# Oracle Utilities
# Customer Self Service

Whitepaper:

OUCSS Web Application

Release 2.1.0 Service Pack 1

**E50365-02**

May 2014

ORACLE

Oracle Utilities Customer Self Service Whitepaper: OUCSS Web Application

Release 2.1.0 Service Pack 1

E50365-02

May 2014

Copyright © 2011, 2014, 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.

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, 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 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** ..... **5**

Additional Resources ..... 5

Software Requirements ..... 5

Abbreviations ..... 6

**Installation and Uninstallation** ..... **7**

Installing on a WebLogic Server ..... 7

Prerequisites ..... 7

Installation Steps ..... 7

UnInstalling the Web Application ..... 9

**Extending the OUCSS Web Application Sample** ..... **11**

**Setting up JDeveloper for Development** ..... **12**

Adding Shared Libraries to the Project ..... 12

Creating a File System Connection to the Taskflows Directory ..... 13

**Building an ADF Web Application** ..... **15**

Managed Beans ..... 15

Look and Feel: Page Templates and Skins ..... 16

Applying Skins ..... 18

Creating *jspx* Pages ..... 21

Creating Custom Taskflows ..... 24

Consuming OUCSS Taskflows ..... 24

Navigation: Creating Menus and Page Flows ..... 27

Security ..... 32

Testing and Deployment ..... 33

Testing the Integrated WebLogic Server in JDeveloper ..... 34

Pre-testing Checklist ..... 34

Running the Test ..... 36
Deploying to a Standalone WebLogic Server Using JDeveloper ..... 37
Deploying to a Standalone WebLogic Server Through WebLogic Enterprise Manager (EM) ..... 41

# Chapter 1

# Overview

This whitepaper describes the reference Oracle Utilities Customer Self Service (OUCSS) Web Application that is developed using the Oracle Application Development Framework (ADF) and is pre-integrated with the core OUCSS application modules and security model. It provides flexible deployment options and extension methodology to adapt the reference application to specific implementation requirements.

## Additional Resources

For additional details see the following resources:

- The *Oracle Utilities Customer Self Service Installation Guide* and the *Oracle Utilities Customer Self Service Implementation Guide* provide information on product installation/uninstallation, requirements, configuration, and administration.
- *Oracle® Fusion Middleware Fusion Developer's Guide for Oracle Application Development Framework 11g Release 1 (11.1.1.7.0). (*http://docs.oracle.com/cd/E28280_01/web.1111/b31974/toc.htm*).*

**Note**: This document and the documentation mentioned above is subject to revision and updating. For the most recent version of this and related documentation, 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 Documentation area on the Oracle Technology Network (OTN) web site (http://www.oracle.com/technetwork/apps-tech/utilities/documentation/index.html).

## Software Requirements

- Oracle JDeveloper version is 11gR1 PS6 (11.1.1.7.0). Download JDeveloper 11.1.1.7 from OTN at http://www.oracle.com/technetwork/developer-tools/jdev/downloads/JDeveloper11117-1917330.html.
- Oracle Utilities Customer Self Service version 2.1.0.1 (OUCSS2101.zip, available from the Oracle Software Delivery Cloud at edelivery.oracle.com).

- OUCSS Web Application: OUCSSWebApp2101.zip, available from the Oracle Utilities Documentation area on the Oracle Technology Network (OTN) web site (http://www.oracle.com/technetwork/apps-tech/utilities/documentation/index.html).
- Application Server with ADF Runtime installed (see the Oracle® Fusion Middleware Third-Party Application Server Guide for information on the application servers supported by Oracle).
- MetaData Services.

# Abbreviations

| | |
|---|---|
| OUCSS | Oracle Utilities Customer Self Service |
| CCB | Oracle Utilities Customer Care and Billing |
| MDM | Oracle Utilities Meter Data Management |
| NMS | Oracle Utilities Network Management System |
| MDS | Metadata Services |
| EAR | Enterprise Archive |
| OUCSS Web App | OUCSS Web Application |
| MW_HOME | Middleware Home |
| OUCSS_Product_Home | The folder in which the webcenter_WSS.zip file is extracted. |
| OUCSS-PortalInstall2.1.0.1 | OUCSS Portal install package for Service Pack 1 downloaded from Oracle Software Delivery Cloud (edelivery.oracle.com) |

# Chapter 2

# Installation and Uninstallation

## Installing on a WebLogic Server

### Prerequisites

Oracle Utilities Customer Self Service Web Application depends on OUCSS Schema tables, OUCSS data sources, OUCSS Shared Libraries, OUCSS Security credentials and Mail Session, and OUCSS BPEL flows which are part of the Oracle Utilites Customer Self Service package. It also requires suppported versions of Oracle Utilities edge applications as defined in *Chapter 2, "Installation - Software Requirements",* in the *OUCSS 2.1.0 Service Pack 1 Installation Guide*.

### Installation Steps

1. Download <OUCSS-PortalInstall2.1.0.1>.zip from the Oracle Software Delivery Cloud (edelivery.oracle.com) and extract into a target installation directory (e.g., `/u01/Oracle/Products/OUCSS` on Linux or `D:\Oracle\Products\OUCSS` on Windows).
2. Locate *webcenter_WSS.zip* in the installation folder and unzip the contents to <<MW_HOME>>/<<OUCSS_Product_Home>> subfolder.
3. Download OUCSSWebApp2.1.0.1.zip from the OTN and extract into a target installation directory (e.g., `/u01/Oracle/Products/OUCSSWebApp` on Linux or `D:\Oracle\Products\OUCSSWebApp` on Windows).
4. Copy CustomInstallBuild.xml and CustomUnInstallBuild.xml located under ./Deployables into <<OUCSS_Product_Home>>/ bin
5. Copy the ./OUCSSWebApp/Deployables/OUCSSWebApp_EAR.ear file into the ./<<OUCSS_Produce_Home>>/Install/application folder.
6. Create a **custom** folder under <<OUCSS_Product_Home>>/Install/DB/ and copy the ./Deployables/UpdateResource.sql file to the <<OUCSS_Product_Home>>/Install/DB/custom folder.

7 Apply the following procedures from the referenced sections of the *OUCSS 2.1.0 Service Pack 1 Installation Guide*:

**A** Perform all steps in the "Pre-Installation Tasks (Cluster or Standalone)" procedure in Chapter 2 of the *OUCSS Installation Guide*.

**B** Perform all steps in the "Configuring Edge Applications" procedure in Chapter 2 of the *OUCSS 2.1.0 Service Pack 1 Installation Guide*.

**C** Perform Steps 1 through 10 of the "Installing OUCSS Portal" procedure in Chapter 2 of the *OUCSS 2.1.0 Service Pack 1 Installation Guide*.

**Notes**:

- Enter server information for OUCSSWebApp under OUCSSPortal tags in InstallProperties.xml
- Since this install does not install OUCSS Mobile and OUCSSPortletProducer, in InstallProperties.xml, details under <oucssMobile></oucssMobile> and <oucssPortletProducer></oucssPortletProducer> can be left empty.

**Important**: After performing these procedures, check the logs to make sure there are no errors before proceeding.

8 Take a backup of the SS_RESOURCE and SS_CONFIGURATION table in the OUCSS Schema.

9 Run the following update command to update the SS_RESOURCE table.

**On Linux:**

```
ant -f CustomInstallBuild.xml UpdateWebAppData -
DInstallProperties=$PRODUCT_HOME/config/InstallProperties.xml -l DBUpdateTask.log
```

**On Windows:**

```
ant -f CustomInstallBuild.xml UpdateWebAppData -
DInstallProperties=%PRODUCT_HOME%\config\InstallProperties.xml -l DBUpdateTask.log
```

**Important**: After running the update command, check the logs to make sure there are no errors before proceeding.

10 Run the OUCSS Web Application installation command (shown below) to deploy the following OUCSS artifacts:

- Security Credentials (CSF Keys)
- *extend.oucss.portal.war* , *OUCSS_Extension.war* ,OUCSS_Commercial_Extension.war as shared libraries in WebLogic.
- Create Mail Session.
- Create OUCSS data source.
- Configure the web service connections to point to the edge applications as is configured in InstallProperties.xml.
- OUCSS Web Application.

**On Linux:**

```
ant -f CustomInstallBuild.xml InstallWebApp -
DInstallProperties=$PRODUCT_HOME/config/InstallProperties.xml -
DApplicationName=OUCSSWebApp -DApplicationVersion=2.1.0.1 -
DEARFileName=OUCSSWebApp_EAR.ear -l InstallWebApp.log
```

**On Windows:**

```
ant -f CustomInstallBuild.xml InstallWebApp -
DInstallProperties=%PRODUCT_HOME%\config\InstallProperties.xml -
DApplicationName=OUCSSWebApp -DApplicationVersion=2.1.0.1 -
DEARFileName=OUCSSWebApp_EAR.ear -l InstallWebApp.log
```

**Important**: After running the installation command, check the logs to make sure there are no errors before proceeding.

11 Perform the following Post-Installation checks as defined under ***Post-Installation Checklist*** *in OUCSS Installation Guide - Chapter 2)*

- Verify the OUCSS Schema tables.

- Verify the OUCSS Data Source.
- Verify Deployments.
- Verify that OUCSS Mail Session is Created.
- Verify web service connections.
- Verify the OUCSS Security Credential.

**12** Also perform the following verification:

- Verify through Weblogic console that OUCSSWebApp is created and is in Active state .
- Verify tokenization of the CCB Edge Application wsdl using the OUCSSWebApp(v2.1.0.1) from Application Deployments in the Oracle Enterprise Manager .

**13** Log in to the Application using http://<WebappHost>:<WebAppPort>/OUCSSWebApp.

The following screen should appear:

**14** Perform Post Installation steps on OUCSSWebApp as described in the *OUCSS Installation Guide Chapter 2 – Post Installation Steps*

**15** Restart the WC_CustomPortal managed server.

See *Chapter 2 "Installation - Installing OUCSS Inbound Services" and "Create Security Key Store" of OUCSS 2.1.0 Service Pack 1 Installation Guide* for installing OUCSS Inbound Services application if any of the functionality provided by the OUCSS Inbound Services is required.

See the *OUCSS 2.1.0 Service Pack 1 Installation Guide* for additional information regarding configuring BPEL flows and Edge applications.

# UnInstalling the Web Application

**1** Perform Step 1 from the procedure in Chapter 6, "Uninstalling OUCSS Portal, in the *OUCSS 2.1.0 Service Pack 1 Installation Guide*.

**2** Run the command shown below to uninstall the following OUCSS artifacts:

- Security Credentials Map
- OUCSS Mail Session
- OUCSS Shared libraries and extend .oucss.portal, used to extend OUCSS Web App
- OUCSS Web Application and corresponding MDS Partition
- OUCSS datasource

**On Linux:**

```
ant -f CustomUnInstallBuild.xml UnInstallWebApp -
DInstallProperties=$PRODUCT_HOME/config/InstallProperties.xml -
DApplicationName=OUCSSWebApp -DApplicationVersion=2.1.0.1 -
DEARFileName=OUCSSWebApp_EAR.ear -l UninstallWebApp.log
```

**On Windows:**

```
ant -f CustomUnInstallBuild.xml UnInstallWebApp -
DInstallProperties=%PRODUCT_HOME%/config/InstallProperties.xml -
DApplicationName=OUCSSWebApp -DApplicationVersion=2.1.0.1 -
DEARFileName=OUCSSWebApp_EAR.ear -l UninstallWebApp.log
```

Check the logs to make sure there are no errors before proceeding

3. Perform Steps 3 through 5 in the "OUCSS Portal" procedure (in Chapter 6 of the *OUCSS 2.1.0 Service Pack 1Installation Guide*) to delete the OUCSS Schema, restart managed servers, and remove Users and Groups.
4. Verify the uninstallation operation:
   - **A** Log in to the WebLogic console as wlsadminuser/wlsadminpasswd.
   - **B** Choose **portal_domain > Deployments** and verify that the following are no longer listed:
     - com.oracle.ugbu.ss.lib (11.1.1, 11.1.1.8.0)
     - extend.oucss.portal (11.1.1, 11.1.1.*)
     - OUCSSInBound (enterprise application)
     - OUCSSWebApp(v2.1.0.1)
   - **C** Choose **Services > Data Sources** and verify that the **OUCSSDS** data source is no longer present.
   - **D** Choose **Services** > **Mail Sessions**. Verify that the **OUCSS** mail session is no longer present.
   - **E** Verify that OUCSS DB User is dropped from the database.

# Chapter 3

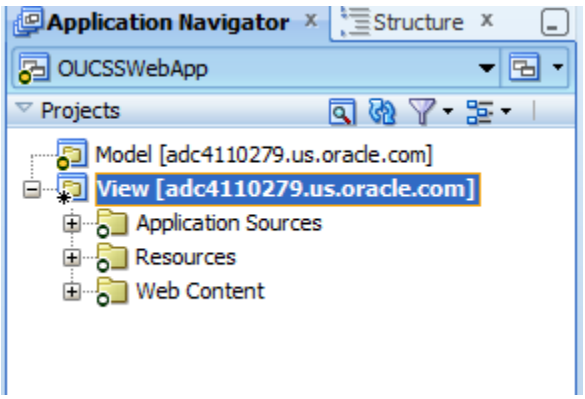
# Extending the OUCSS Web Application Sample

This section explains the various components of the OUCSS Web application, and describes how to extend the component using the ExtendOUCSSPortal application.

This approach may require changes in a few configuration files, but does not otherwise modify the sample application so that future upgrades to the sample application are not impacted.

Alternatively, the sample web application itself can be modified using the same general steps described to extend ExtendOUCSSPortal.

To get started, download the OUCSSWebApp2.1.0.1.zip from OTN and extract the contents to a folder on your hard disk drive. Locate and open the the OUCSSWebApp.jws project in JDeveloper. JDeveloper's Application Navigator will display the following project Model and View:

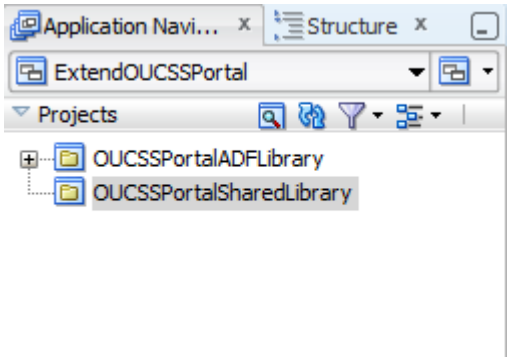



Various components in the OUCSS web application are grouped in the following directory structure:

- **Jspx pages** : ./OUCSSWebApp/View/public_html/oracle/ugbu/ss/portal/pages/*
- **Page template:** ./OUCSSWebApp/View/public_html/oracle/ugbu/ss/portal/pages/template/OUCSSTemplate.jspx

- **Managed beans**: ./OUCSSWebApp/View/src/java/oracle/ugbu/ss/portal/backing/*
- **Menu model**: ./OUCSSWebApp/View/public_html/WEB-INF/oucss/*
- **Page flows (Unbounded taskflows):** ./OUCSSWebApp/View/public_html/WEB-INF/menu/*

To extend the application using ExtendOUCSSPortal, extract the contents of ExtendOUCSSPortal.zip located under <<OUCSS_Portal_home>>/Install/application and open ExtendOUCSSPortal.jws in JDeveloper. The Application Navigator will show two projects (OUCSSPortalADFLibrary and OUCSSPortalSharedLibrary):




ExtendOUCSSPortal has the following directory structure under OUCSSPortalADFLibrary to enable security on these taskflows and jspx:

- 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 permissions to these folders are already set up in jazn.xml of OUCSSWebApp.

# Setting up JDeveloper for Development

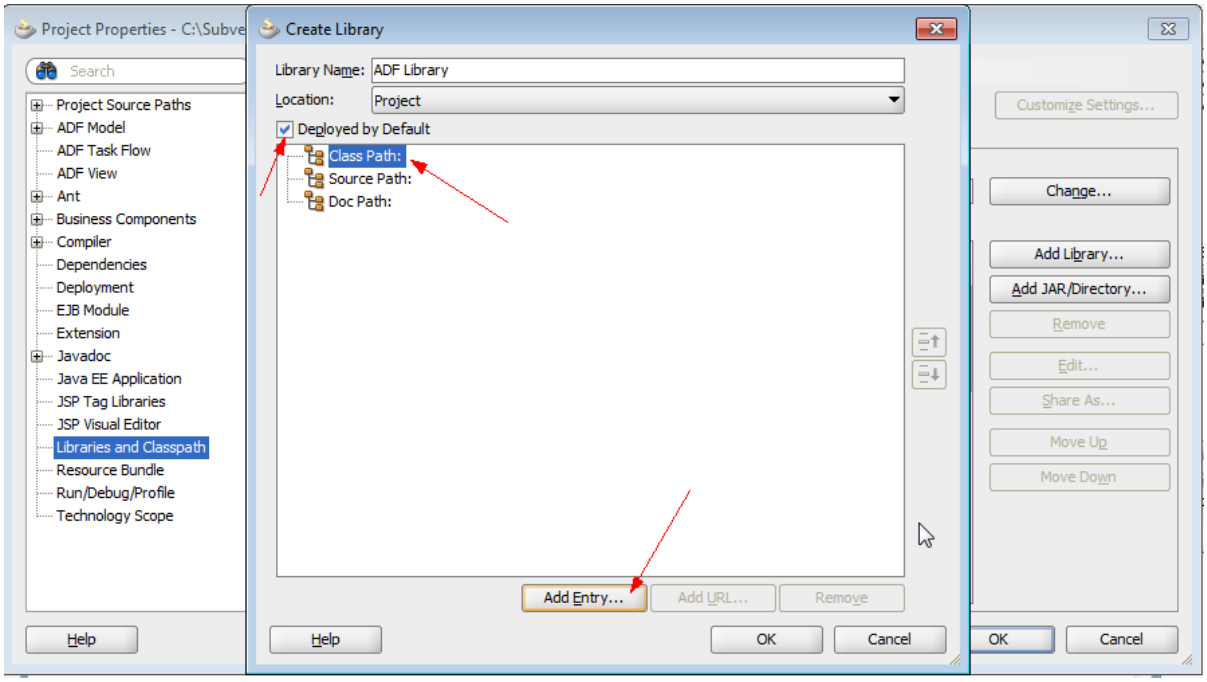
## Adding Shared Libraries to the Project

1. Locate *webcenter_WSS.zip* in the installation folder and unzip the contents to <<MW_HOME>>/<<OUCSS_Product_Home>> subfolder.
2. Extract <<OUCSS_Product_Home>>/Install/application/OUCSS_Extension.war and <<OUCSS_Product_Home>>/Install/application/OUCSS_Commercial_Extension.war into a folder.
3. Right click on **View project > Project Properties > Libraries and ClassPath > Add Library**.
4. Click **New** to add a new Library.
5. Enter a name for the New Library, For eg: ADF Library as shown below.
6. Click **Add Entry** and add the path to the folder containing the OUCSS jars.( <<OUCSS_Product_Home>>/Install/application/OUCSS_Extension/WEB-INF/lib/oracle. ugbu.ss.shared.*.jar and

<<OUCSS_Product_Home>>/Install/application/OUCSS_Commercial_Extension/WEB-INF/lib/oracle.ugbu.ss.shared.commercial.*.jar ).

7 Ensure that the **Deployed by Default** box is checked.

8 Click **OK**.

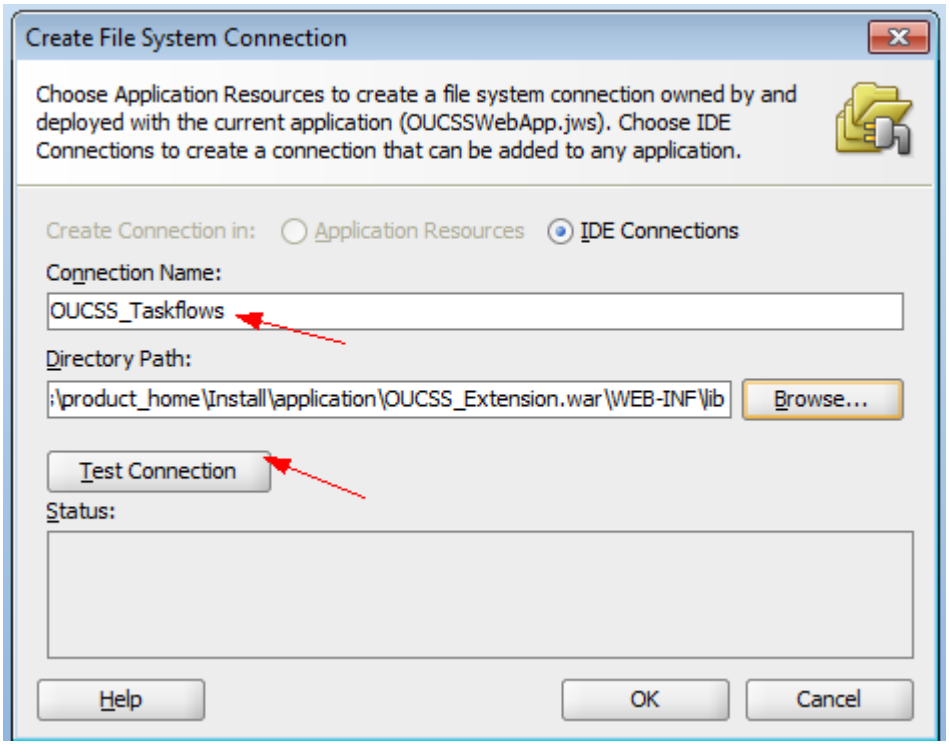



# Creating a File System Connection to the Taskflows Directory

If you want to build custom jspx pages that consume OUCSS taskflows, you can also create a File System connection to the directory containing the OUCSS taskflows. This lets you drag-and-drop taskflows to the jspx pages.

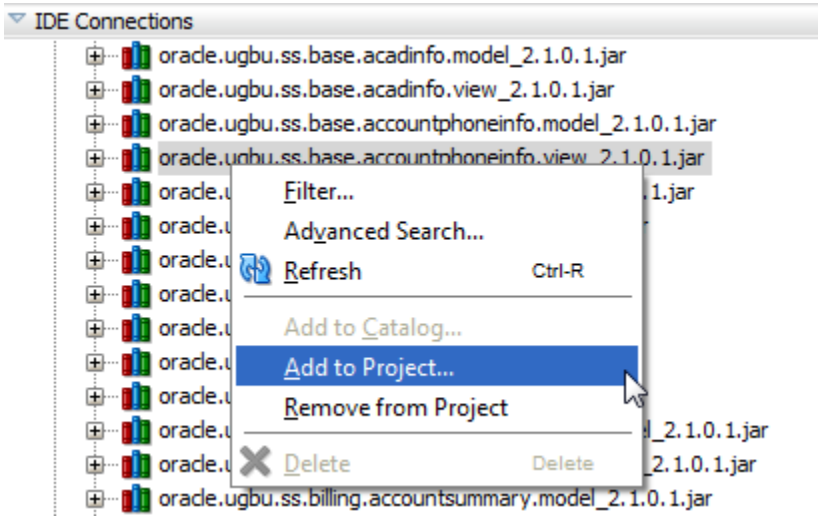
1 Using the JDeveloper Resource Palette, create a New File System connection as "OUCSS_Taskflows" with the Directory Path pointing to <<OUCSS_Product_Home>>/Install/application/OUCSS_Extension/WEB-INF/lib.

2 Click **Test Connection** to validate.

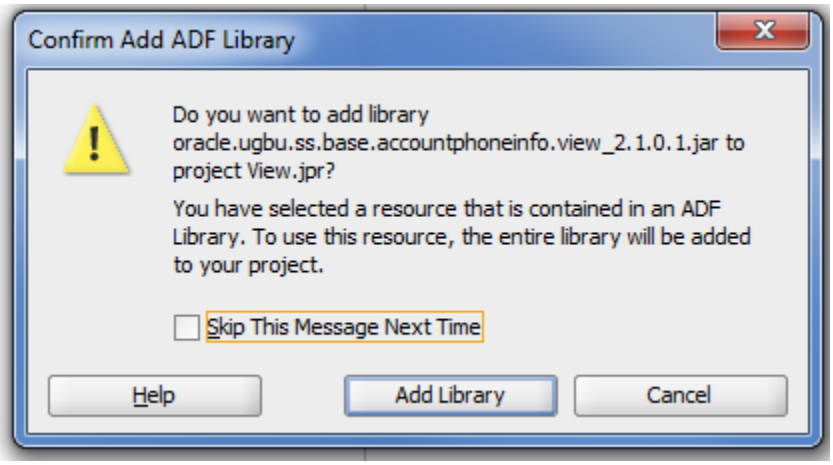



3 Create another connection for OUCSS_Commercial_Taskflows with directory Path pointing to <<OUCSS_Product_Home>>/Install/application/OUCSS_Commercial_Extension/WEB-INF/lib.

4 Add the required libraries to the project as shown below:




5 Click Add Library when prompted. This adds the library to the OUCSSWebApp.




# Building an ADF Web Application

## Managed Beans

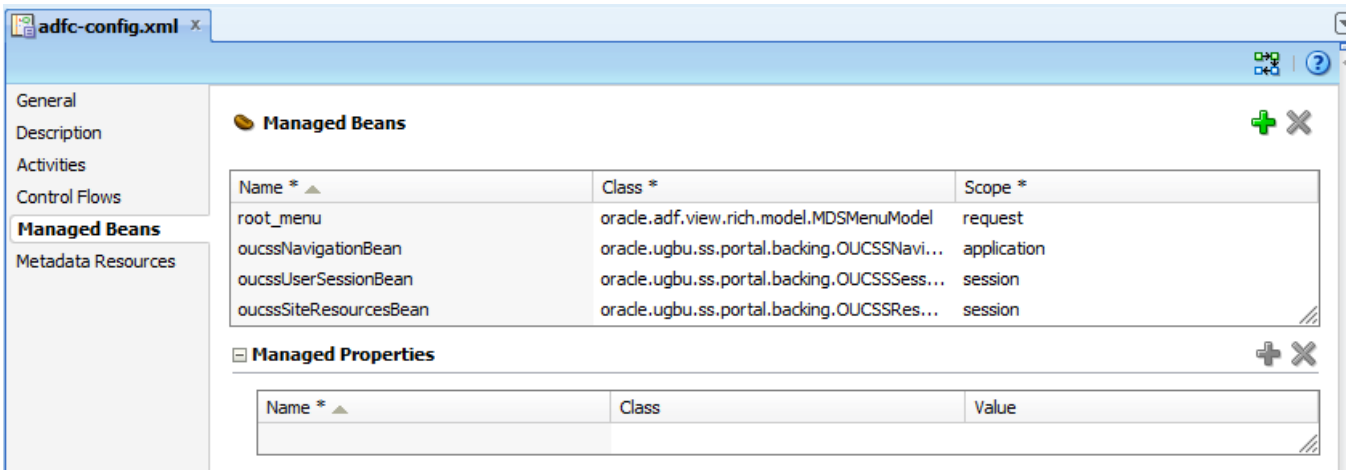
Managed beans are Java classes which are used to store logic related either to the UI , a taskflow, or to an application in general. Managed beans, along with their properties and methods, can be referenced through EL expressions .They are registered to the application through configuration files.

The following are the managed beans in the OUCSS Web application that are referenced in the top level jspx pages, page templates, and menus:

- OUCSSNavigationBean
- OUCSSResourcesBean
- OUCSSSessionBean
- OUCSSErrorProxyBean
- OUCSSPortalCSRViewBean
- OUCSSPortalLandingPageBean

These beans handle various application-related logic such as user validation, getting resources, and handling navigation.

They are registered with the main unbounded taskflow (adfc-config.xml).




**Note**: For more information on managed beans, registering with different configuration files, and a description of their scope within an application, see Getting Started with Your Web Interface in the *Oracle® Fusion Middleware Fusion Developer's Guide for Oracle Application Development Framework* documentation.

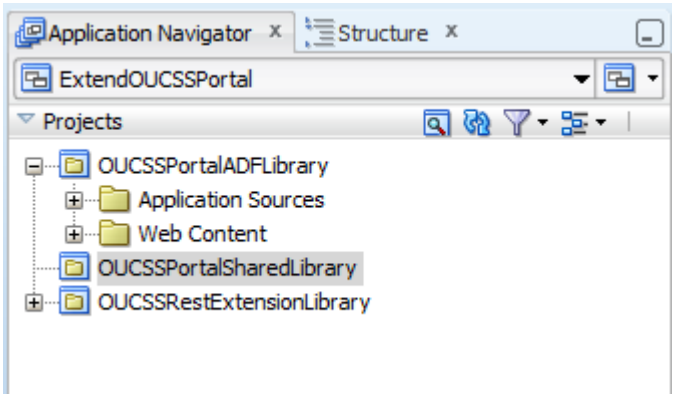
You can create your own managed beans in the OUCSSPortalADFLibrary project in the ExtendOUCSSPortal application within the proper package structure.

# Look and Feel: Page Templates and Skins

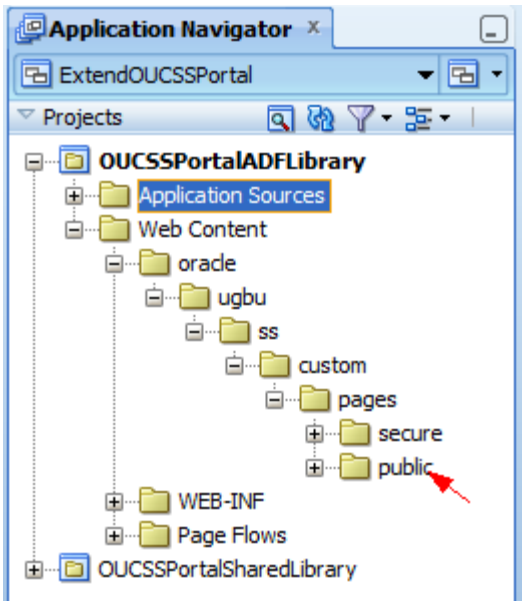
The OUCSS Web application sample is packaged with OUCSSTemplate.jspx (./OUCSSWebApp/View/public_html/oracle/ugbu/ss/portal/pages/template/OUCSSTemplate.jspx), which is used to define the layout of the navigation model and the page content.

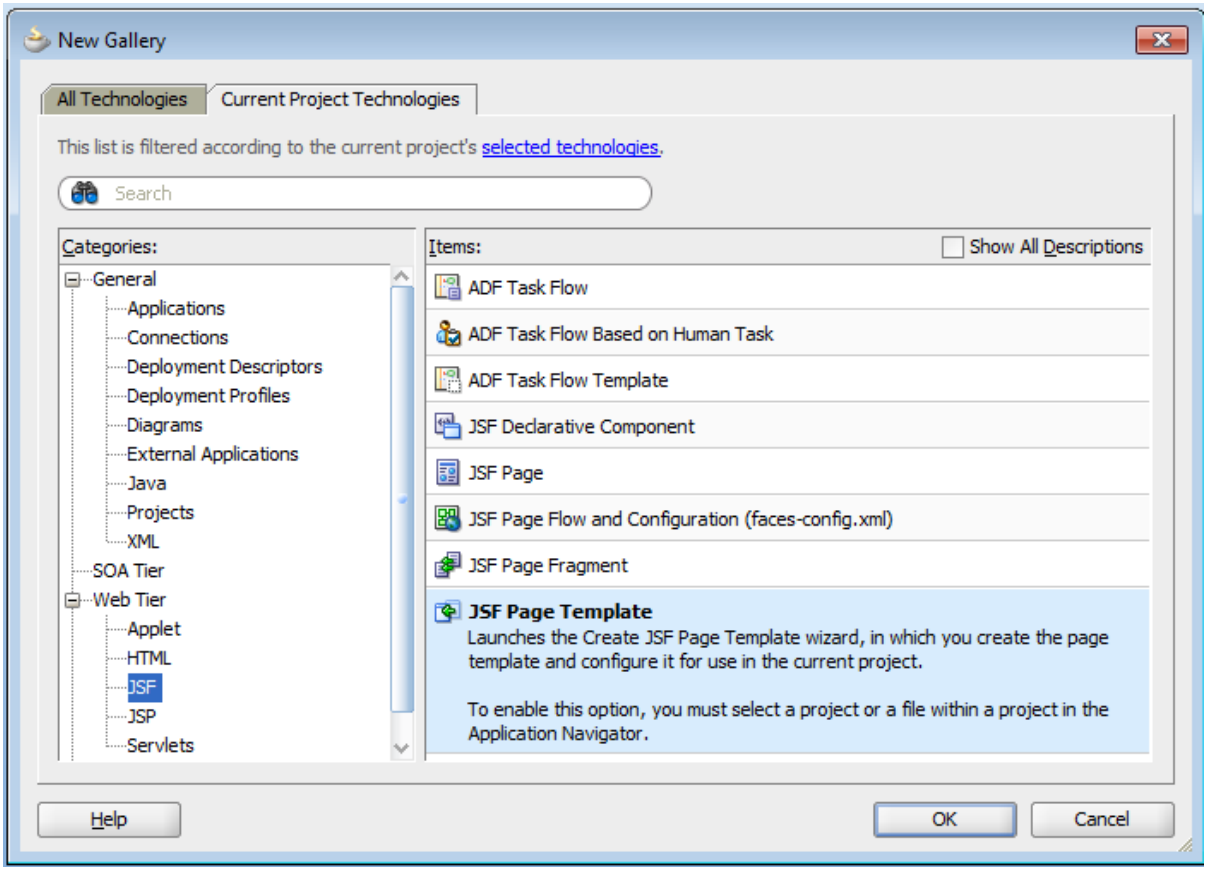
The default page template can be customized for a different look and feel, or a new page template can be created. To create a new template in the ExtendOUCSSPortal application:

1. Open the ExtendOUCSSPortal application in JDeveloper.



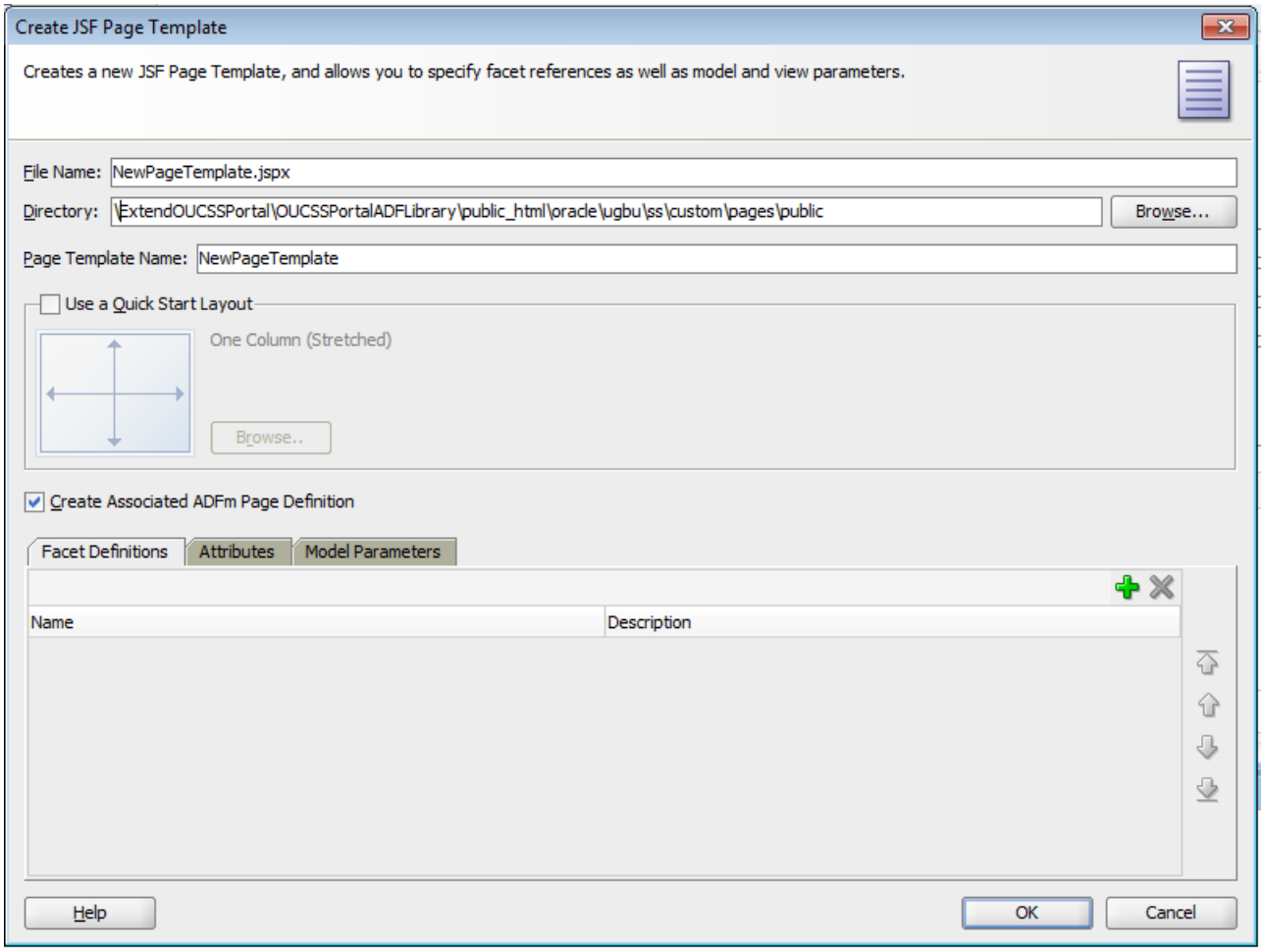

2. In OUCSSPortalADFLibrary, right-click the folder under oracle.ugbu.ss.custom.pages.public for which you would like the page template created. Click on **New>JSF>JSF Page Template**.




3 Check the "Create Associated AdFm Page Definition" check box.

4 Click **OK** to open a wizard that lets you define various components of page template.




**Note**: For more information on creating page templates see section 19.3.1, How to Create a Page Template, in the *Oracle® Fusion Middleware Web User Interface Developer's Guide for Oracle Application Development Framework*.

5 Update the SS_RESOURCE table if any of the Residential ,Commercial, or Default page templates are changed.

| RESOURCE_CD | Description |
|---|---|
| PORTAL_ST_COMMERCIAL | Site Template to be used for Commercial Customers. |
| PORTAL_ST_PUBLIC | Site Template to be used for Public Customers |
| PORTAL_ST_RESIDENTIAL | Site Template to be used for Residential Customers |

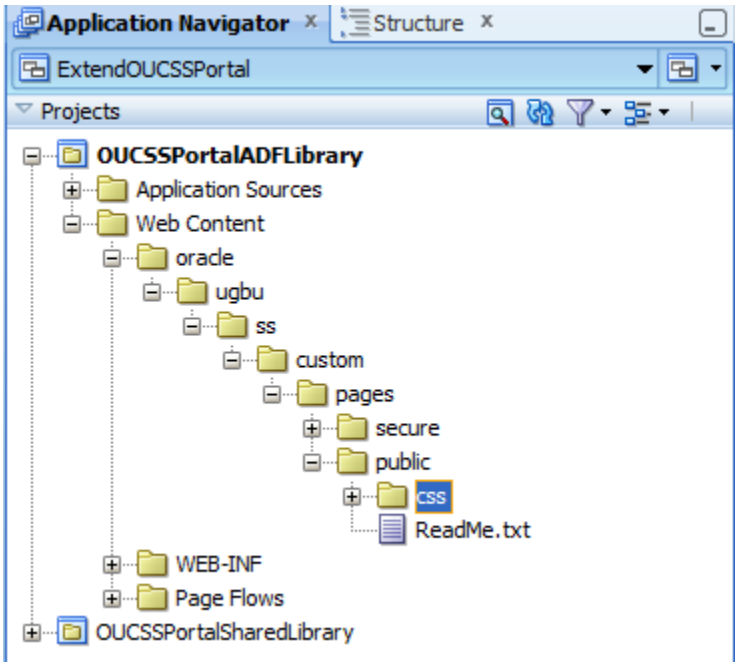
**Note**: The Resource table can also be changed through the **Admin > Resources** screen in OUCSS.

# Applying Skins

A skin is a stylesheet based on CSS 3.0 which can be applied to entire application for consistent look and feel. OUCSS Web Application uses the default skin **skyros** provided by ADF 11.1.1.7. The skin used by the application is defined in trinidad-config.xml:

```
<?xml version="1.0" encoding="windows-1252"?>
<trinidad-config xmlns="http://myfaces.apache.org/trinidad/config">
  <skin-family>skyros</skin-family>
  <skin-version>v1</skin-version>
</trinidad-config>
```

You can create a custom skin by extending one of the skins provided by ADF Faces. To create a custom skin through the ExtendOUCSSPortal project, create a custom CSS file in a folder under the Web Content structure as shown in the following image:




The customization could then be implemented as follows:

1. Create a new CSS file named CustomSkin.css under .\ExtendOUCSSPortal\OUCSSPortalADFLibrary\public_html\oracle\ugbu\ss\custom\pages\public\css folder and edit it with the following CSS specifications:

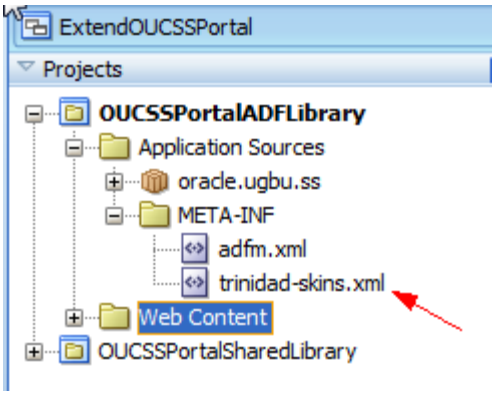
   CustomSkin.css

   Find

   ```
   1  .AFDefaultFontFamily:alias {font-family: Arial, Helvetica, sans-serif}
   2  .AFDefaultFont:alias {font-size: 16px}
   3  af|goButton::access-key {color: red;}
   4  af|inputText::content{background-color:Gray}
   5
   ```

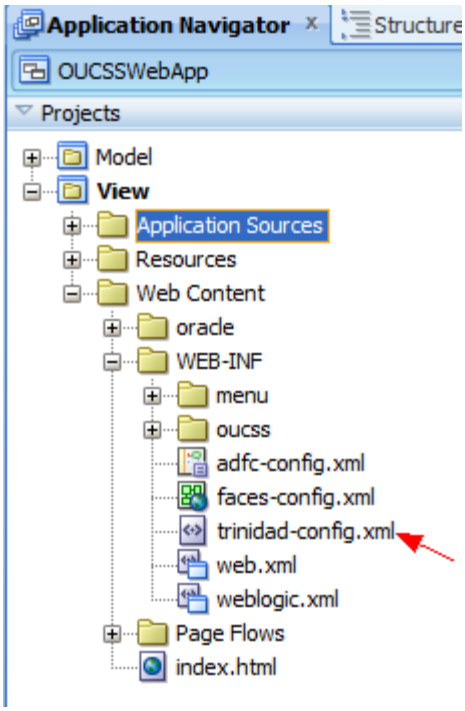
   The basic steps required to add a custom skin are described in the topics, "How to Add a Custom Skin to an Application", "How to Register the XML Schema Definition File for a Custom Skin", and "How to Register a Custom Skin" in the Oracle Fusion Middleware Web User Interface Developer's Guide for Oracle Application Development Framework documentation (http://docs.oracle.com/cd/E28280_01/web.1111/b31973/af_skin.htm%23ADFUI331).

2. Create trinidad-skins.xml under META-INF as shown in the following image. This example extends the *skyros* skin from ADF.




The trinidad-skins.xml file would contain the following:

**3** By opening the OUCSSWebApp application in JDeveloper, the trinidad-config.xml file could then be modified to reflect the new skin, as shown in the following images:



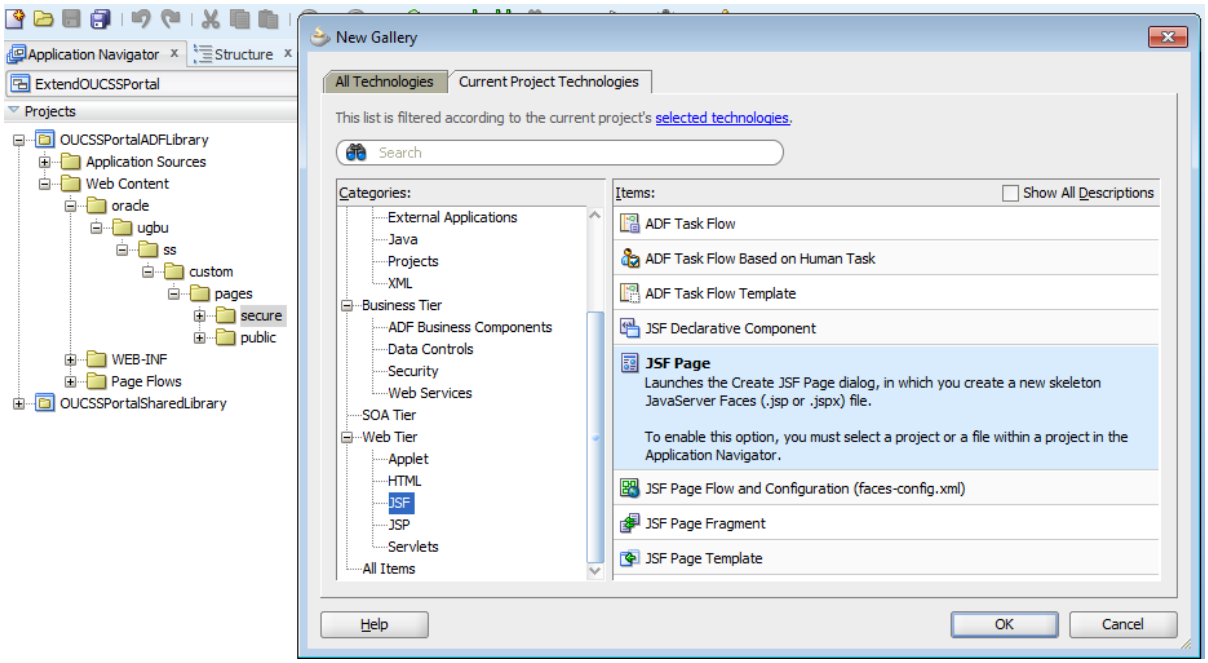
```
trinidad-config.xml
Find
1  <?xml version="1.0" encoding="windows-1252"?>
2  <trinidad-config xmlns="http://myfaces.apache.org/trinidad/config">
3
4    <skin-family>custom</skin-family>
5    <skin-version></skin-version>
6  </trinidad-config>
```

**Note**: For more information on how to add a custom skin, register a custom skin and use custom skin in the application, see "Introduction to Skins, Style Selectors, and Style Properties" in the *Oracle Fusion Middleware Web User Interface Developer's Guide for Oracle Application Development Framework* documentation.

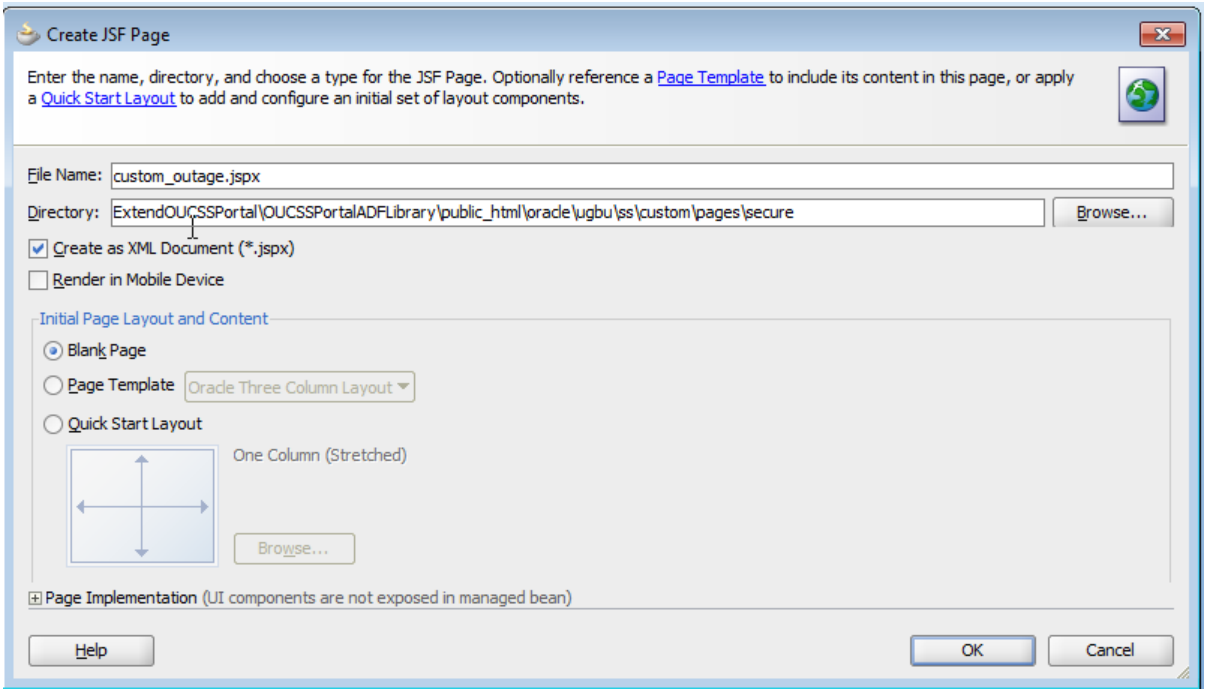
# Creating *jspx* Pages

OUCSSWebApp uses the OUCSSTemplate under ./OUCSSWebApp/View/public_html/oracle/ugbu/ss/portal/pages/template/* to structure the layout.

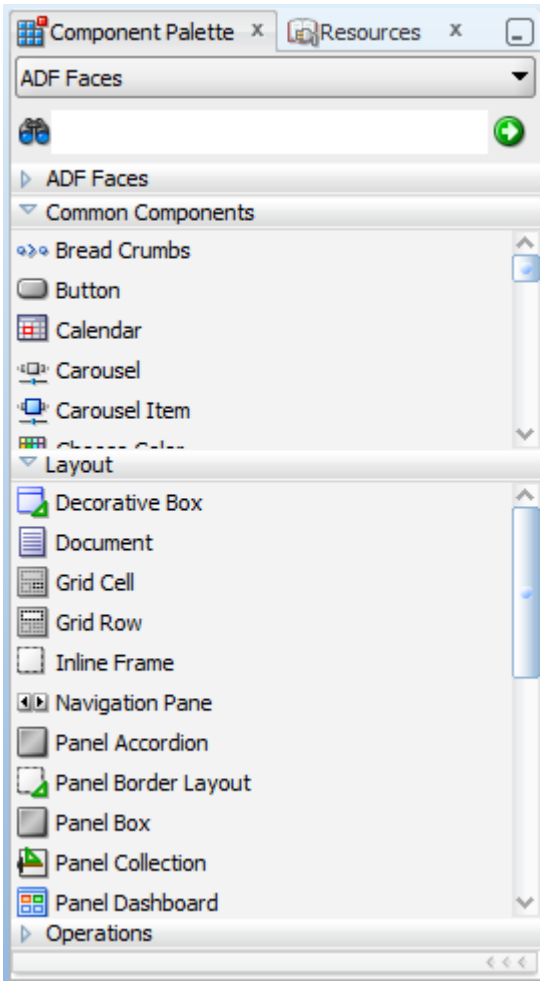
1. To create new jspx pages in ExtendOUCSSPortal application, open ExtendOUCSSPortal application in JDeveloper.
2. To create a new jspx page right-click on the folder in which you wish to create your jspx page in the OUCSSPortalADFLibrary project.
   - Create pages under `\oracle\ugbu\ss\custom\pages\secure` 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).
3. Choose **New>JSF>JSF Page** on the folder you wish to create the new jspx pages.




4. Enter the name and click **OK**.




**5** Drag and Drop the required components from the Component Palette or taskflows from the shared libraries into the page.




6 Make sure the newly created jspx has a reference to the page template.

```
<?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">
  <jsp:directive.page contentType="text/html;charset=UTF-8"/>
  <f:view>
    <af:document id="d1" title="Custom Outage">
      <af:form id="f1">
        <af:pageTemplate value="#{bindings.pageTemplateBinding.templateModel}" id="wssptl">
          <f:facet name="content">
            <af:panelGroupLayout id="pgl1">
              <af:outputText value="Custom Jspx !!!" id="ot1"/>
            </af:panelGroupLayout>
          </f:facet>
        </af:pageTemplate>
      </af:form>
    </af:document>
  </f:view>
</jsp:root>
```

7 Update the SS_RESOURCE table if any new landing pages are created.

| RESOURCE_CD | Description |
|---|---|
| PORTAL_LAND_COMMERCIAL | Landing page for Commercial Login for users with Multiple Accounts. |
| PORTAL_LAND_COMMERCIAL_SA | Landing page for Commercial Login for user with Single Account. |
| PORTAL_LAND_DEFAULT | Default Landing page if the context is not configured. |
| PORTAL_LAND_RESIDENTIAL | Landing page for Residential Login for users with Multiple Accounts. |
| PORTAL_LAND_RESIDENTIAL_SA | Landing page for Residential Login for user with Single Account. |

**Note**: The Resource table can also be changed through **Admin** > **Resources** screen in OUCSS.

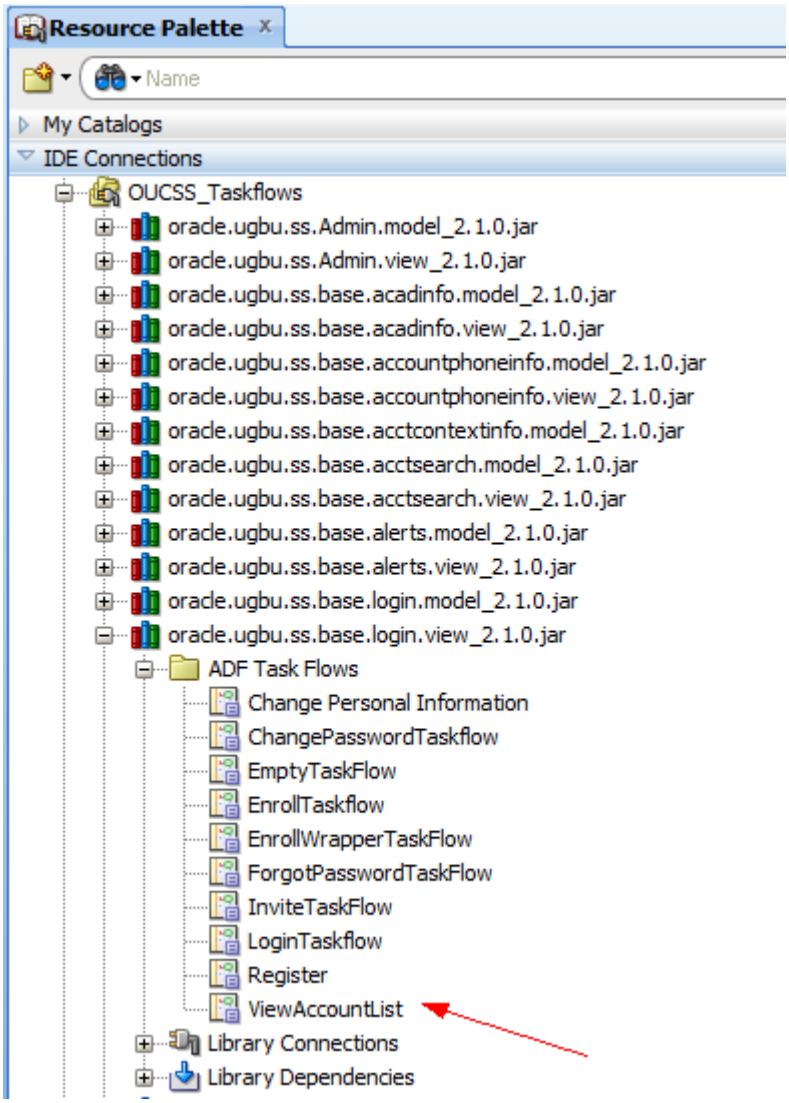
# Creating Custom Taskflows

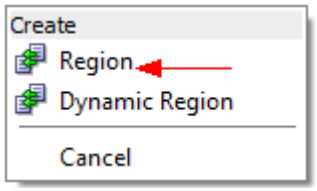
To create custom taskflows and pages using the extension project (ExtendOUCSSPortal), see steps 1-3 in the chapter "Extending the OUCSS Portal " in the *Customizing and Extending OUCSS Custom* Portals whitepaper (available for download in the Oracle Utilities Customer Self Service section of the Oracle Utilities Documentation area on the OTN web site at http://www.oracle.com/technetwork/apps-tech/utilities/documentation/index.html).

# Consuming OUCSS Taskflows

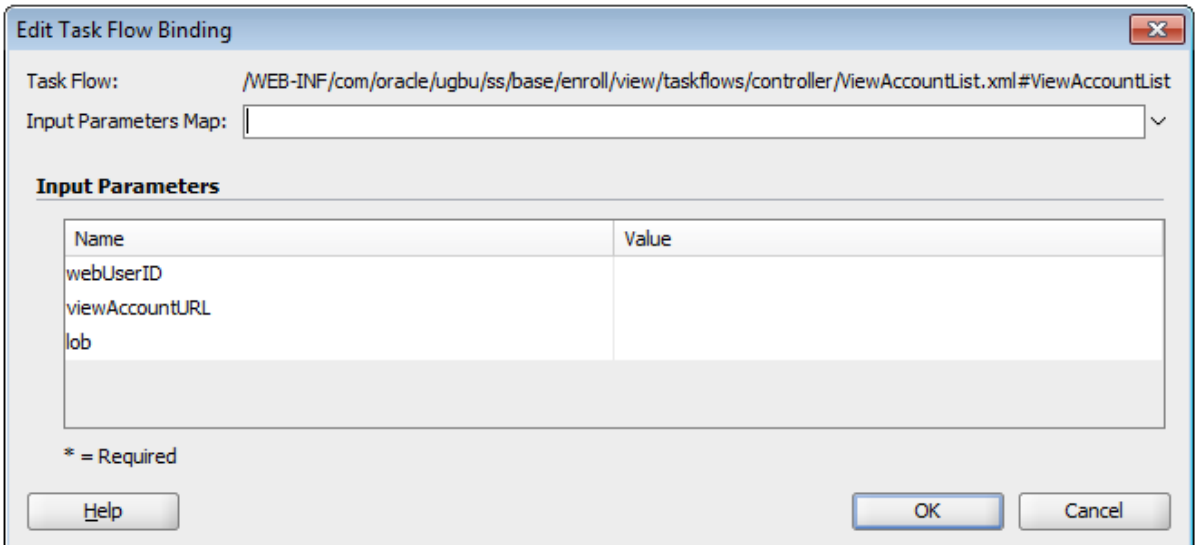
To consume an OUCSS taskflow, drop the taskflow from the Resource Palette through to the jspx.

Example: Select the ViewAccountList taskflow from the Resource Palette and drop it on the jspx as a "Region", as shown in the following two images.




After dropping the taskflow on the jspx, enter the parameters required for the taskflow as either static values or as an EL expression.




**Notes**:

- Multiple taskflows can be dropped on to the same jspx page.
- When a library is added to the project for the first time, you must click "Add Library" in the JDeveloper confirmation popup.

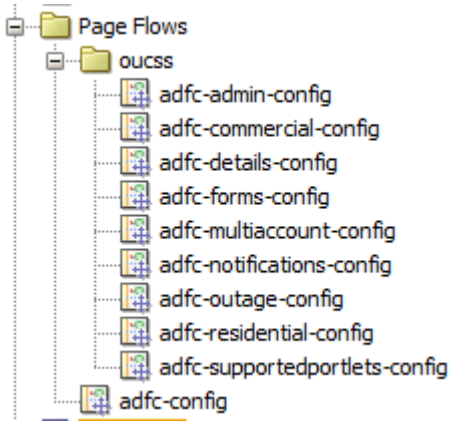
The following image shows AccountList.jspx source code after ViewAccountList and Enroll taskflows are dropped. Note that the created pages reference the page templates.

```
<?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">
  <jsp:directive.page contentType="text/html;charset=UTF-8"/>
  <f:view>
    <af:document id="d1" title="#{portalBundle['ACCOUNT_LIST_TITLE']}">
      <af:form id="f1">
        <af:pageTemplate value="#{bindings.pageTemplateBinding.templateModel}" id="wssptl">
          <f:facet name="content">
            <af:panelGroupLayout id="g1" layout="vertical" halign="start" valign="top">
              <af:region value="#{bindings.ViewAccountList1.regionModel}"
                         id="r1"/>
              <af:region value="#{bindings.EnrollWrapperTaskFlow1.regionModel}"
                         id="r2"/>
            </af:panelGroupLayout>
          </f:facet>
        </af:pageTemplate>
      </af:form>
    </af:document>
  </f:view>
</jsp:root>
```

# Navigation: Creating Menus and Page Flows

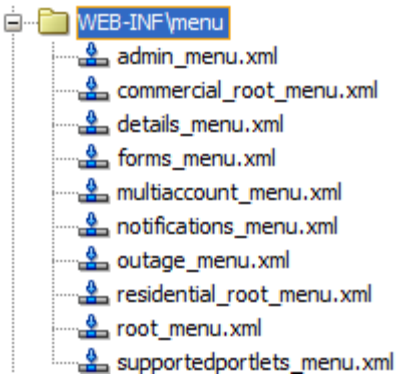
An unbounded taskflow is created for every main menu and sub menu item.

The following image shows the unbounded taskflows used in OUCSSWebApp.

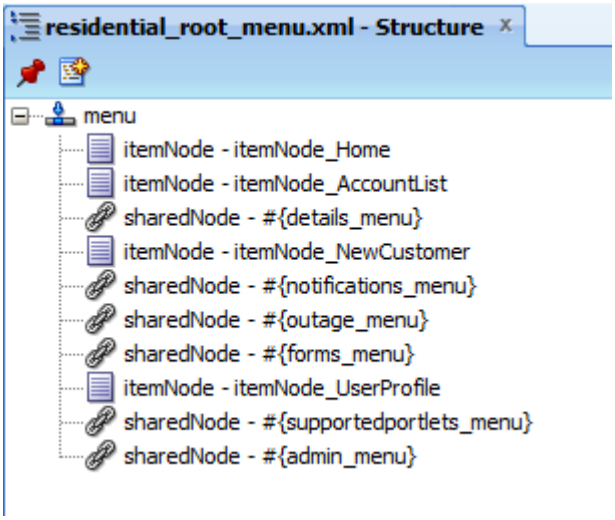



An ADF menu is created for every unbounded taskflow created for navigation.

The following image shows the ADF Menus used in OUCSSWebApp:




The following image shows a sample structure for the residential_root_menu:

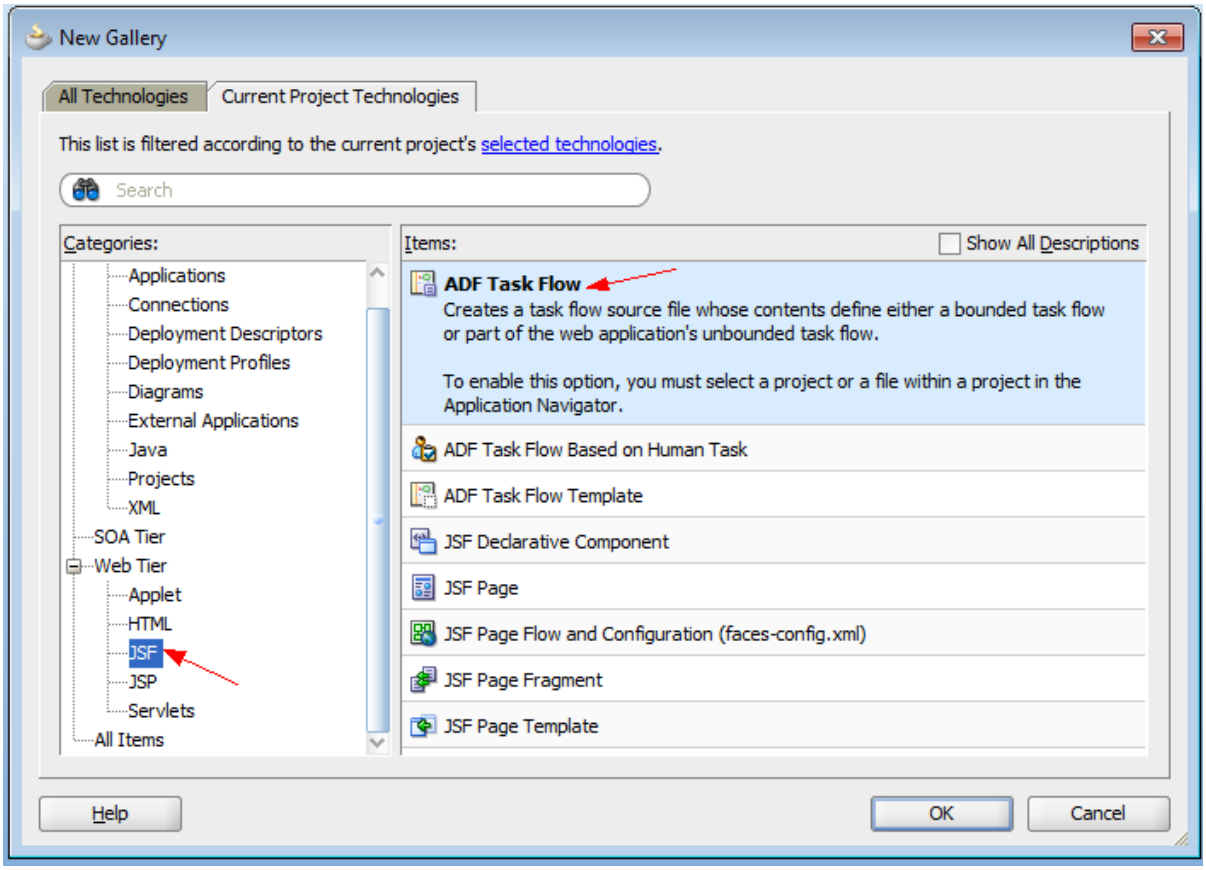


The following table shows the mapping between OUCSS menus, the ADF menu item, and its corresponding unbounded taskflow.

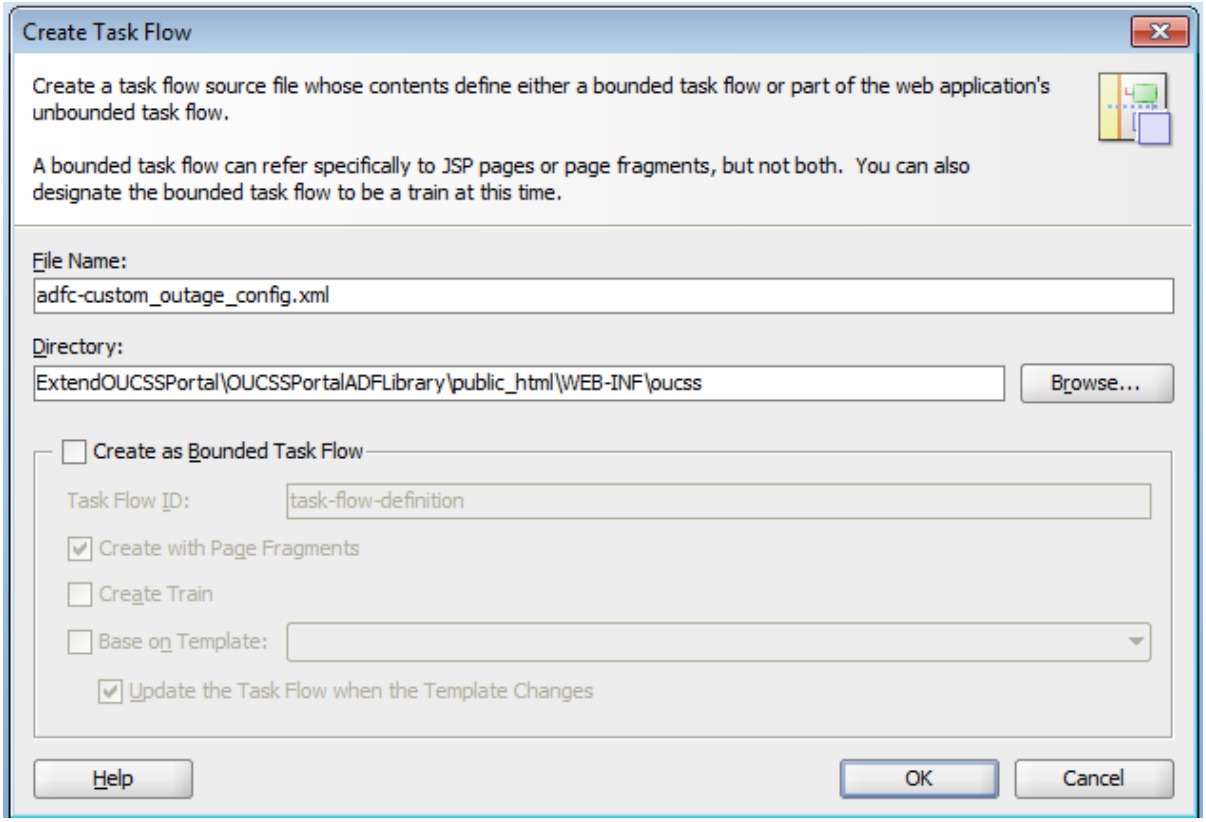
| OUCSS Web App Menu | Unbounded Taskflow | ADF Menu |
|---|---|---|
| Public menu - Main | adfc-config.xml | root_menu.xml |
| Residential - Main menu | adfc-residential-config.xml | residential_root_menu.xml |
| Commercial- Main menu | adfc-commercial-config.xml | commercial_root_menu.xml |
| Admin - Sub menu | adfc-admin-config.xml | admin_menu.xml |
| Account Details - Sub menu | adfc-details-config.xml | details_menu.xml |
| Outage - Sub menu | adfc-outage-config.xml | outage_menu.xml |
| Notifications - Sub menu | adfc_notifications-config.xml | notifications_menu.xml |
| Old (Supported) Portlets-(Sub menu) | adfc-supportedportlets-config.xml | supportedportlets_menu.xml |
| Forms Management- Sub menu | adfc-forms-config.xml | forms_menu.xml |
| Multi Account(Commercial) - Sub menu | adfc-multiaccount-config.xml | multiaccount_menu.xml |

**To create custom Unbounded taskflows in an ExtendOUCSSPortal application:**

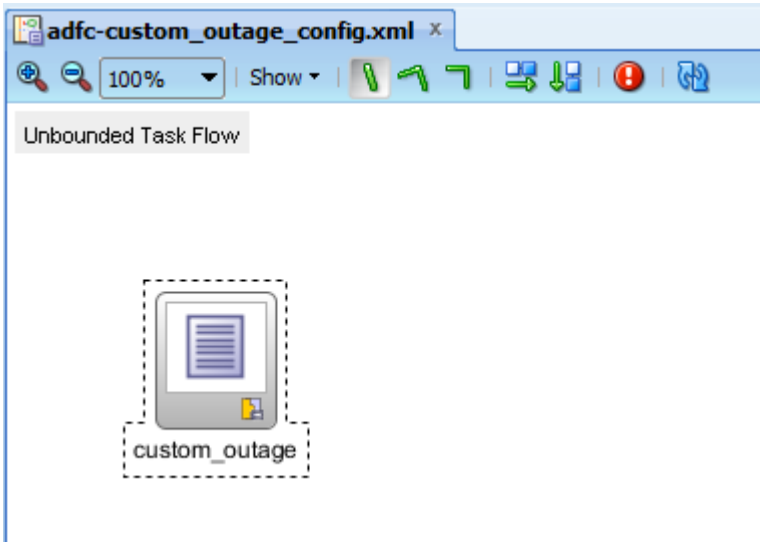
1. Open the ExtendOUCSSPortal application in JDeveloper.
2. In the OUCSSPortalADFLibray project under WEB-INF, create a folder in which the unbounded taskflow is to be created.
3. Right-click the folder and choose **New** > **Web tier** > **JSF** > **ADF Taskflow**.




4 Enter the name of the unbounded taskflow. Uncheck **"Create as Bounded Taskflow**" and press **OK**.

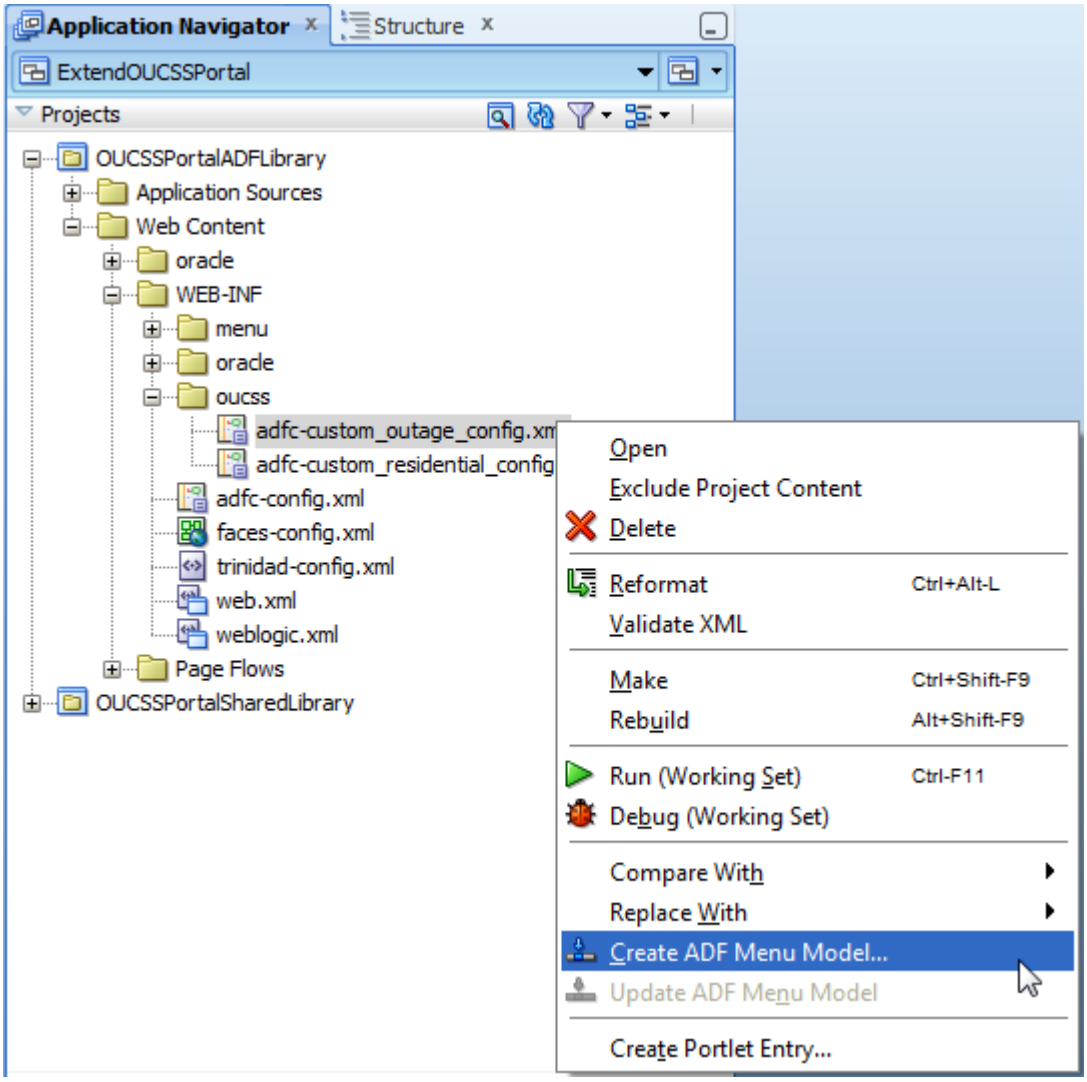



**5** Drop the required jspx pages in this unbounded taskflows as a "View" Activity (e.g., custom_outage.jspx).

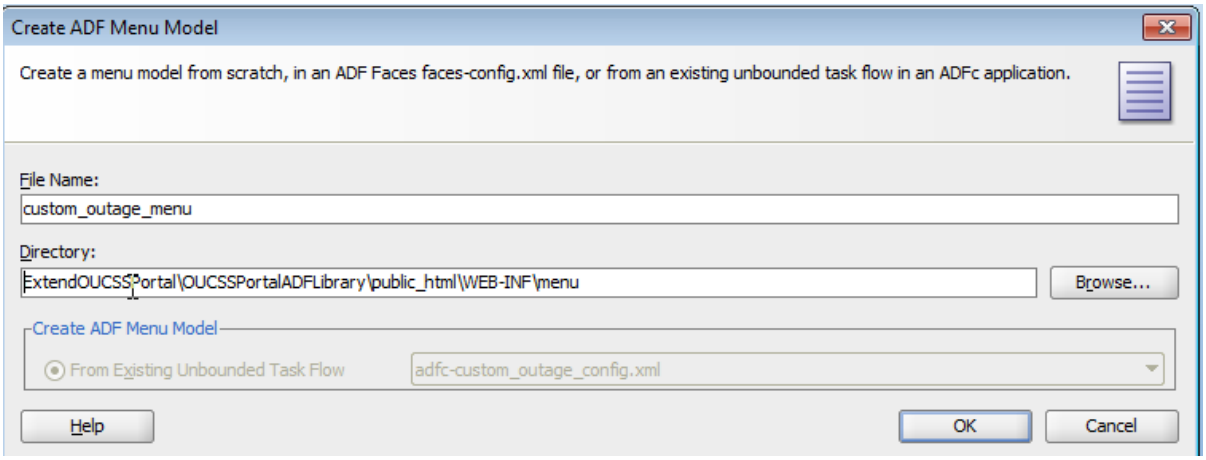


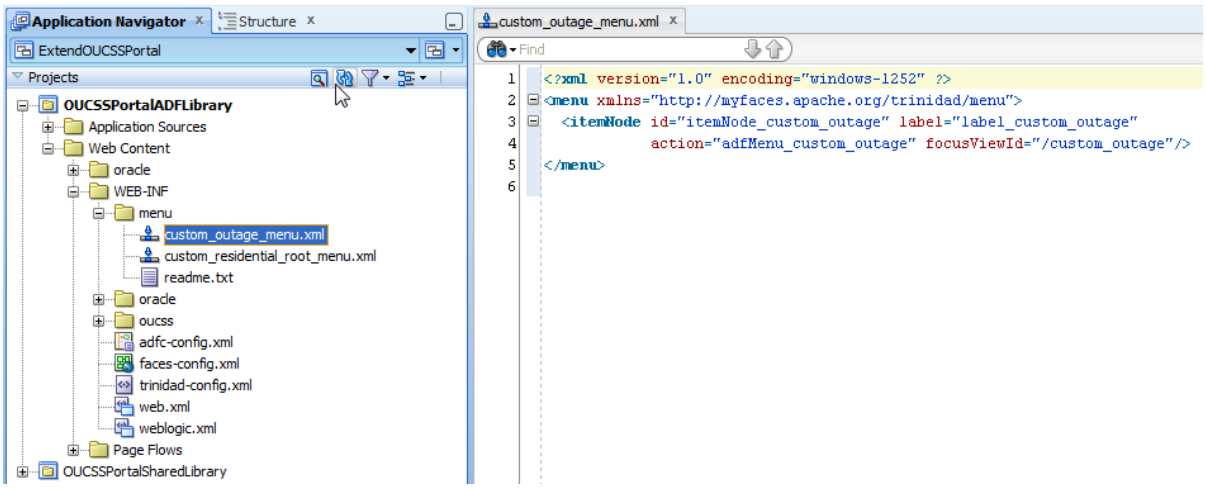

**To create a custom ADF Menu Item in the ExtendOUCSSPortal application:**

**1** To create the corresponding ADF Menu item, right click on the adfc-custom_outage-config.xml and choose **Create ADF Menu Model**.

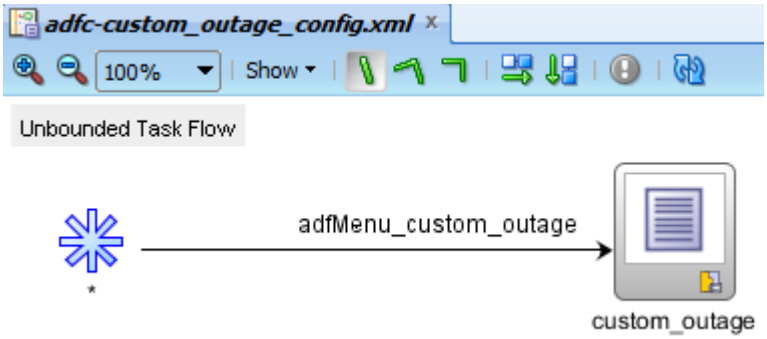



**2** Enter the name and click **OK**.






The following image represents the adfc-custom_outage-config.xml:




**Note**: For more information on creating itemNode, sharedNodes, and groupNodes, see Using a Menu Model to Create a Page Hierarchy" in the *Oracle® Fusion Middleware Web User Interface Developer's Guide for Oracle Application Development Framework* documentation.

3 Update SS_RESOURCE table if any of the Residential or Commercial main menus are changed.

| RESOURCE_CD | Description |
|---|---|
| PORTAL_NAV_COMMERCIAL | Navigation Model URL for Commercial Customers |
| PORTAL_NAV_PUBLIC | Navigation Model URL for Public Customers |
| PORTAL_NAV_RESIDENTIAL | Navigation Model URL for Residential Customers |

**Note**:The Resource table can also be changed through the **Admin>Resources** screen in OUCSS.

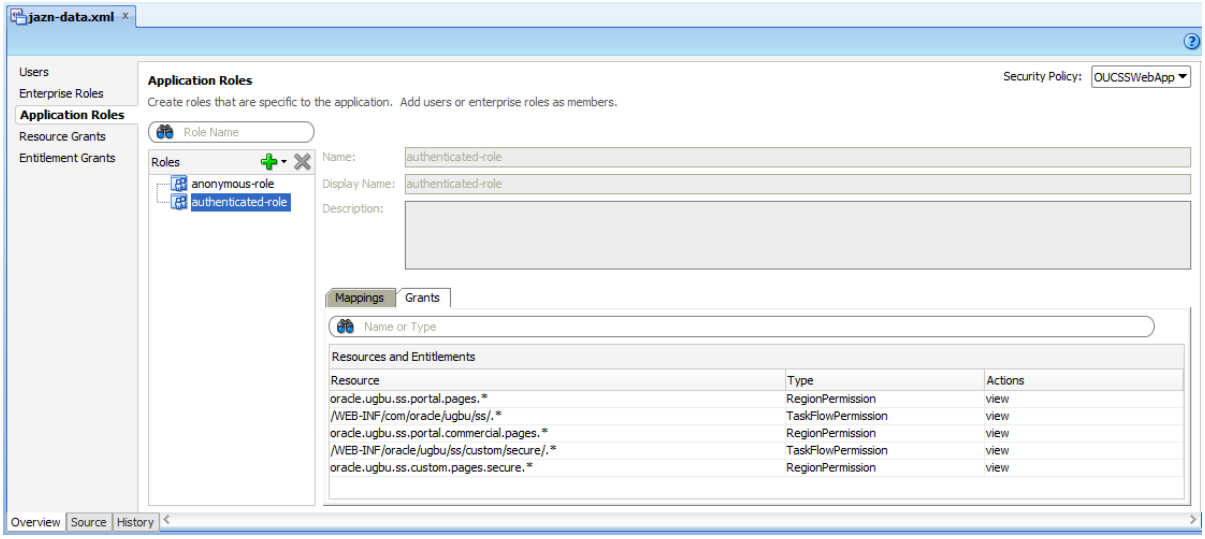
# Security

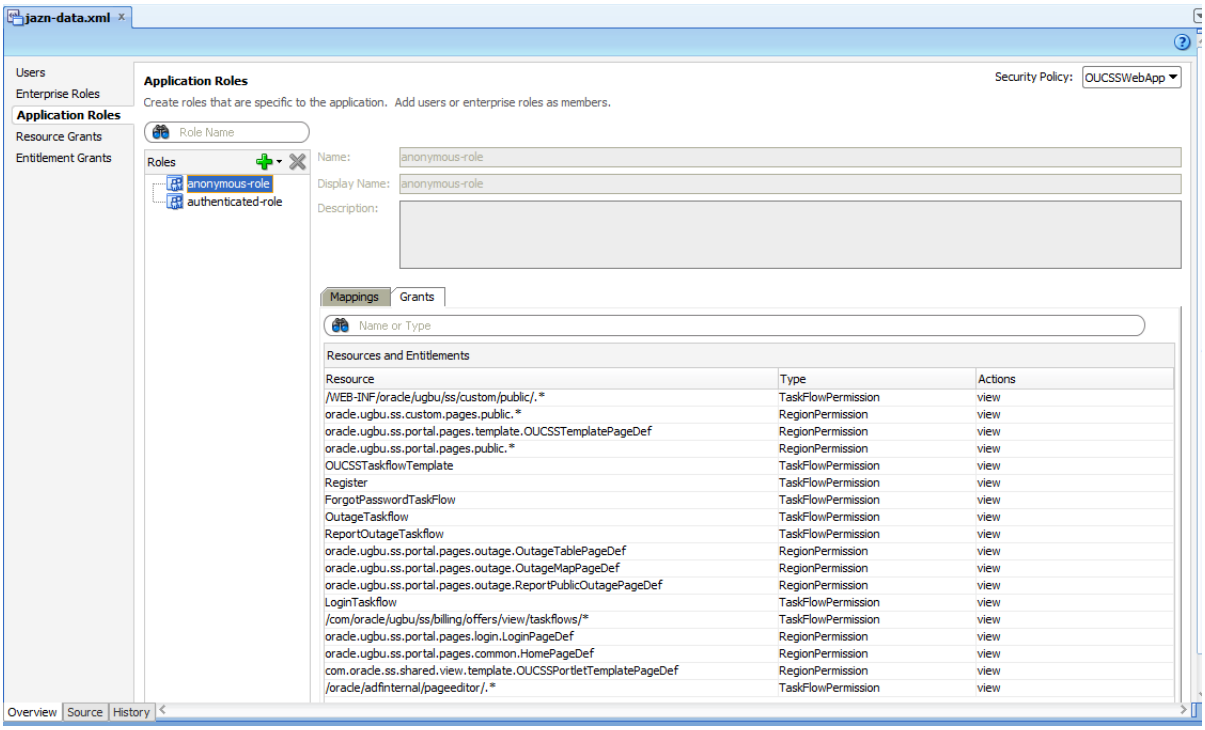
OUCSSWeb application are secured using the Oracle ADF Security framework. Oracle ADF Security is built on top of the Oracle Platform Security Services (OPSS) architecture.

Two Enterprise roles–WSSAdminGroup and WSSCSRGroup–are created to allow users belonging to these roles to have access to Admin and CSR modules.

Security configurations are stored in jazn-data.xml, web.xml, adf-config.xml, jps-config.xml, and weblogic.xml.

The following snapshot of jazn-data.xml shows the resources available to the authenticated user and public user. You can access it by right-clicking on OUCSSWebApp in **Application navigator>Secure>Application Roles**.

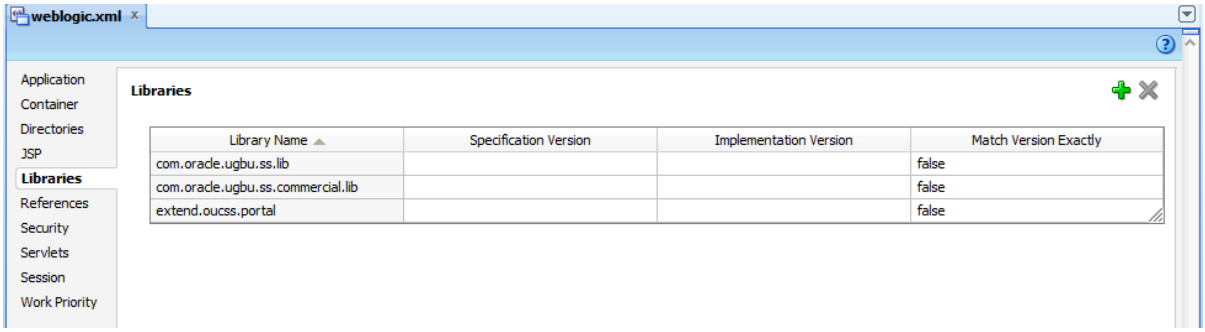



**Note**: For more information on enabling security in an ADF application, see "Enabling ADF Security in a Fusion Web Application" in the *Oracle® Fusion Middleware Web User Interface Developer's Guide for Oracle Application Development Framework documentation.*

# Testing and Deployment

All testing and deployment requires that WEB-INF/weblogic.xml contains references to the com.oracle.ugbu.ss.lib, com.oracle.ugbu.ss.commercial.lib, and extend.oucss.portal shared libraries, as shown in the following image.



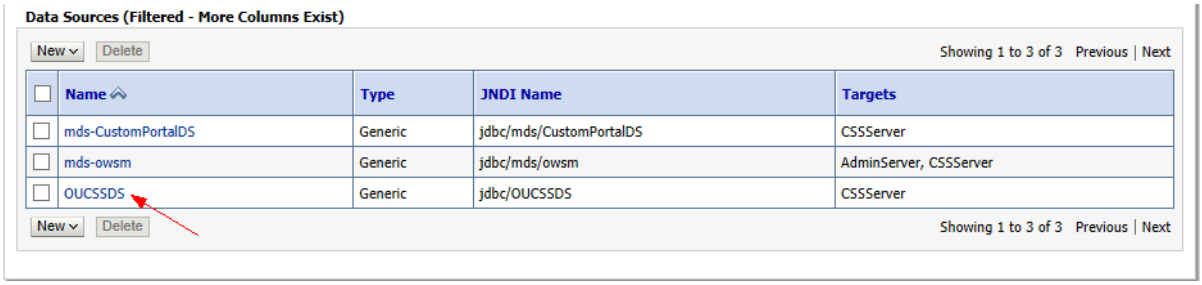

# Testing the Integrated WebLogic Server in JDeveloper

## Pre-testing Checklist

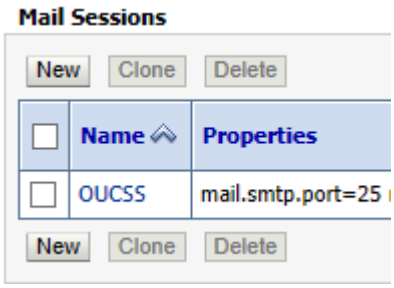
Before running on the Integrated WebLogic server, check that all of the following files, libraries, datasources, mail sessions, keys, and users are in place.

- Ensure that the following shared libraries are deployed on the Integrated WebLogic server:
  - <OUCSS_Product_home>>\ Install\application\OUCSS_Extension.war
  - <OUCSS_Product_home>>\ Install\application\OUCSS_Commercial_Extension.war
  - <OUCSS_Product_home>>\ Install\application\extend.oucss.portal.war

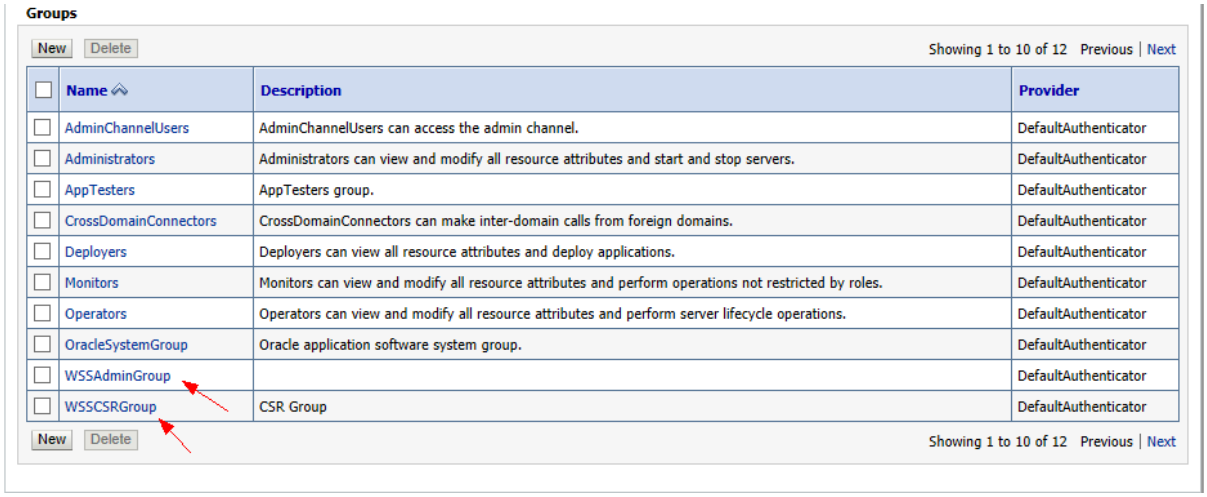
  **Note**: If any new extensions are added to OUCSSPortalADFLibrary project of ExtendOUCSSPortal application, then a new extend.oucss.portal.war must be generated and deployed. For details see the procedure in the "Deploying the extend.oucss.portal.war as Shared Library" chapter of the *Customizing and Extending OUCSS Custom Portals* whitepaper available at http://www.oracle.com/technetwork/apps-tech/utilities/documentation/index.html.
- Check that the OUCSSDS datasource is created and that it points to the OUCSS schema where the OUCSS tables are created.




- Check that the OUCSS Mail Session is created.




- Check that the WSSAdminGroup and WSSCSRGroup groups are created



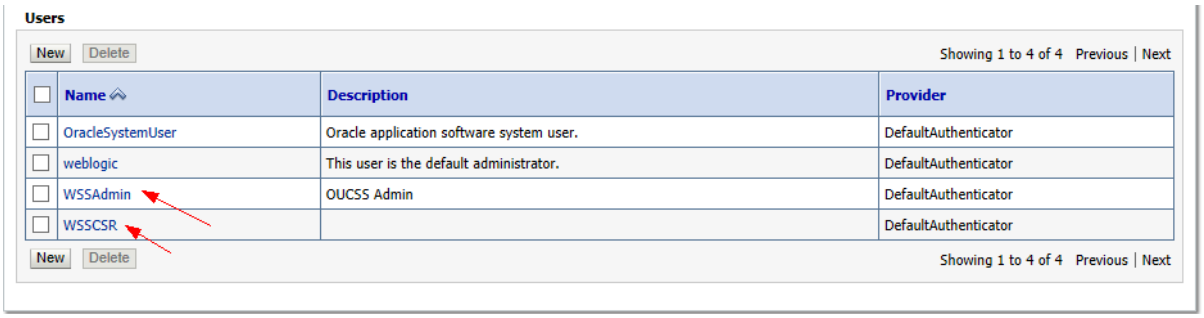
**Groups**

New | Delete — Showing 1 to 10 of 12 Previous | Next

| Name | Description | Provider |
|---|---|---|
| AdminChannelUsers | AdminChannelUsers can access the admin channel. | DefaultAuthenticator |
| Administrators | Administrators can view and modify all resource attributes and start and stop servers. | DefaultAuthenticator |
| AppTesters | AppTesters group. | DefaultAuthenticator |
| CrossDomainConnectors | CrossDomainConnectors can make inter-domain calls from foreign domains. | DefaultAuthenticator |
| Deployers | Deployers can view all resource attributes and deploy applications. | DefaultAuthenticator |
| Monitors | Monitors can view and modify all resource attributes and perform operations not restricted by roles. | DefaultAuthenticator |
| Operators | Operators can view and modify all resource attributes and perform server lifecycle operations. | DefaultAuthenticator |
| OracleSystemGroup | Oracle application software system group. | DefaultAuthenticator |
| WSSAdminGroup | | DefaultAuthenticator |
| WSSCSRGroup | CSR Group | DefaultAuthenticator |

New | Delete — Showing 1 to 10 of 12 Previous | Next

- Check that WSSAdmin and WSSCSR users are created.



**Users**

New | Delete — Showing 1 to 4 of 4 Previous | Next

| Name | Description | Provider |
|---|---|---|
| OracleSystemUser | Oracle application software system user. | DefaultAuthenticator |
| weblogic | This user is the default administrator. | DefaultAuthenticator |
| WSSAdmin | OUCSS Admin | DefaultAuthenticator |
| WSSCSR | | DefaultAuthenticator |

New | Delete — Showing 1 to 4 of 4 Previous | Next

- Check that the CSF-Keys are created on the WebLogic server:
  - OUCSS_XAI_BASIC_KEY - To connect to CCB web services
  - OUCSS_INTG_BASIC_KEY - To connect to BPEL web services
  - OUCSS_OUNC_BASIC_KEY - To connect to Notification web services

  If any or all of the keys are missing, they can be created on an Integrated WebLogic server through wlst commands:

  - Connect to Integrated WebLogic server with the following wlst command:

```
connect("<<username>>","<<password>>","http://<<IntegratedWebLogic server>>:<<port>>")
```

  - Run following wlst commands for each key:

```
createCred(map='oracle.wsm.security', key='<key name>', user='<user name>',
password=<password>')
```

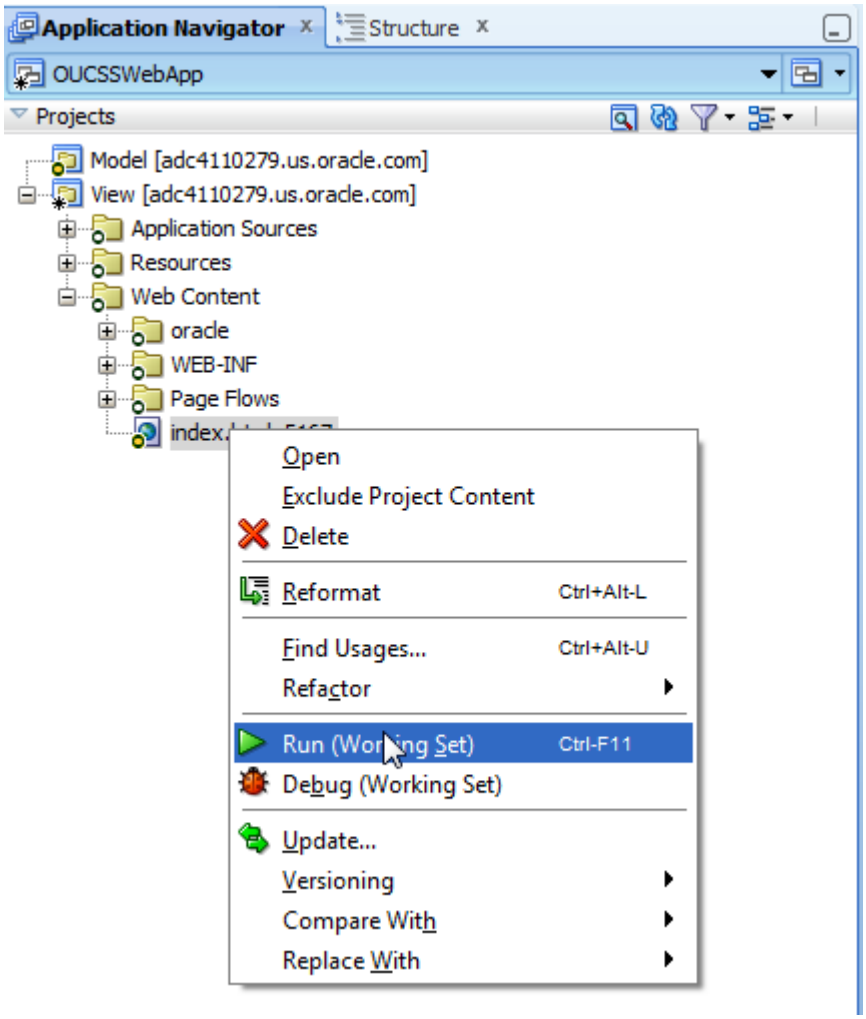
- Check that connections.xml is detokenized to point to the respective CCB and SOA servers.

| Tokens | Interpretation |
|---|---|
| OUNMSHOST:OUNMSPORT | SOA server name and port that host BPEL service that connects to NMS |
| OUMDMHOST: OUMDMPORT | SOA server name and port that host BPEL service that connects to MDM |
| OUNCHOST:OUNCPORT | SOA server name and port that host BPEL service related to Notifications |
| CCBorSOAHOST:CCBorSOAPORT | CCB server name and port or SOA server name and port to |

| | |
|---|---|
| | connect to optional BPEL flows |
| CCBorNMSHOST: CCBorNMSPORT | CCB server name and port or SOA server name and port that host BPEL service that connects to NMS |
| mapViewerHOST:mapViewerPORT | Map View connection |
| OfferServiceConditionHost:OfferServiceConditionPort | Server name and port where Offers service is deployed |

# Running the Test

In JDeveloper, open the OUCSSWebApp project by right-clicking on index.html and choosing **Run**.




The test is successful if the default browser page opens:

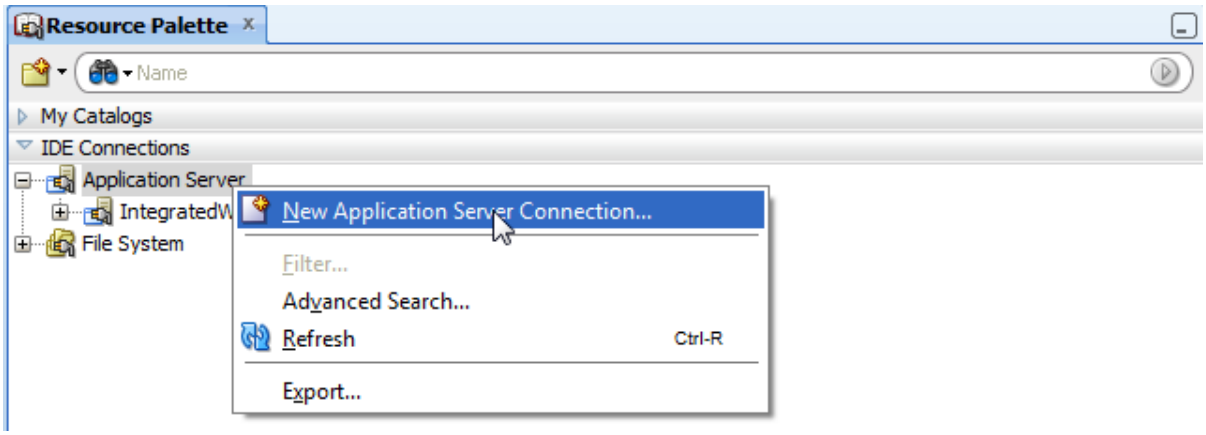
# Deploying to a Standalone WebLogic Server Using JDeveloper

The EAR file can also be deployed to a Standalone WebLogic server from JDeveloper.

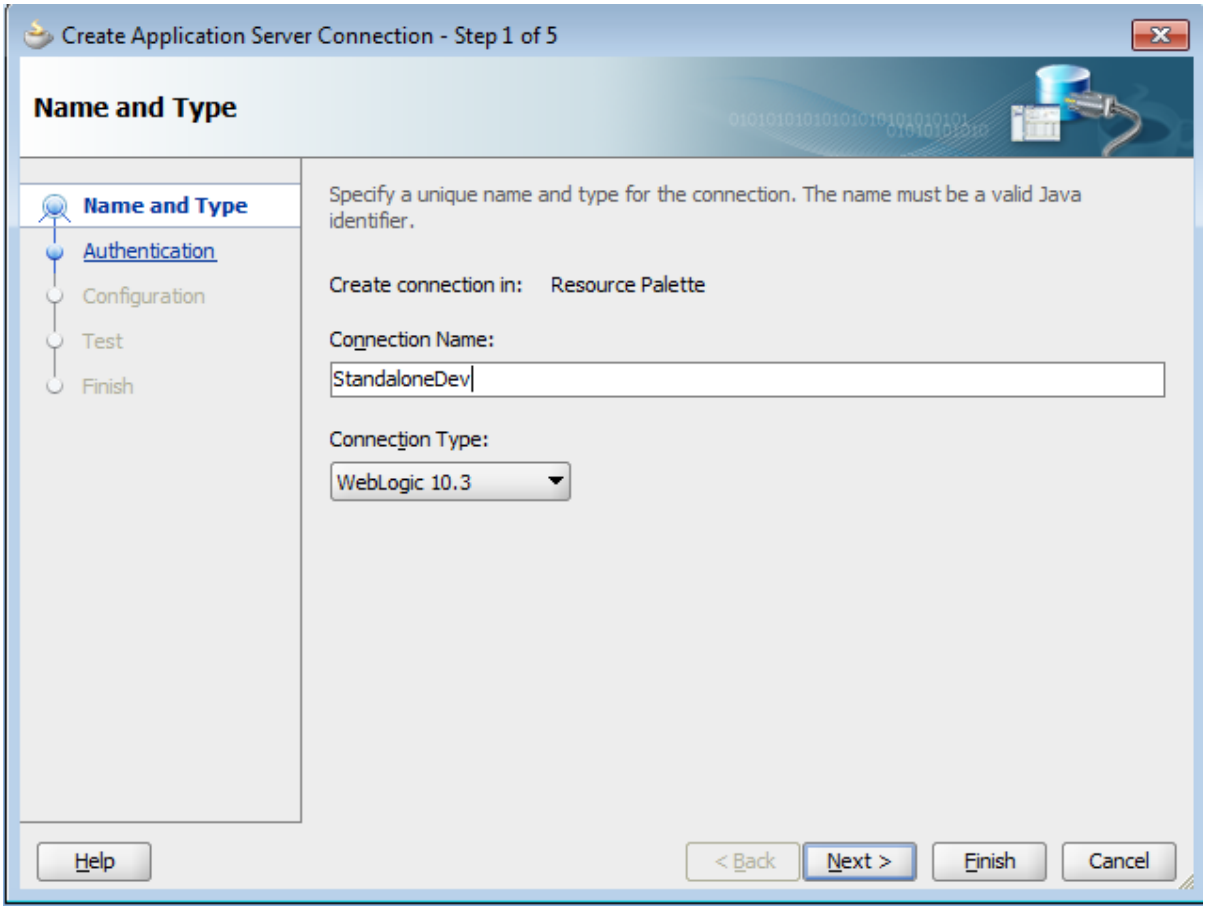
To faciliate deployment a number of required artifacts must be installed on the Standalone WebLogic server. For a checklist of the artifacts, see the "Pre-testing Checklist" topic elsewhere in this document.

**To deploy to a standalone WebLogic server:**

1. Ensure that the WC_CustomPortal managed server is up and running.
2. In JDeveloper,open the Resource Palette window. Under IDE Connections, create a new connection to your Standalone WebLogic server.



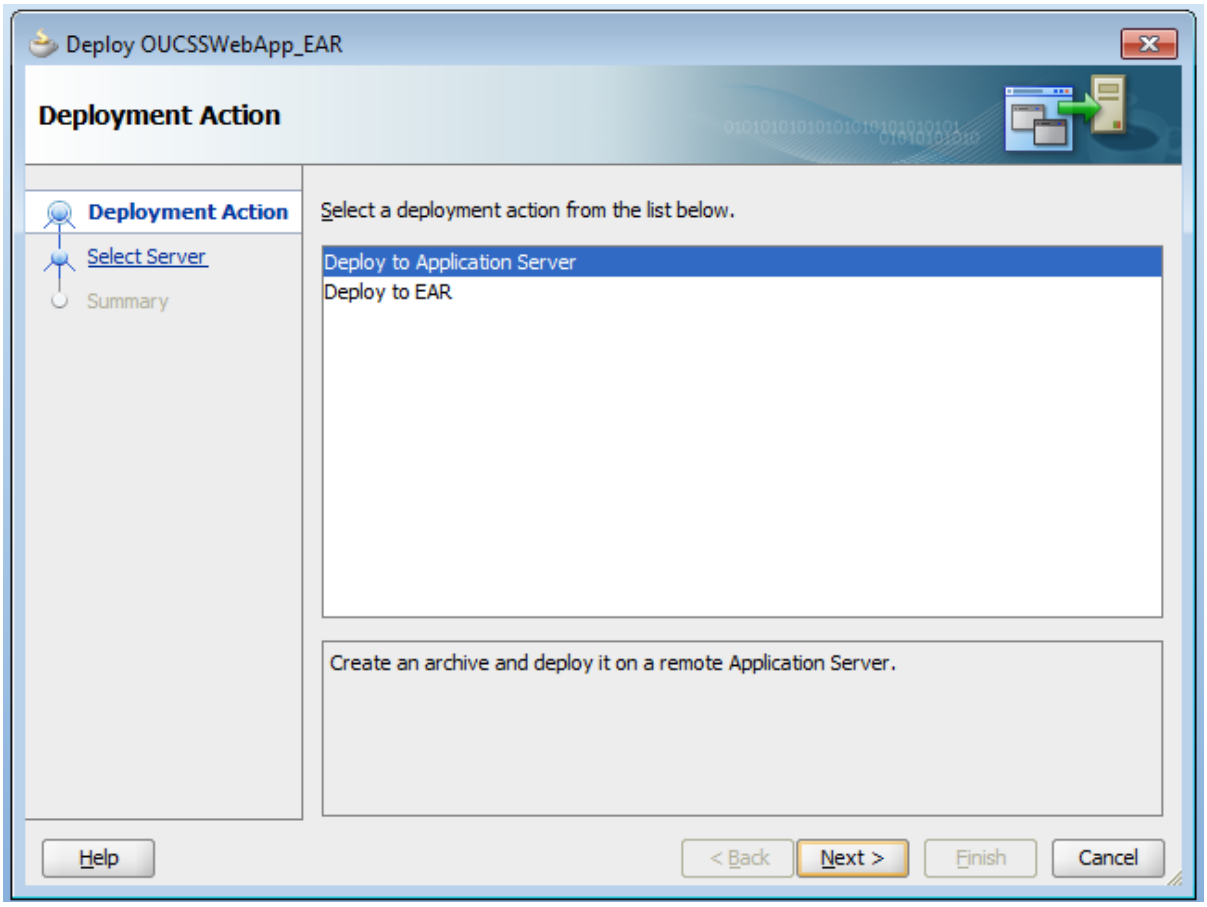

A wizard opens.



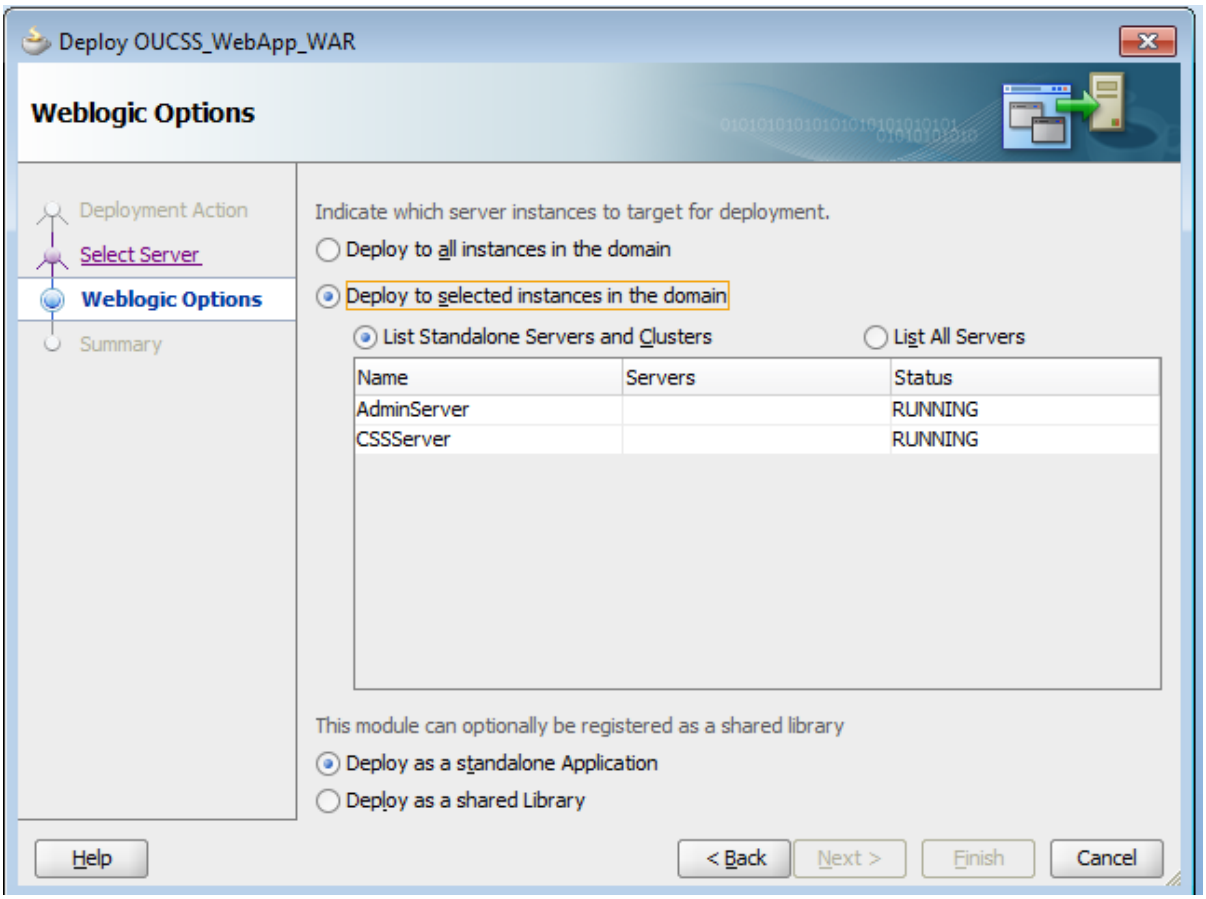

3 Follow the wizard to provide the appropriate connection information for the Standalone WebLogic server. In this example, the connection name is **StandaloneDev**.

4 Test the connection:

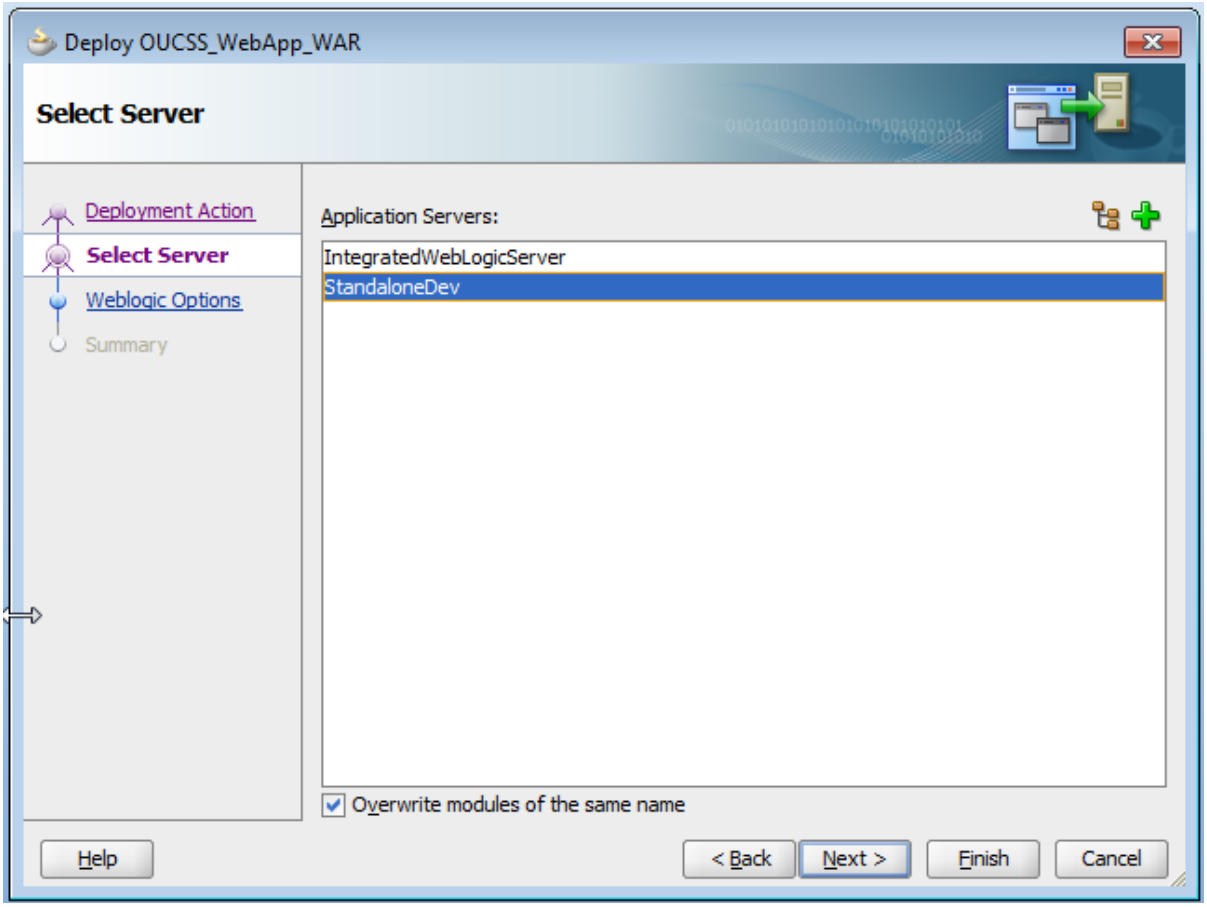
5 In the JDeveloper Application Navigator, right-click the application name and choose **Deploy>Select the deployment profile>Deploy to Application Server>Next**.




**6** Select the name of the connection ("StandaloneDev" in this example) to deploy the application.




**7** Test the application by invoking http://<<Standalonehost>>:<<StandalonePort>>/OUCSSWebApp.

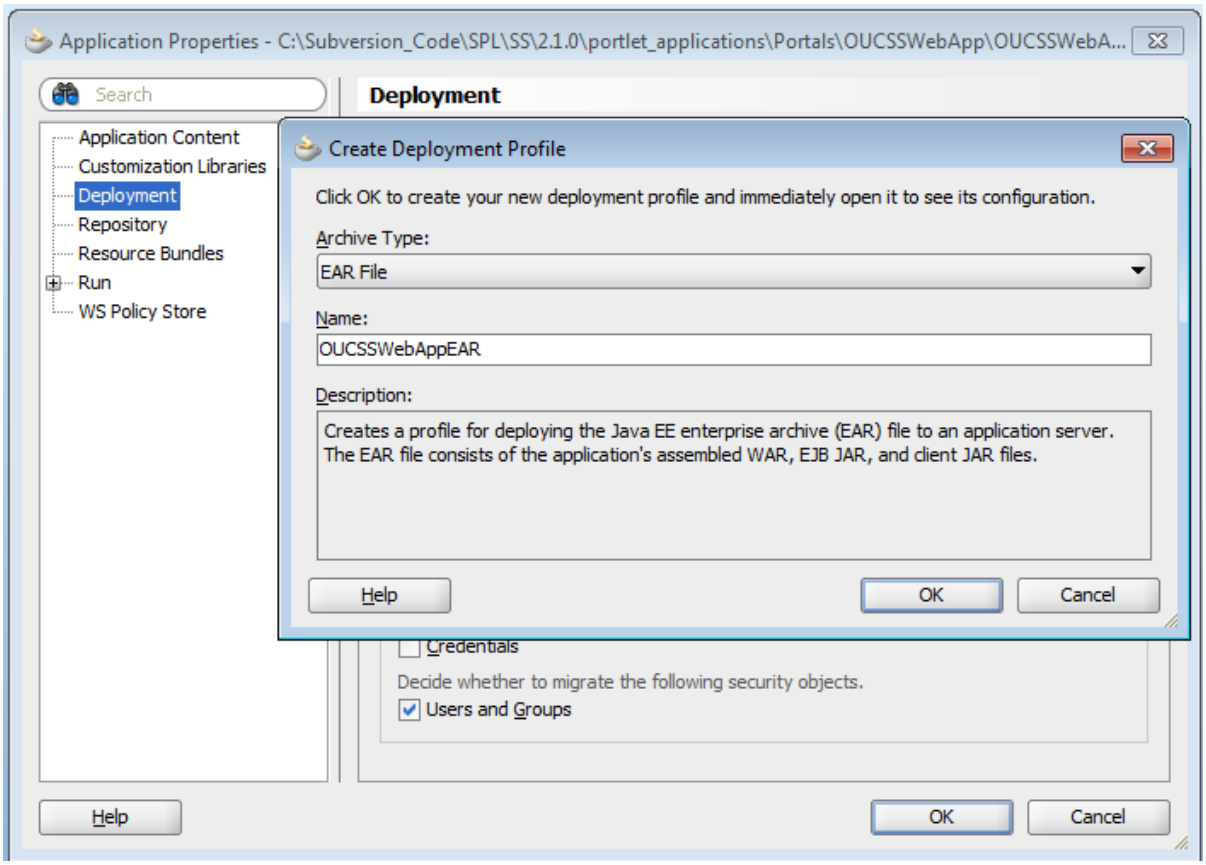


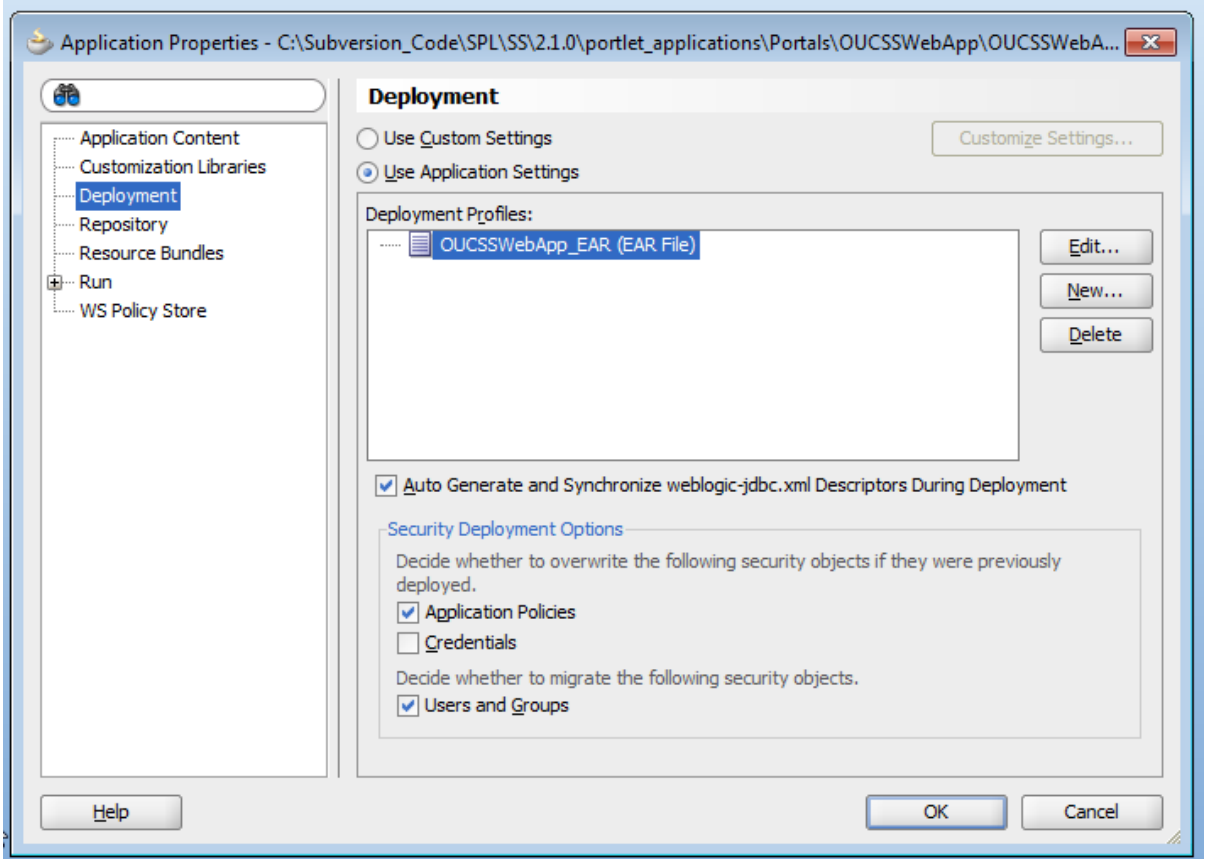

# Deploying to a Standalone WebLogic Server Through WebLogic Enterprise Manager (EM)

OUCSS Web Application comes with a default deployment profile named OUCSSWebApp_EAR. If any modifications are made to the OUCSSWebApp, this deployment profile can be used to generate the ear archive file for deployment on an application.

You can also create a new deployment profile as follows:

1. Right click on **OUCSSWebApp>Application Properties>Deployment>New**.
2. Select **EAR file** as the **Archive Type**, enter the name of the archive, and click **OK**.


Application Properties - C:\Subversion_Code\SPL\SS\2.1.0\portlet_applications\Portals\OUCSSWebApp\OUCSSWebA...
Search
Application Content
Customization Libraries
Deployment
Repository
Resource Bundles
Run
WS Policy Store
Deployment
Create Deployment Profile
Click OK to create your new deployment profile and immediately open it to see its configuration.
Archive Type:
EAR File
Name:
OUCSSWebAppEAR
Description:
Creates a profile for deploying the Java EE enterprise archive (EAR) file to an application server. The EAR file consists of the application's assembled WAR, EJB JAR, and client JAR files.
Help
OK
Cancel
Credentials
Decide whether to migrate the following security objects.
Users and Groups
Help
OK
Cancel




**Note**: See Deploying Fusion Applications in the *Oracle® Fusion Middleware Fusion Developer's Guide for Oracle Application Development Framework* documentation for additional information on creating deployment profiles and deploying to Application servers.

Deploying on a Standalone WebLogic server through EM assumes required artifacts and a managed server are already installed as described in Installing on a WebLogic Server.

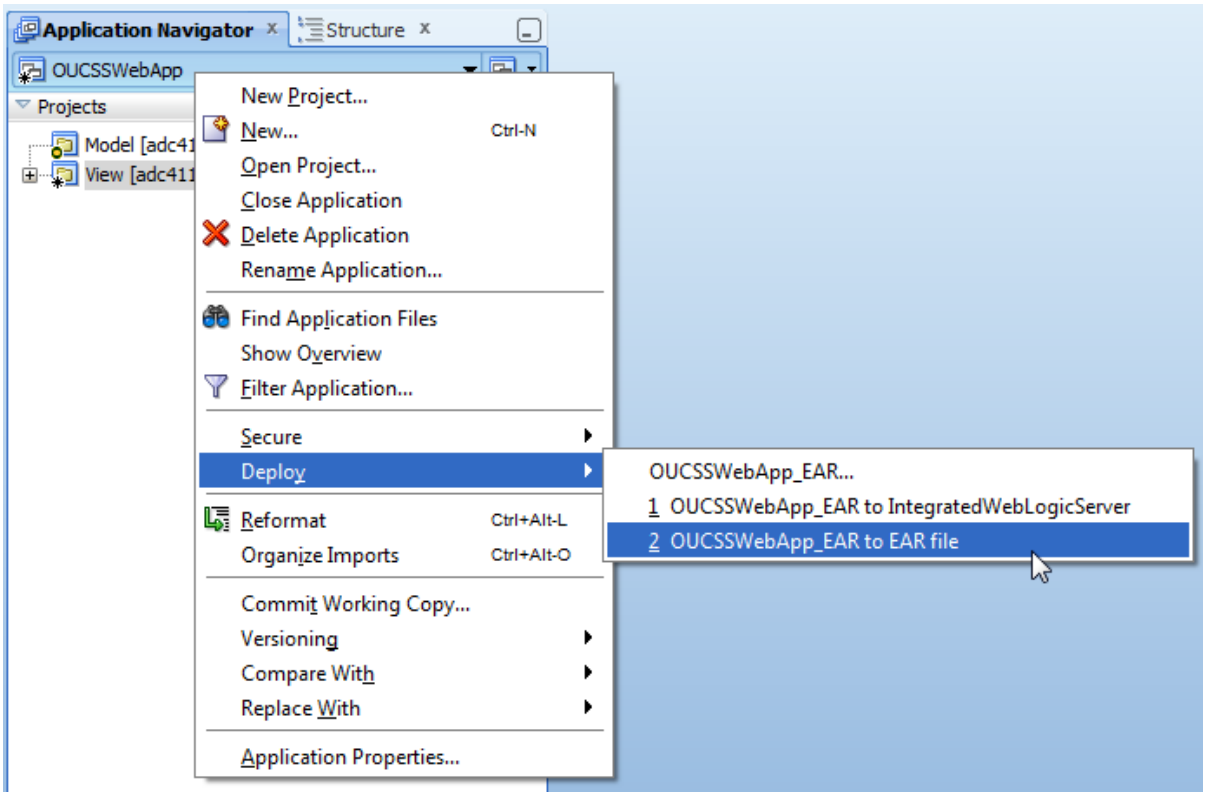
For a checklist of the artifacts, see the "Pre-testing Checklist" topic in this document.

**Note**: If any new extensions are added to the OUCSSPortalADFLibrary project of ExtendOUCSSPortal application, a new extend.oucss.portal.war must be generated and deployed. See the steps in the chapter "Deploying the extend.oucss.portal.war as Shared Library" chapter in the *Customizing and Extending OUCSS Custom Portals* whitepaper, available in the OUCSS section of the OTN web site at http://www.oracle.com/technetwork/apps-tech/utilities/documentation/index.html.

3 Deploy to an EAR archive file as follows:

   A Select OUCSSWebApp application from the Application Navigator.

   B Right-click on **OUCSSWebApp>Deploy>OUCSSWebApp_EAR**.




4. Ensure that the WC_CustomPortal managed server is up and running.
5. Log in to WebLogic Server EM with admin credentials (wlsadminuser/wlsadminpw).
6. Expand **WebLogic Domain>portal_domain**.
7. Right click on **WC_CustomPortal(managed server)>Application Deployment>Deploy**.
8. Follow the wizard to deploy the generated EAR file, specifying the MDS partition names.
9. To test the application, invoke http://<<Standalonehost>>:<<StandalonePort>>/OUCSSWebApp.