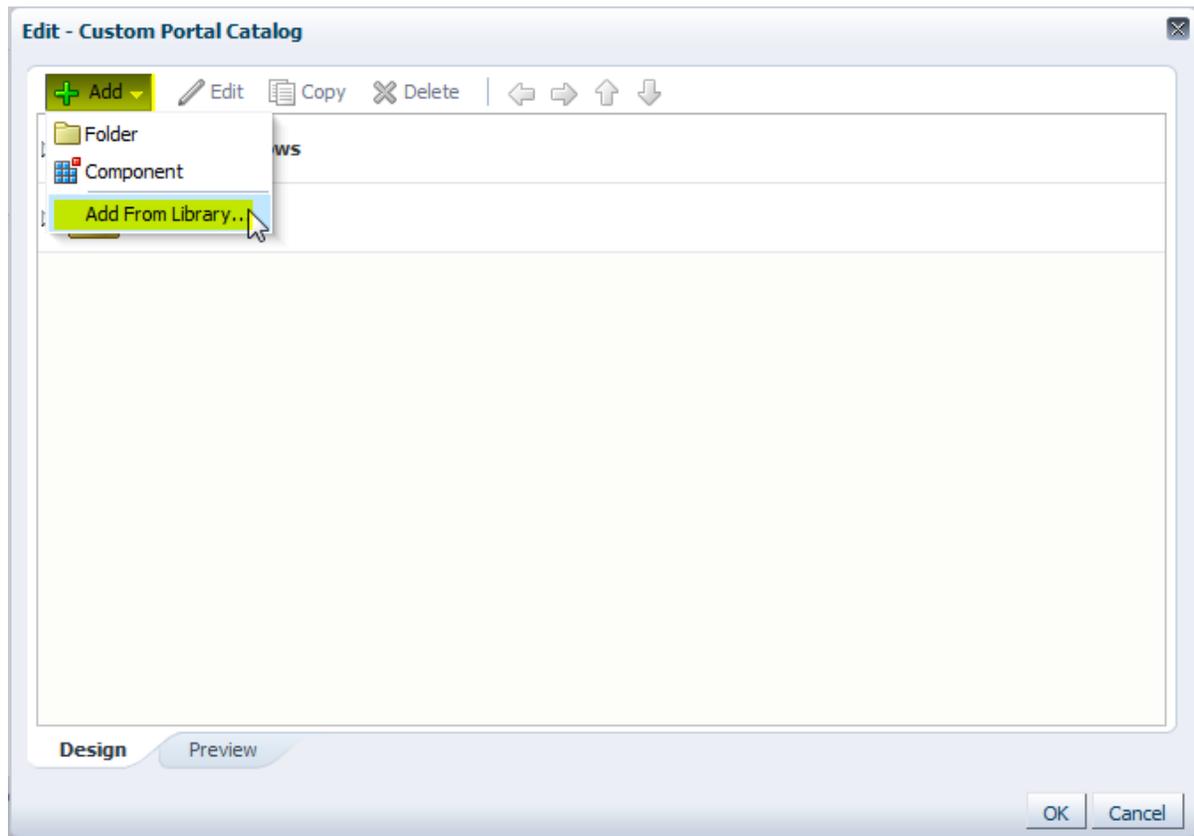


- 5 Select the new catalog from the list and click **Edit** > **Edit**.



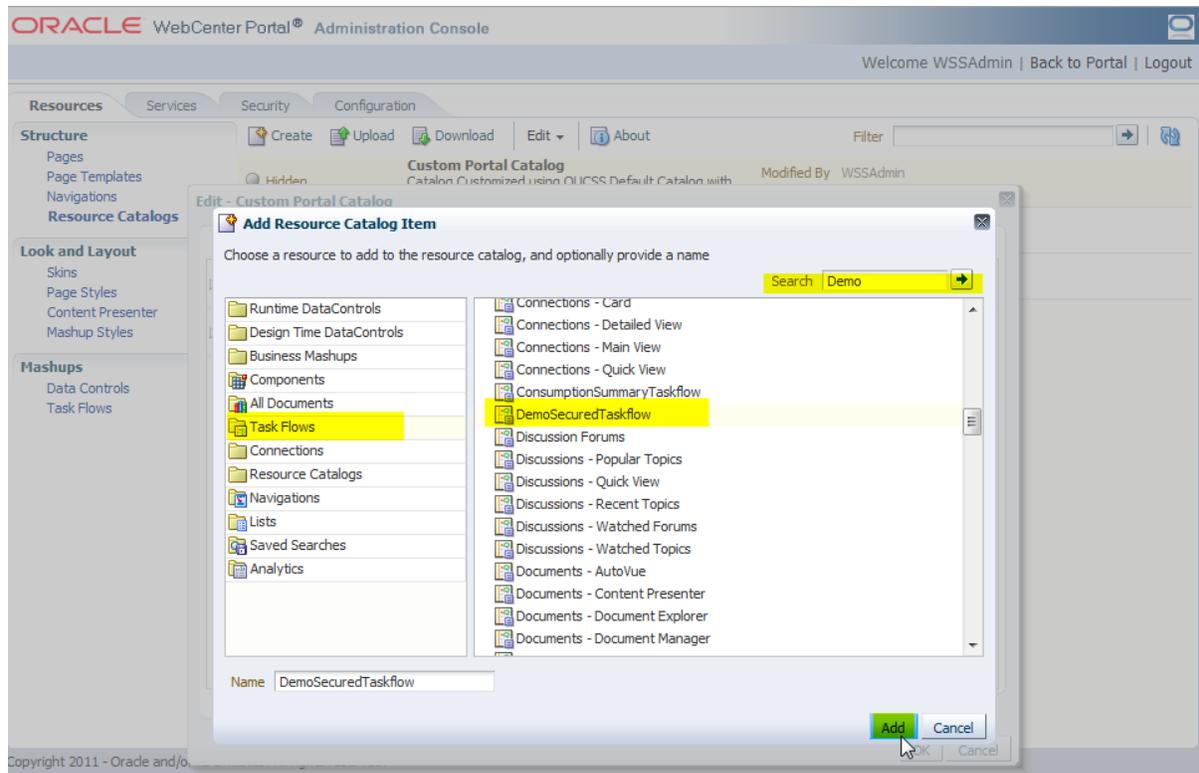
- 6 In the **Edit** dialog, click **Add** > **Add From Library...** to add the new taskflow to the catalog.

**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.

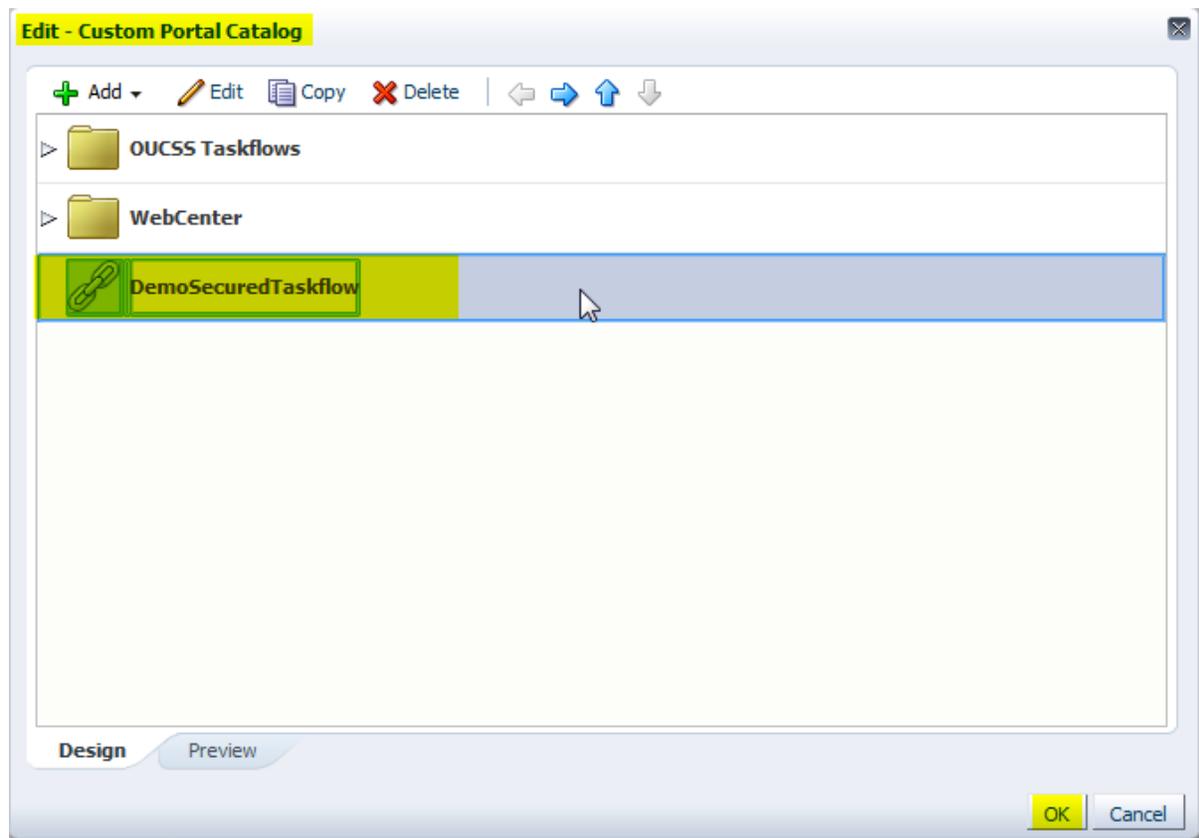


- 7 In the **Add Resource Catalog Item** dialog, select **Taskflows** from the menu on the left, then scroll down and choose the new taskflow (in this demo, the **DemoSecuredTaskflow** created earlier). Note that you can also search for the taskflow using the Search field in the dialog.

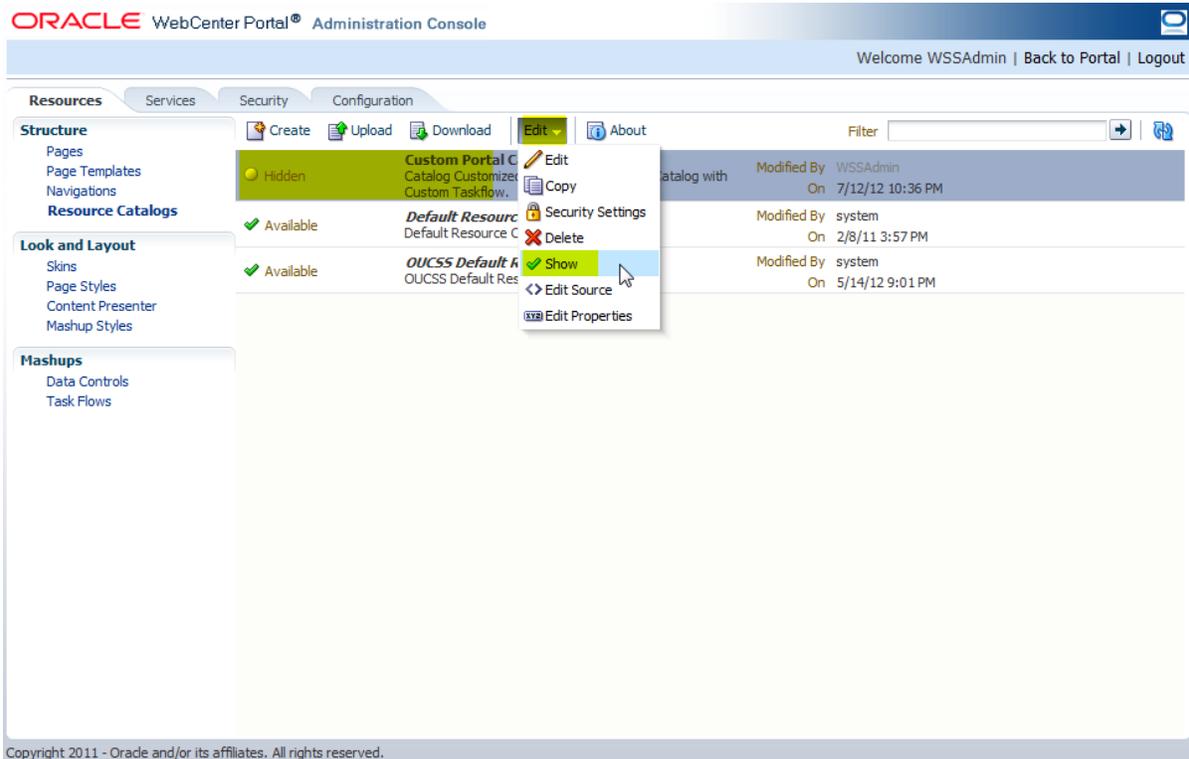
When the taskflow is selected, click the **Add** button.



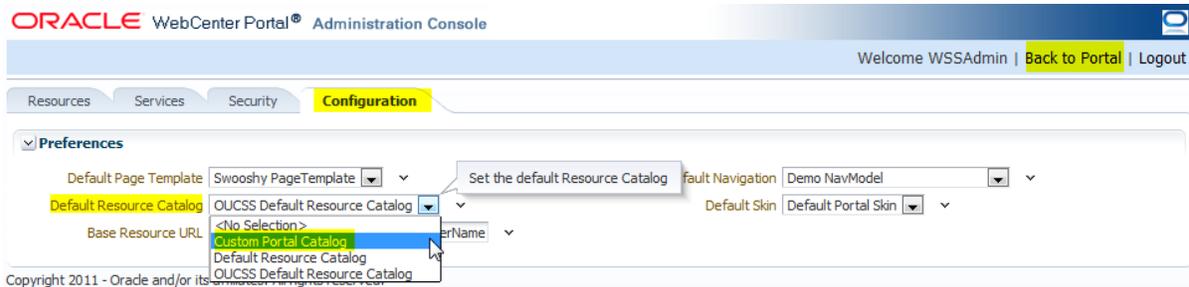
8 The new taskflow now appears in the catalog list. Click **OK** to save the changes.



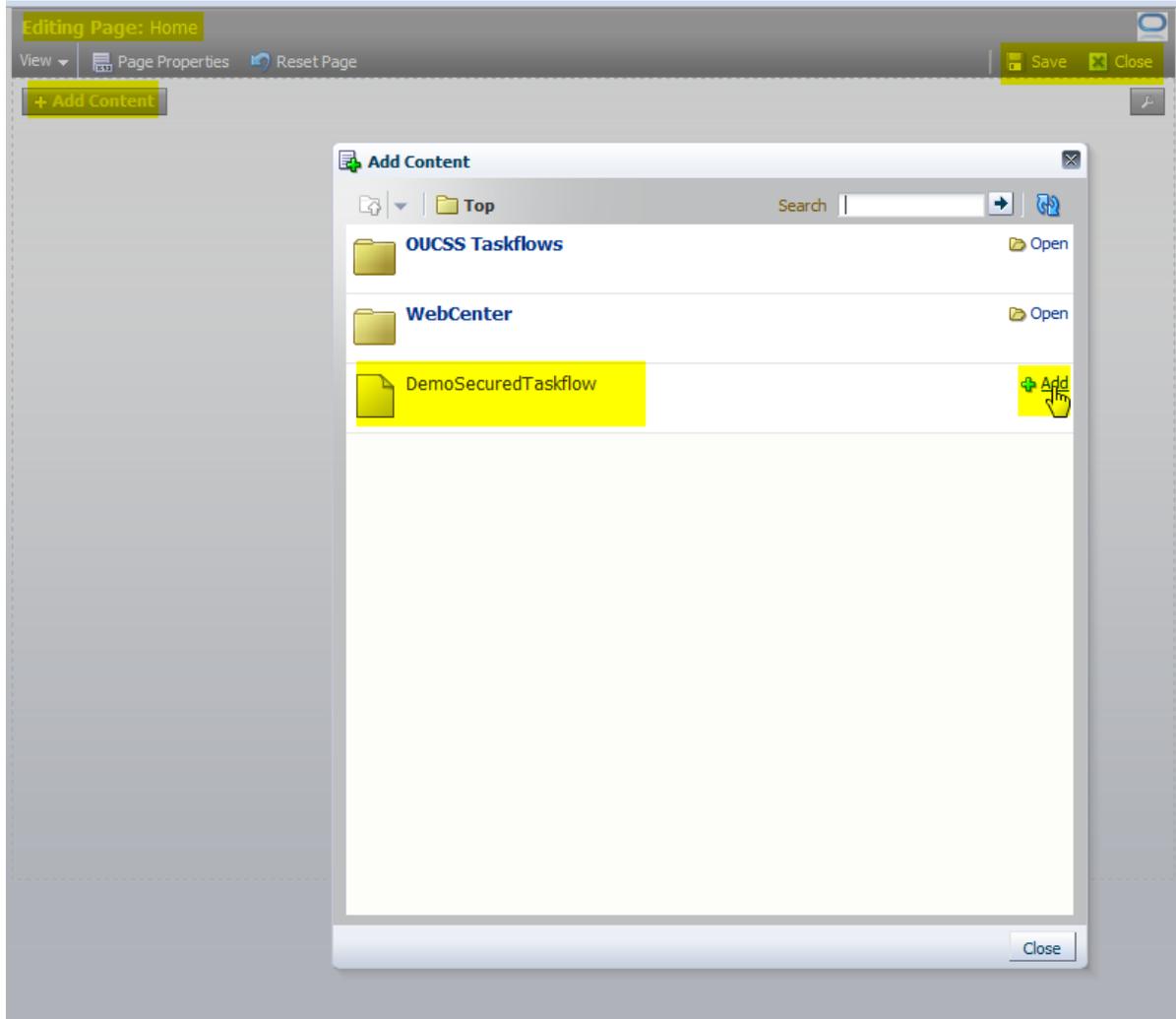
9 Select the new catalog (**Custom Portal Catalog** in this demo) and unhide it by clicking on **Edit > Show**.



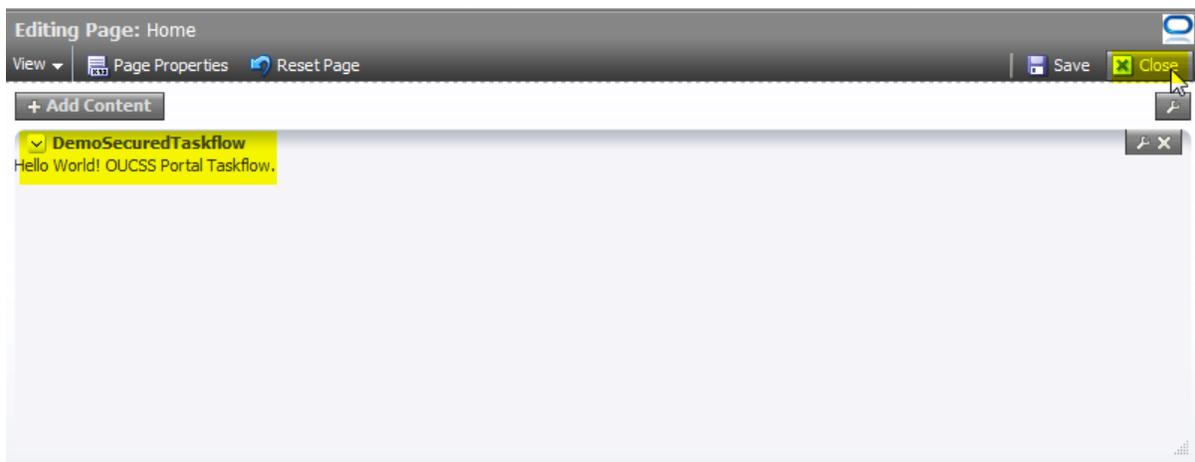
10 Set the new customized resource catalog as the new default by selecting it in the **Default Resource Catalog** dropdown list in the **Preferences** section of the **Configuration** tab .

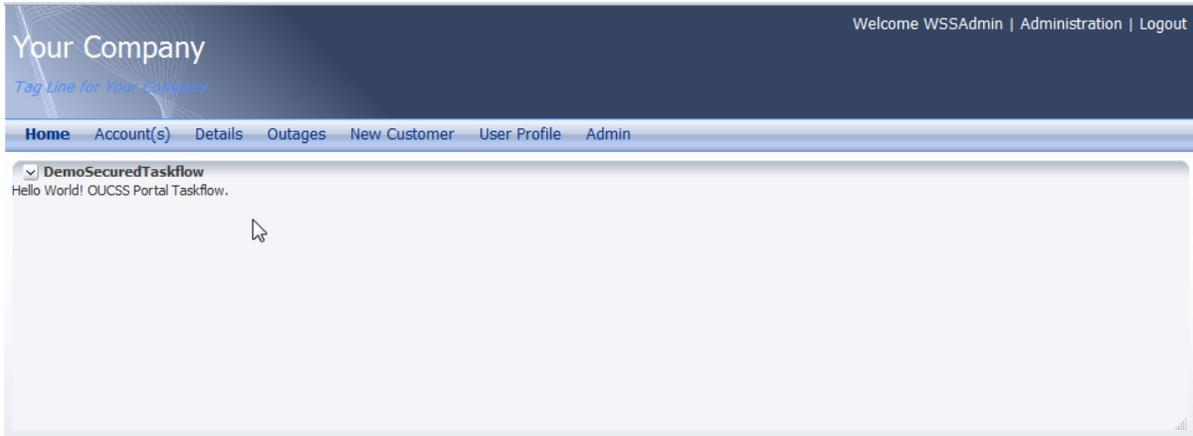


- 11 Click **Back to Portal**. In the **Add Content** dialog, choose the custom taskflow, click **Add**, then click the **Close** button.



- 12 Click the **Close** button in the **Editing Page** view to commit and view the changes.

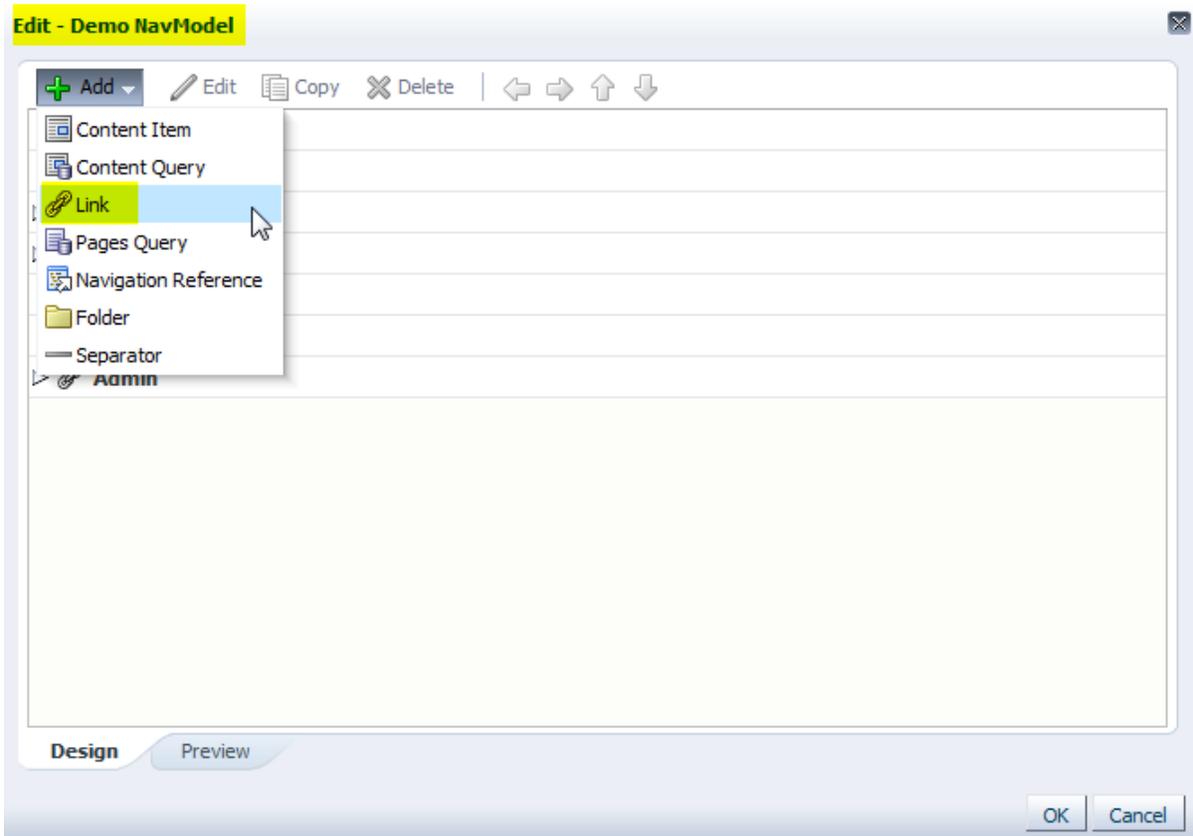




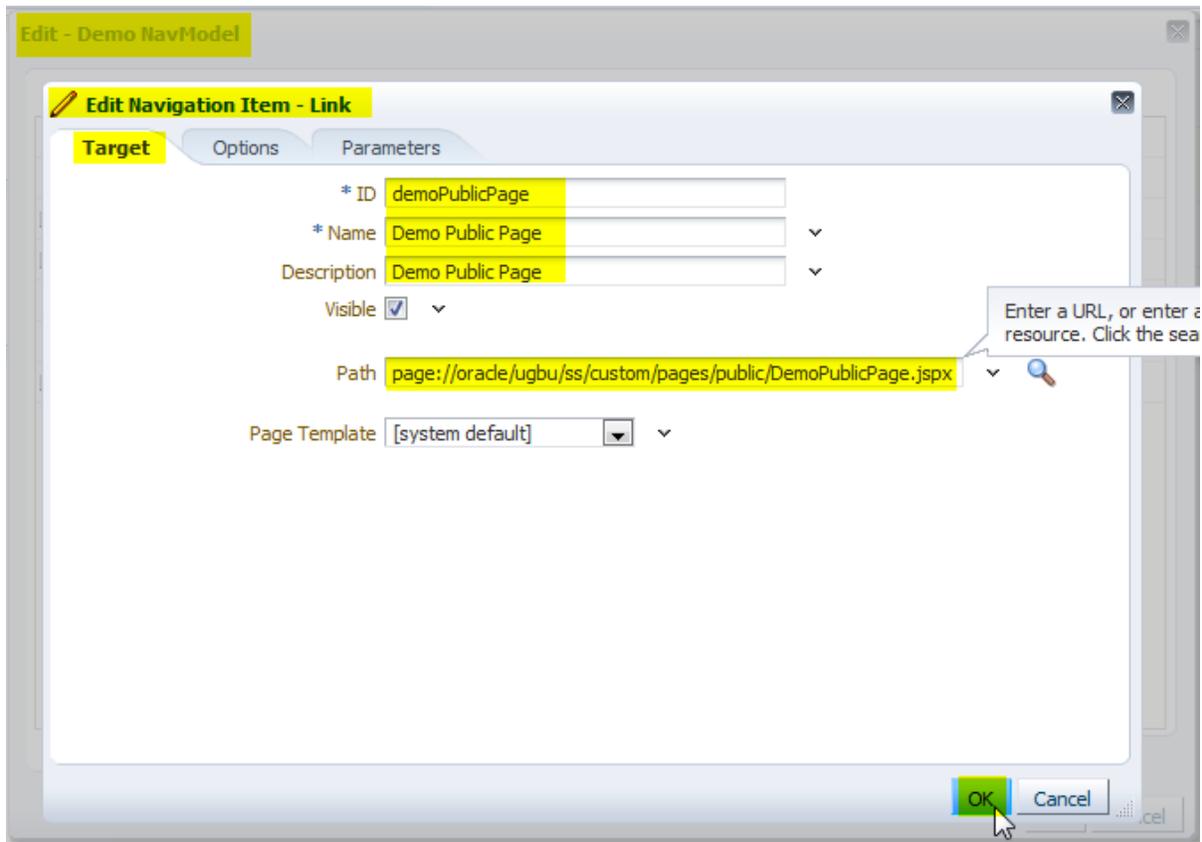
### Accessing Custom Pages

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

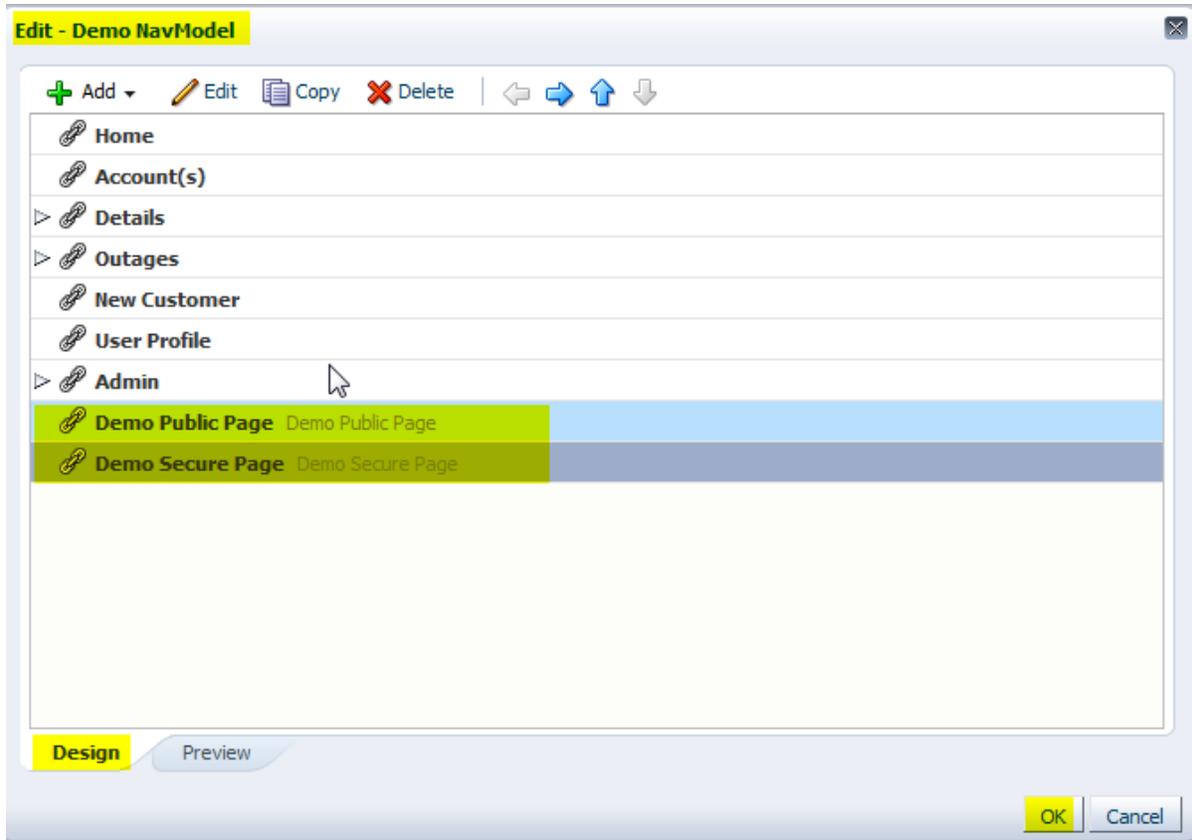
- 1 Log in to the OUCSS Portal with Administrator privileges and go to the Administration Console.
- 2 Choose the **Resources** tab and click on the **Navigations** link in the **Structure** section.
- 3 Select a navigation model for the new links, then click **Edit > Edit**.  
For the purposes of this demo, Demo NavModel will be used for the new links.
- 4 In the **Edit** dialog, click **Add > Link**.



- 5 In the **Edit Navigation Item** dialog, choose the **Target** tab, enter link details, then click **OK**.



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



7 Verify that the navigation model is configured as the **Default**, then click **Back to Portal**.



8 Log out of the portal.

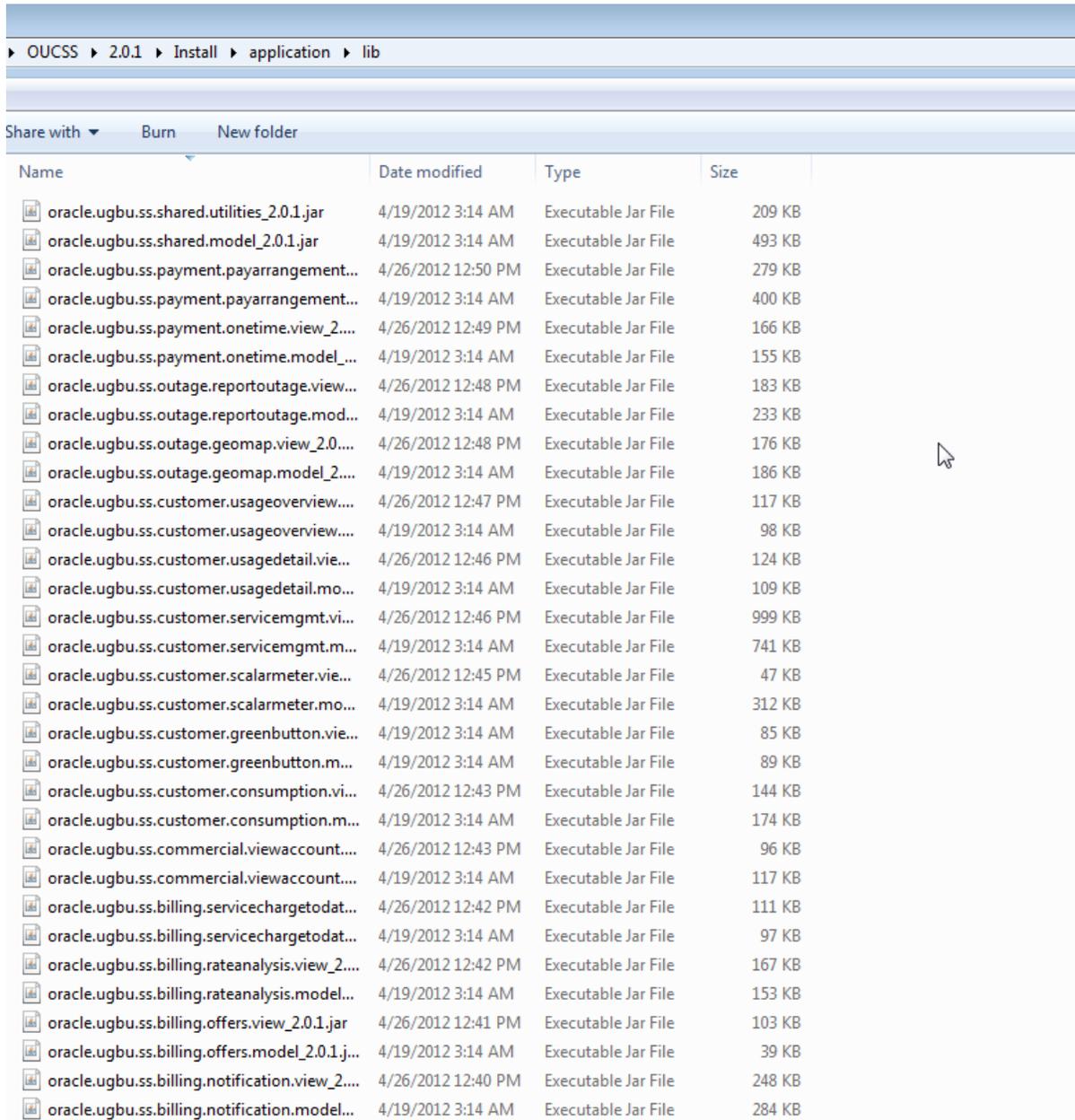


## Accessing User Account Details

Account details and other OUCSS API can be accessed in custom pages and taskflow views by referring to the Utilities Shared jar from the OUCSS Shared module.

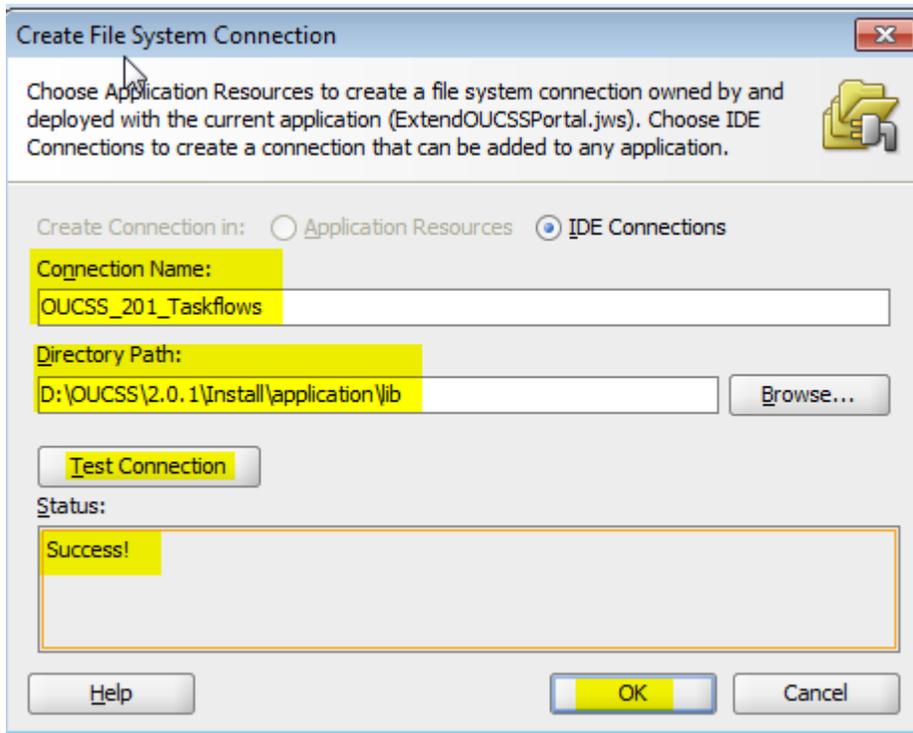
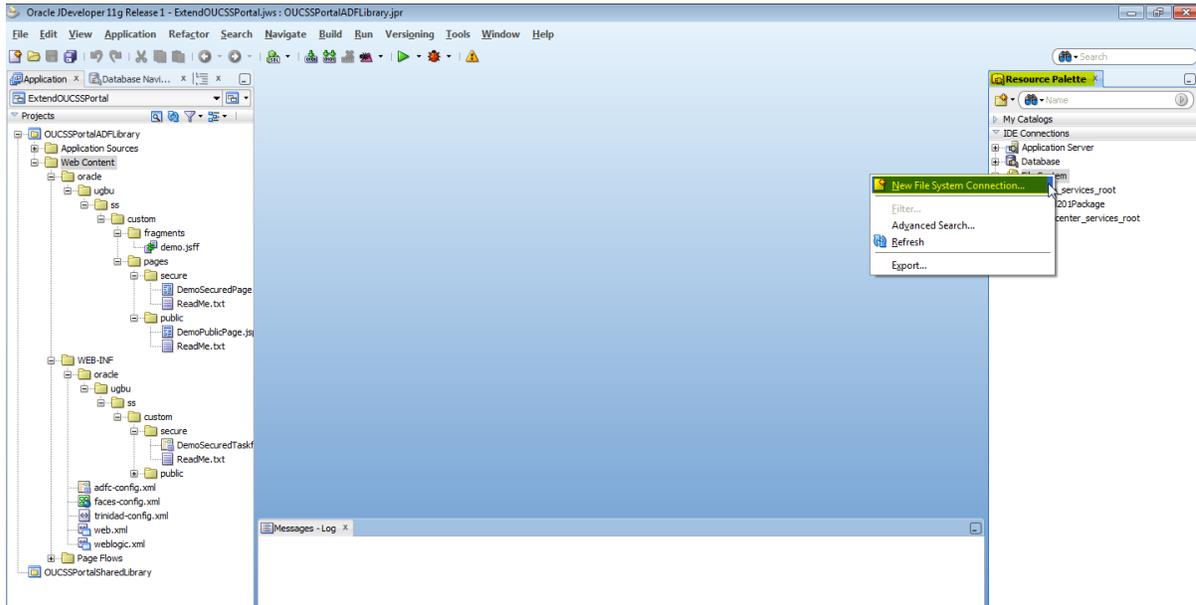
In this example, a managed bean is created and accessed using the Secured Taskflow. The managed bean will retrieve accounts of a logged-in user and display the list in the taskflow.

- 1 Unjar/unzip the OUCSS\_Extension.war file from <<webcenter\_WSS\_zip>>\Install\application\ into a folder that is accessible to JDeveloper.

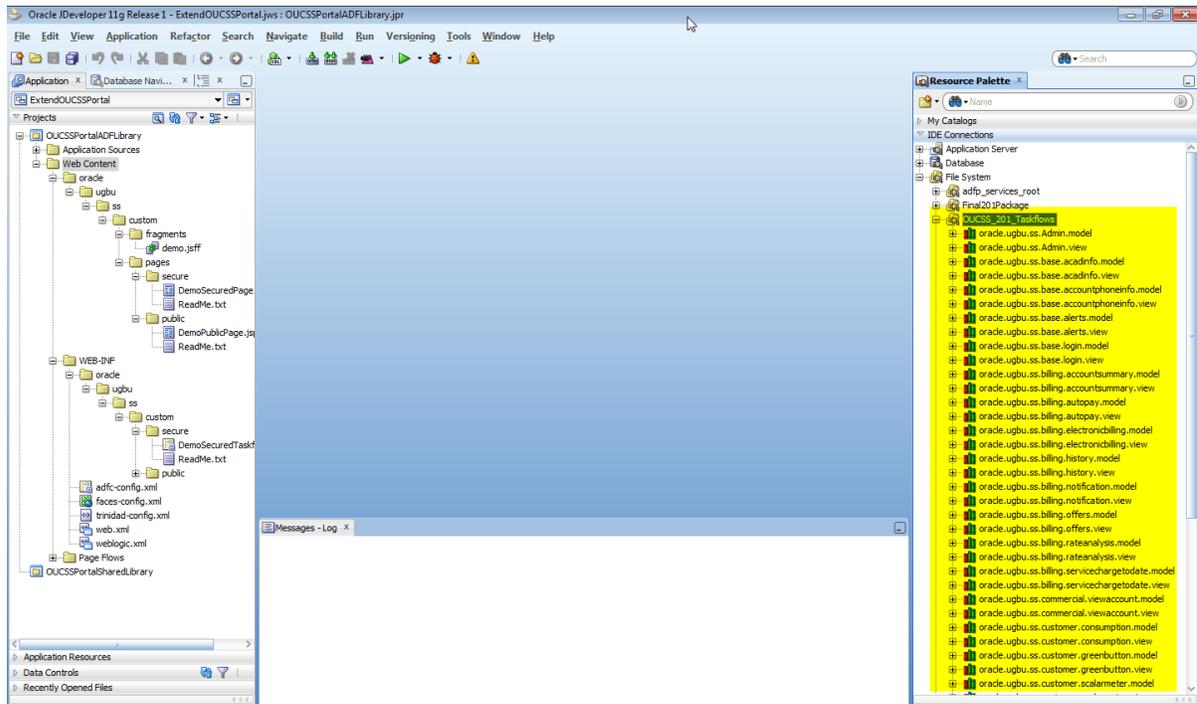


Name	Date modified	Type	Size
oracle.ugbu.ss.shared.utilities_2.0.1.jar	4/19/2012 3:14 AM	Executable Jar File	209 KB
oracle.ugbu.ss.shared.model_2.0.1.jar	4/19/2012 3:14 AM	Executable Jar File	493 KB
oracle.ugbu.ss.payment.payarrangement...	4/26/2012 12:50 PM	Executable Jar File	279 KB
oracle.ugbu.ss.payment.payarrangement...	4/19/2012 3:14 AM	Executable Jar File	400 KB
oracle.ugbu.ss.payment.onetime.view_2....	4/26/2012 12:49 PM	Executable Jar File	166 KB
oracle.ugbu.ss.payment.onetime.model_...	4/19/2012 3:14 AM	Executable Jar File	155 KB
oracle.ugbu.ss.outage.reportoutage.view...	4/26/2012 12:48 PM	Executable Jar File	183 KB
oracle.ugbu.ss.outage.reportoutage.mod...	4/19/2012 3:14 AM	Executable Jar File	233 KB
oracle.ugbu.ss.outage.geomap.view_2.0....	4/26/2012 12:48 PM	Executable Jar File	176 KB
oracle.ugbu.ss.outage.geomap.model_2....	4/19/2012 3:14 AM	Executable Jar File	186 KB
oracle.ugbu.ss.customer.usageoverview....	4/26/2012 12:47 PM	Executable Jar File	117 KB
oracle.ugbu.ss.customer.usageoverview....	4/19/2012 3:14 AM	Executable Jar File	98 KB
oracle.ugbu.ss.customer.usagedetail.vie...	4/26/2012 12:46 PM	Executable Jar File	124 KB
oracle.ugbu.ss.customer.usagedetail.mo...	4/19/2012 3:14 AM	Executable Jar File	109 KB
oracle.ugbu.ss.customer.servicemgmt.vi...	4/26/2012 12:46 PM	Executable Jar File	999 KB
oracle.ugbu.ss.customer.servicemgmt.m...	4/19/2012 3:14 AM	Executable Jar File	741 KB
oracle.ugbu.ss.customer.scalarmeter.vie...	4/26/2012 12:45 PM	Executable Jar File	47 KB
oracle.ugbu.ss.customer.scalarmeter.mo...	4/19/2012 3:14 AM	Executable Jar File	312 KB
oracle.ugbu.ss.customer.greenbutton.vie...	4/19/2012 3:14 AM	Executable Jar File	85 KB
oracle.ugbu.ss.customer.greenbutton.m...	4/19/2012 3:14 AM	Executable Jar File	89 KB
oracle.ugbu.ss.customer.consumption.vi...	4/26/2012 12:43 PM	Executable Jar File	144 KB
oracle.ugbu.ss.customer.consumption.m...	4/19/2012 3:14 AM	Executable Jar File	174 KB
oracle.ugbu.ss.commercial.viewaccount....	4/26/2012 12:43 PM	Executable Jar File	96 KB
oracle.ugbu.ss.commercial.viewaccount....	4/19/2012 3:14 AM	Executable Jar File	117 KB
oracle.ugbu.ss.billing.servicechargetodat...	4/26/2012 12:42 PM	Executable Jar File	111 KB
oracle.ugbu.ss.billing.servicechargetodat...	4/19/2012 3:14 AM	Executable Jar File	97 KB
oracle.ugbu.ss.billing.rateanalysis.view_2....	4/26/2012 12:42 PM	Executable Jar File	167 KB
oracle.ugbu.ss.billing.rateanalysis.model...	4/19/2012 3:14 AM	Executable Jar File	153 KB
oracle.ugbu.ss.billing.offers.view_2.0.1.jar	4/26/2012 12:41 PM	Executable Jar File	103 KB
oracle.ugbu.ss.billing.offers.model_2.0.1.j...	4/19/2012 3:14 AM	Executable Jar File	39 KB
oracle.ugbu.ss.billing.notification.view_2....	4/26/2012 12:40 PM	Executable Jar File	248 KB
oracle.ugbu.ss.billing.notification.model...	4/19/2012 3:14 AM	Executable Jar File	284 KB

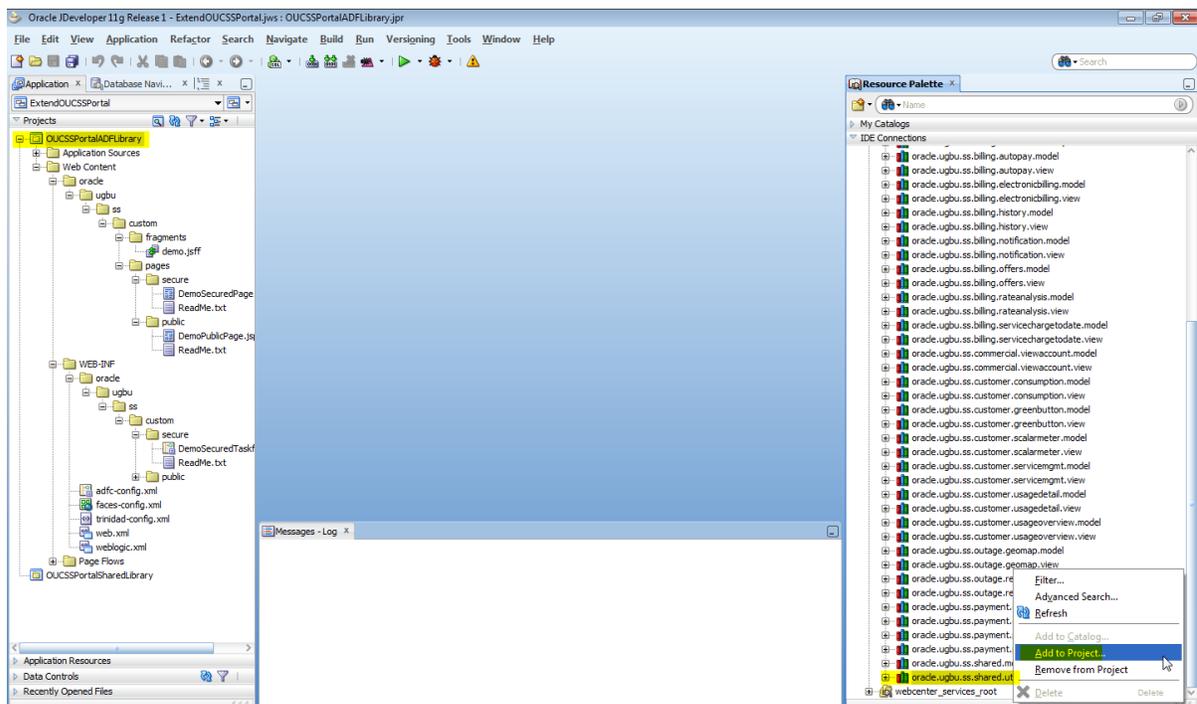
- Using the JDeveloper Resource Palette, create a File System Connection with the name “OUCSS\_Taskflows” using the <<Unzip\_Folder>>\WEB-INF\lib directory created in the previous step.



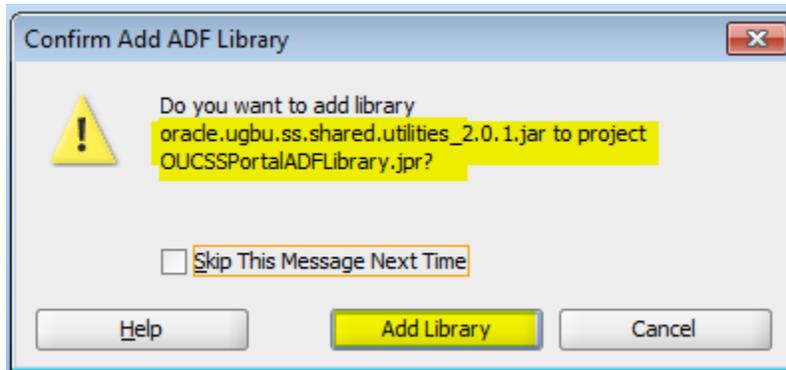
**3** Click **OK** to see the OUCSS libraries under OUCSS\_201\_Taskflows in the Resource Palette.



**4** Make sure the OUCSSPortalADFLibrary project is selected in JDeveloper. Right-click **oracle.ugbu.ss.shared.utilities** under the OUCSS\_201\_Taskflows file system connection, then select **Add to Project** from the context menu.



If a confirmation dialog appears, click **Add Library**:



- 5 Save all.
- 6 Create a Java class (DemoManagedBean.java, sample code below) to access the account details in the task flow.

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

import com.oracle.ugbu.ss.common.SSContext;
import com.oracle.ugbu.ss.security.Account;
import com.oracle.ugbu.ss.security.User;

import java.util.ArrayList;
import java.util.Collections;
import java.util.Iterator;
import java.util.List;
import java.util.Map;
import java.util.Set;

import oracle.adf.share.ADFContext;

public class DemoManagedBean
{
    private String currentUser =
ADFContext.getCurrent().getSecurityContext().getUserName();

    public User getUser()
    {
        SSContext ssContext = new SSContext(currentUser);
        return ssContext.getCurrentUser();
    }

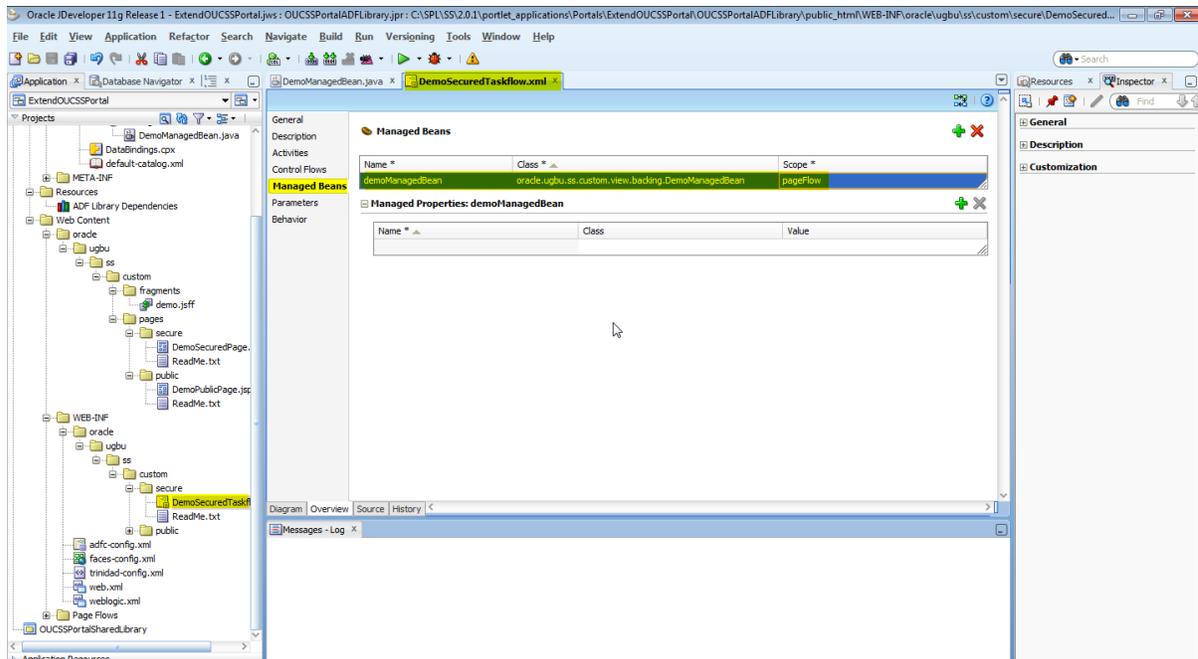
    public List<Account> getUserAccountList()
    {
        List<Account> accountList = null;
        SSContext ssContext = new SSContext(currentUser);
```

```

Map accountsMap = ssContext.getCurrentUser().getUserAccounts();
if (accountsMap == null && accountsMap.isEmpty())
{
    accountList = Collections.emptyList();
}
else
{
    //accountList = (List<Account>) accountsMap.values();

System.out.println(accountsMap.values().getClass().getCanonicalName());
Iterator keys = accountsMap.keySet().iterator();
accountList = new ArrayList(accountsMap.size());
while (keys.hasNext())
{
    accountList.add((Account) accountsMap.get(keys.next()));
}
}
return accountList;
}
}
    
```

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



- 8 Reference the managed bean method to display user and account details as demonstrated in the following 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"
  xmlns:h="http://java.sun.com/jsf/html">
  <af:panelBorderLayout>
    <f:facet name="top">
      <af:outputText value="Hello World! OUCSS Portal Taskflow." id="ot1"/>
    </f:facet>
    <af:panelGroupLayout layout="vertical" halign="start">
      <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.email}"
        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.keyFieldValues['ACCT_ID']}"
id="ot2"/>
        </af:column>
        <af:column headerText="Person Id" id="c2">
          <af:outputText value="#{row.keyFieldValues['PER_ID']}" id="ot3"/>
        </af:column>
      </af:table>
    </af:panelGroupLayout>
  </af:panelBorderLayout>
</jsp:root>
```

- 9 Deploy the extension again and consume the taskflow in any page. In this example, the taskflow is consumed in the Home Page, as shown in the following image:

