

Running Web Dynpro Java Applications in Portal



Applies to:

SAP NetWeaver Composition Environment 7.1. For more information, visit the [Portal and Collaboration homepage](#).

Summary

This Article deals about Running Web Dynpro for Java Applications in Portal. It gives step by step procedure for creating portal Roles, Worksets, Webdynpro pages and navigation between these pages.

Author: Fazal Ahamed

Company: HCL Technologies Limited.

Created on: 22 December 2008

Author Bio

Fazal Ahamed is working as Java developer in HCL Technologies Limited on Sap NetWeaver Technology (CAF, Web Dynpro Java, and GP).

Table of Contents

Overview	3
Creating Web Dynpro Applications	5
Web Dynpro Applications in Portal	8
Creating a Folder in Portal Content:	9
Creating Web Dynpro Pages:	10
Create a Workset:	11
Create a Role:	13
Portal Navigation	13
Related Content.....	16
Disclaimer and Liability Notice.....	17

Overview

This Article deals about Running Web Dynpro for Java Applications in Portal. It gives step by step procedure for creating portal Roles, Worksets, webdynpro pages and navigation between these pages.

For this demo we will be creating the following PCD Objects in the Content Administrator->Portal Content

- 1) Folder-Test Folder
- 2) Role-Image
- 3) Workset-Image
- 4) Webdynpro Application Pages- Three pages from ImageApp, SunsetApp and WinterApp webdynpro Applications.

From the below figure we can visualize,

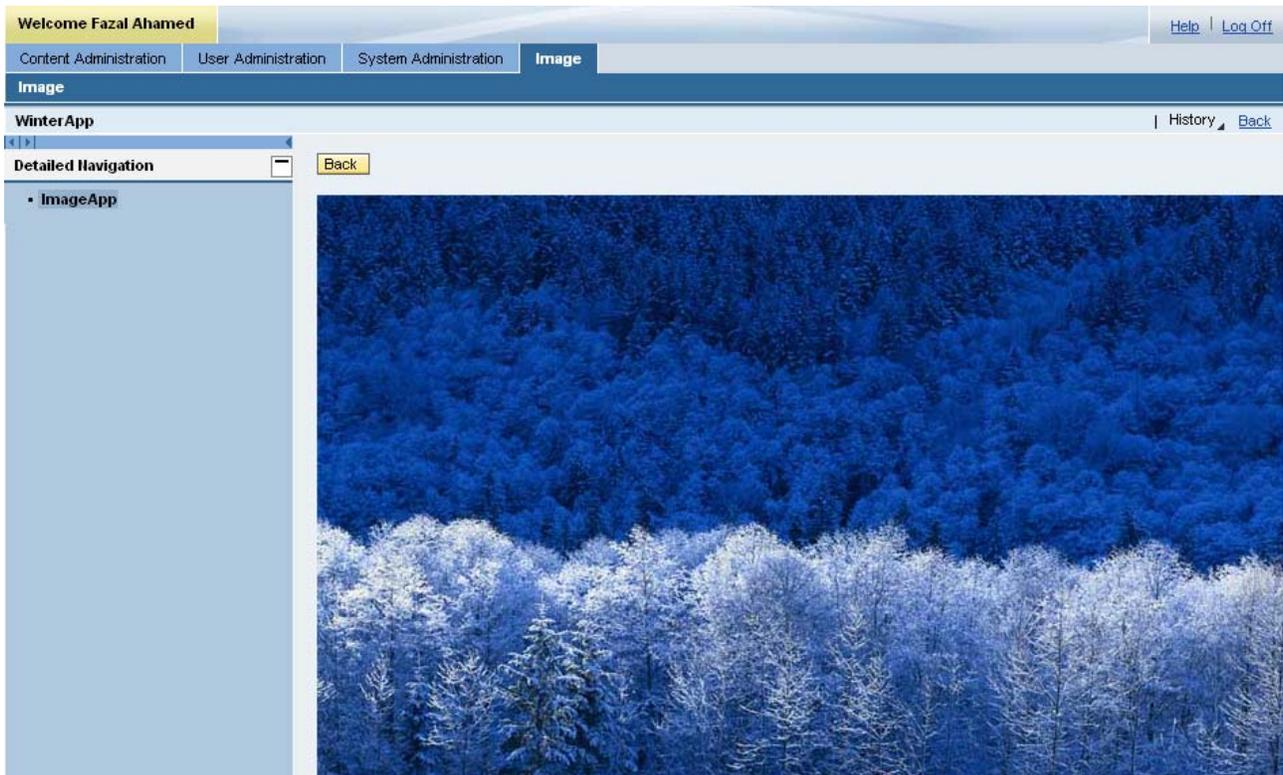
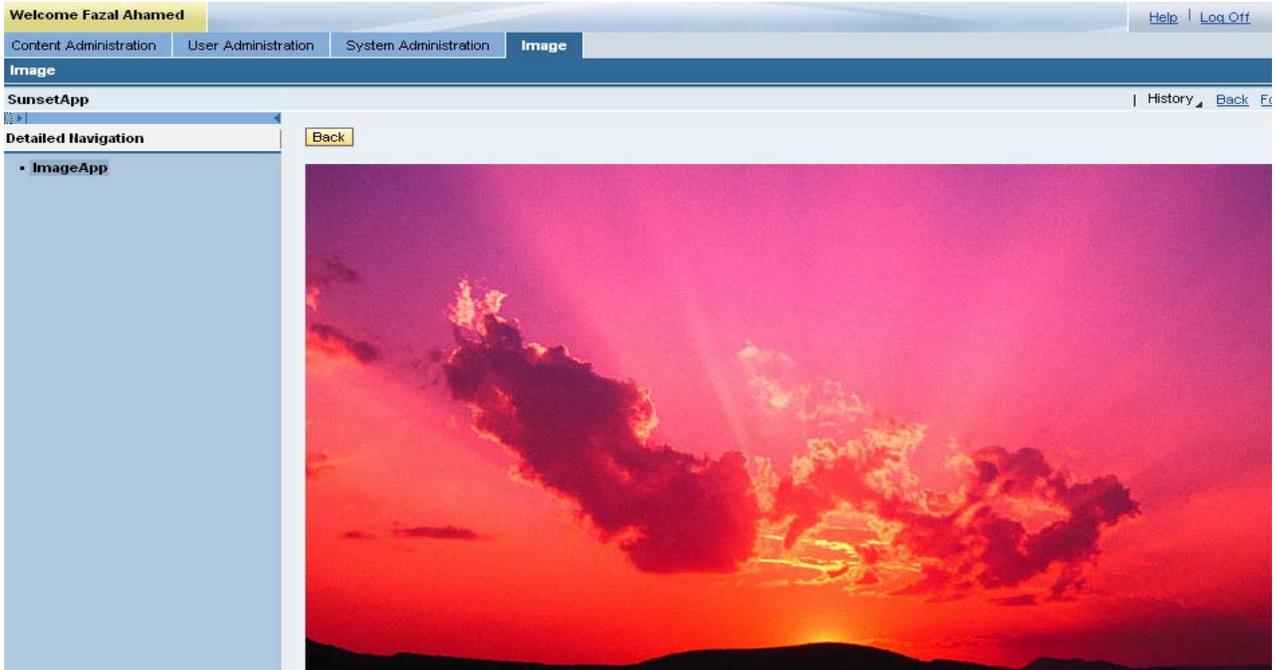
Created Role - Image

Created Worksets - Image

Created Page -ImageApp



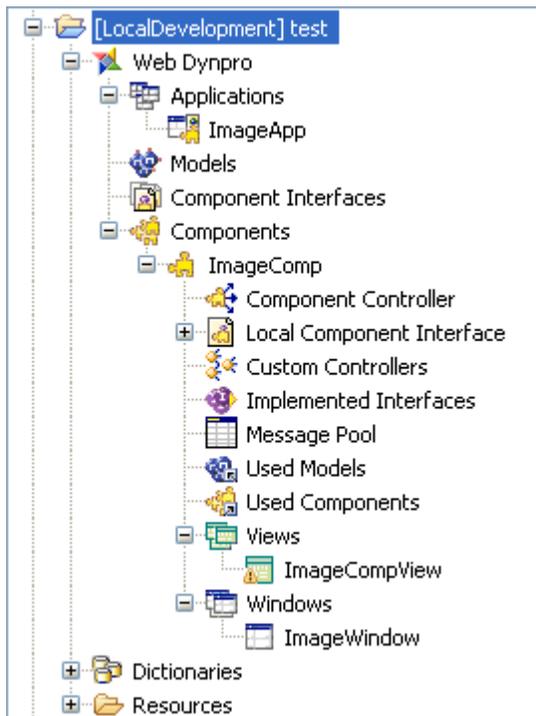
On selecting the Radio Button say “Sunset”, the SunsetApp Page is displayed through Portal Navigation APIs. You will find a “Back” Button for navigating from SunsetApp Page to ImageApp Page.



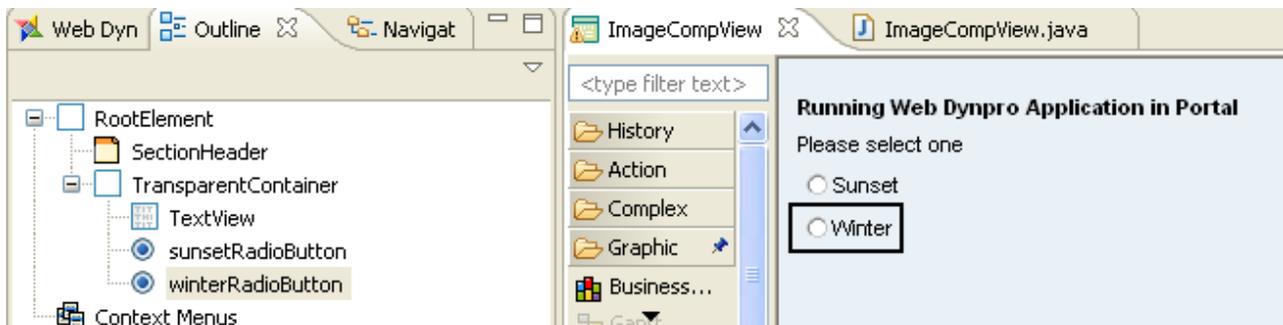
Procedure

Creating Web Dynpro Applications

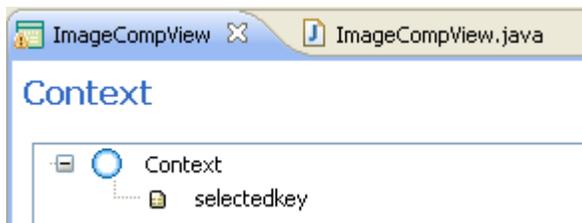
Create a Web Dynpro Project with name “test” as shown below.



Open the ImageCompView and add the UI Elements as shown below.



Select the Context tab and create a Context Attribute with name as “selectedkey” as shown below.



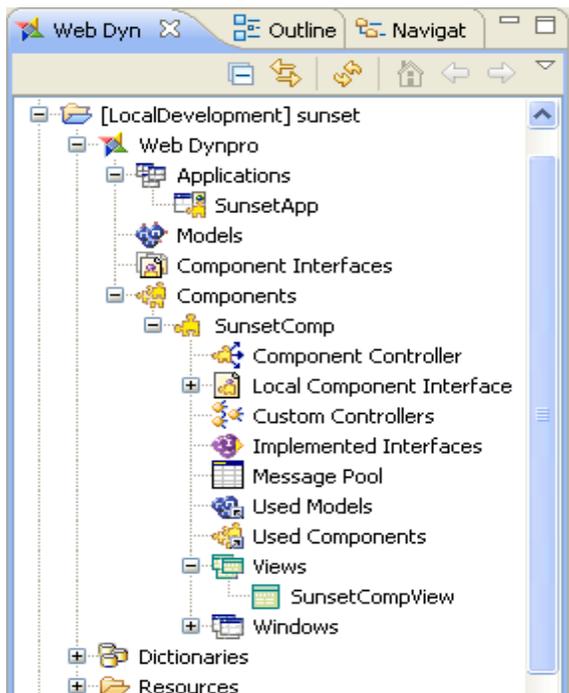
Set the keyToSelect property of sunsetRadioButton to “Sunset” and bind the selectedkey Property to the context Attribute “selectedkey”. Similarly set the keyToSelect property of winterRadioButton to “Winter” and bind the selectedkey Property to the context Attribute “selectedkey”.

Create an Action say “OnSelect” and set the property OnSelect to the Action “OnSelect” for winterRadioButton and sunsetRadioButton as shown below.

Property	Value
id	sunsetRadioButton
activateAccessKey	false
enabled	true
explanation	
keyToSelect	Sunset
readOnly	false
selectedKey	 selectedkey
state	normal*
text	Sunset
textDirection	inherit*
tooltip	
visible	visible*
Events	
onSelect	OnSelect

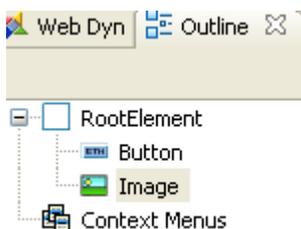
Finally build and deploy the application.

Create a New Web Dynpro Project with name “sunset” as shown below.



Copy the image to the Component –Src – mimes- fully qualified name of the component and set the source property for the image UI element as shown below.

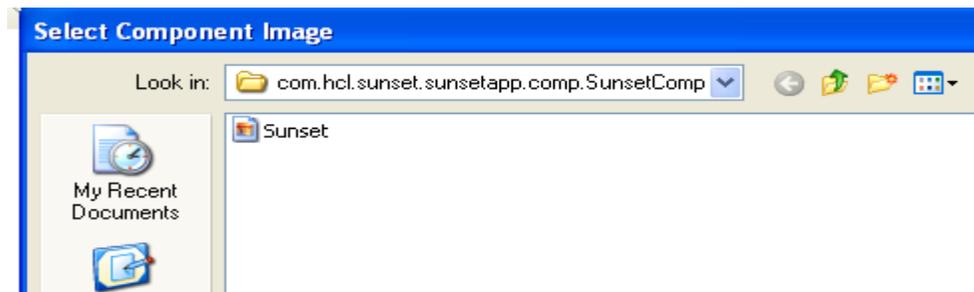
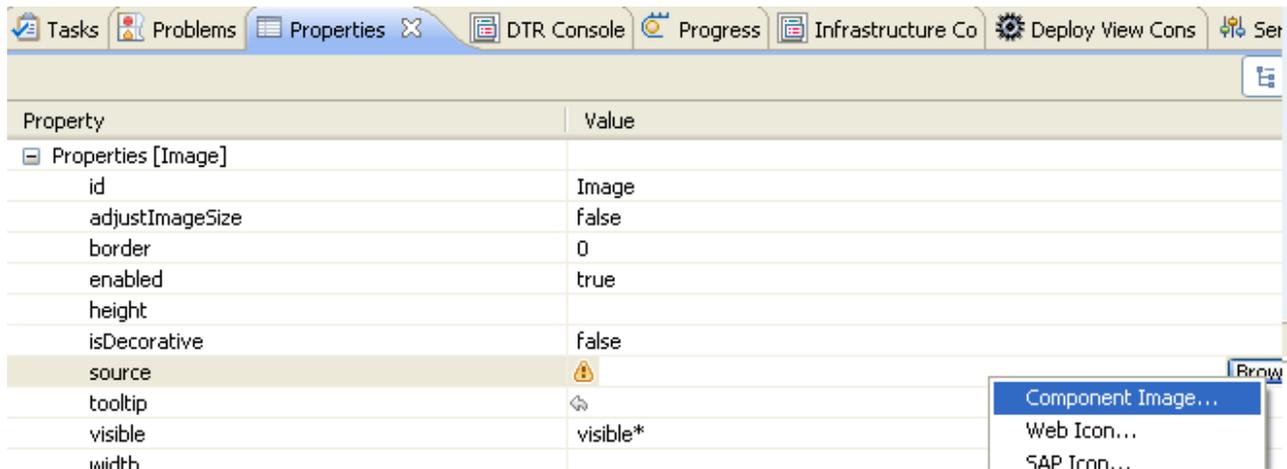
Open the View Editor of SunsetCompView; add button and Image UI Element as shown below.



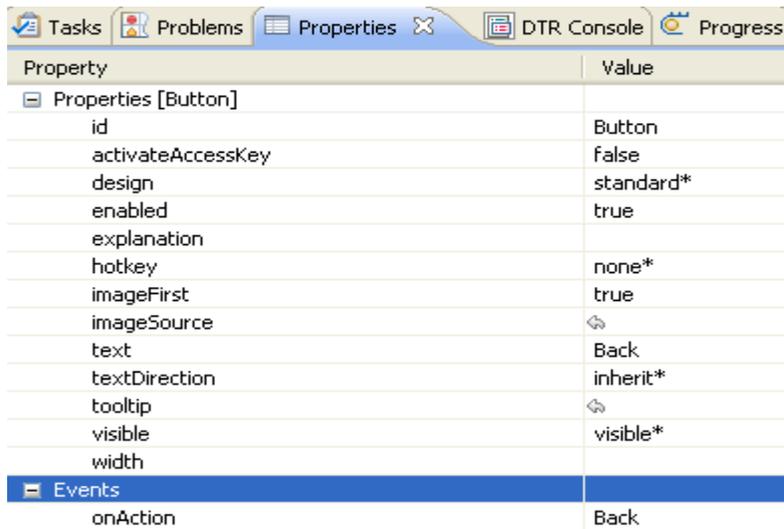
Click image UI element and set the source property of the image to the Sunset Image.

Click browse button as shown below.

Finally click on the component Image and select the Sunset Image present in your component.

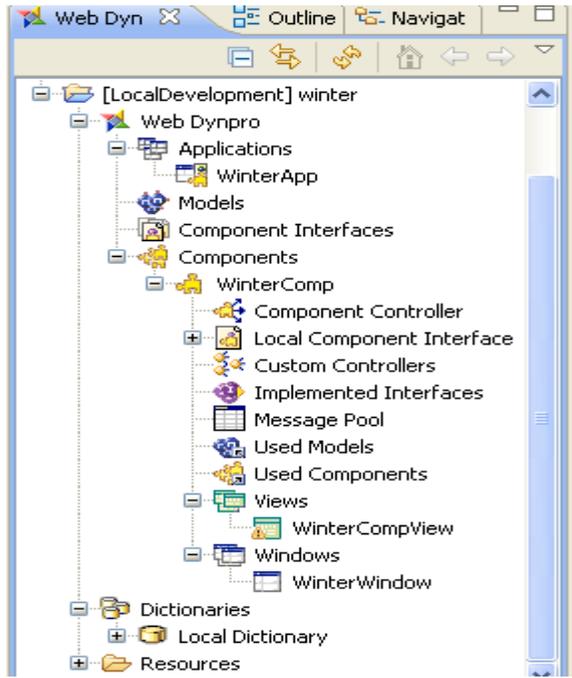


Create a Action say “Back” and bind this action to the onAction Property of Button UI Element.



Finally deploy your component.

In the Similar way, create a new Web Dynpro Project with name “Winter”. As shown below.



Follow the same steps done earlier for the **sunset** project.

Copy the image to the component src-mimes-fully qualified name of the component.

Open the View Editor of WindowCompView and add the Button and Image UI Element.

Create a Action say "Back" and bind it to the OnAction Property of Button UI Element.

Set the source property of Image UI Element to the Image, as done earlier for the Sunset Component.

Finally build and deploy the Web Dynpro Component.

Web Dynpro Applications in Portal

Portal is J2EE Applications, which is designed to handle the HTTP request and send back the response to the user. Portal has the components which are used to investigate on the request and send back the response.

Once we login to portal, u will able to see the roles assigned to you, say Guided procedures, Content Administration, User Administration etc. Role assignment is done in the User Administration by adding the specified role to the user.

In a Company, every user has access to information or data based on his permission level. For e.g.: CEO of the company has permission to access the entire confidential data about the company, whereas the employee of the company will have access to only a certain amount of data.

In order to ensure that user has access to information which is relevant to him is added to an object called **Roles** and in turn roles are added to the user. Roles have a folder hierarchy of content objects, such as Worksets, Pages and IViews.

Role can have many Worksets. Pages and IViews can be assigned directly to a role.

User can access the information which is relevant to him by assigning the appropriate role to him.

Worksets is a content object which has a group of pages and IViews. Worksets is a reusable entity which can be added to roles. Worksets contains the User Specific activities whereas the role has a group of Worksets in a proper structure.

IViews are the small logical packet which represents a Visual Application. Pages have one or several IViews (Integrated Views) in a proper layout.

Roles contain Worksets, Worksets contains Pages and Pages holds IViews.

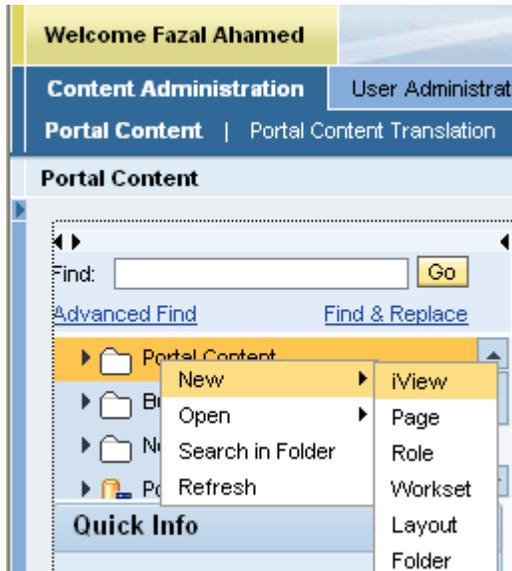
For detailed information, see [Roles, Worksets, Pages and IViews](#)

Creating a Folder in Portal Content:

Log into Portal (<http://nwserver:<port number>/irj/index.html>) and click the Content Administration Role → Portal Content.

First, create a new Folder where we can put our content objects.

Select the Portal Applications → Right Click --> New → Folder as shown below.



Give the following details for the folder:

Folder Name : Test Folder

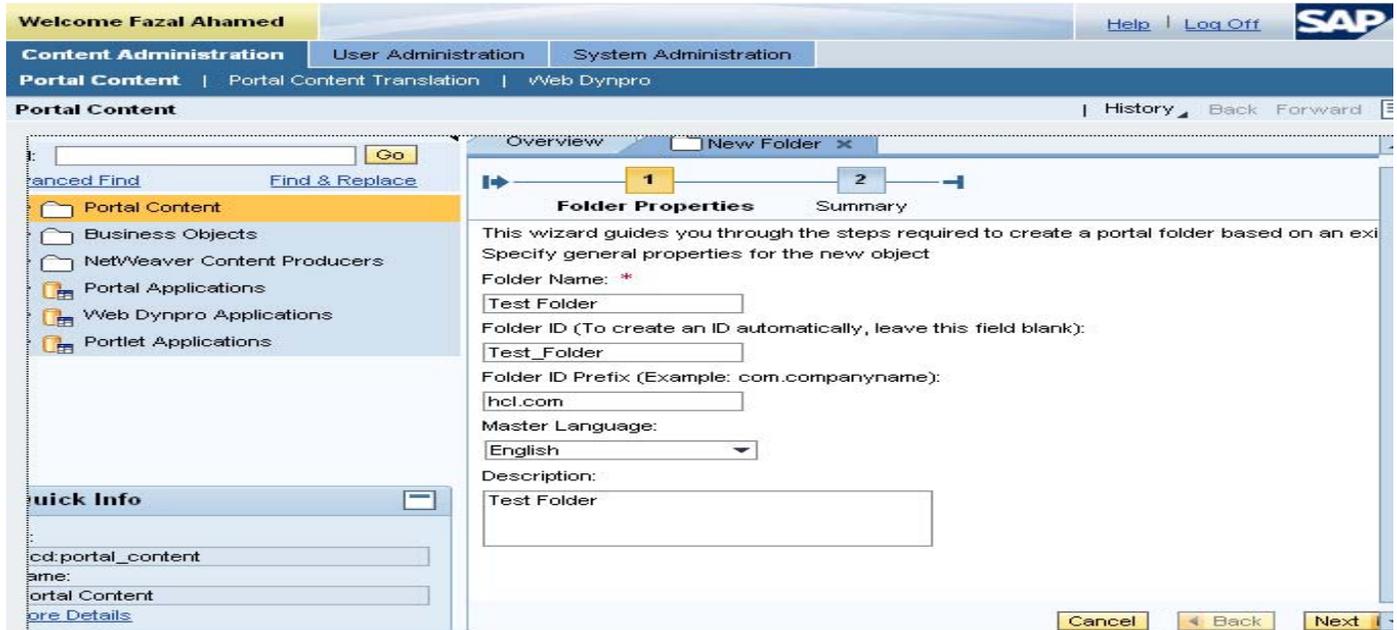
Folder Id : Test_Folder

Folder Id Prefix : hcl.com

Master Language : English

Description : Test Folder

Click next to view the summary of details about the Folder.



Click Finish. Now you will find the Test Folder in the Portal Content Folder.

Creating Web Dynpro Pages:

In the Portal Content, you will find the deployed Web Dynpro Applications.

Expand the Web Dynpro Applications content and find out the deployed Development Component “test”.

Select the Application – Right Click on it →Copy.

Go to the Portal Content folder and select Test Folder which we have created .

Right Click on it →Paste as PCD Object.

Give the Following Details:

Page Name : ImageApp Page Id : Image_App

Page ID Prefix : hcl.com Master Language : English

Description : Image App

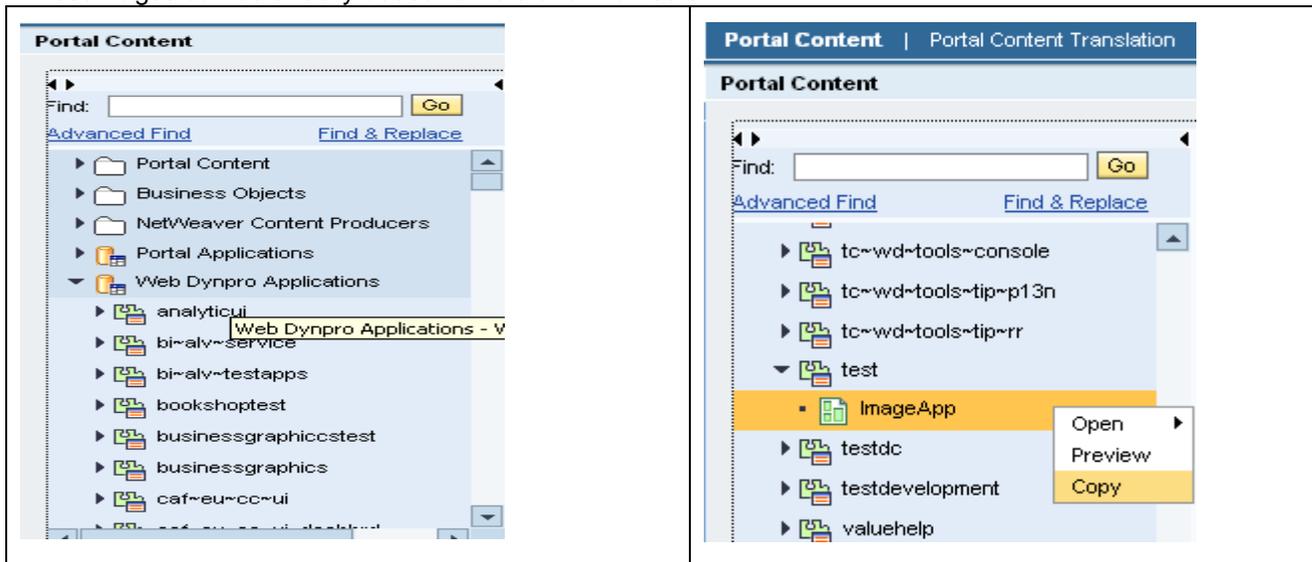
Click next and you will find the detailed summary of new created Page, finally click Finish.

Now the Web Dynpro Application is copied to our Test Folder as Page shown below.

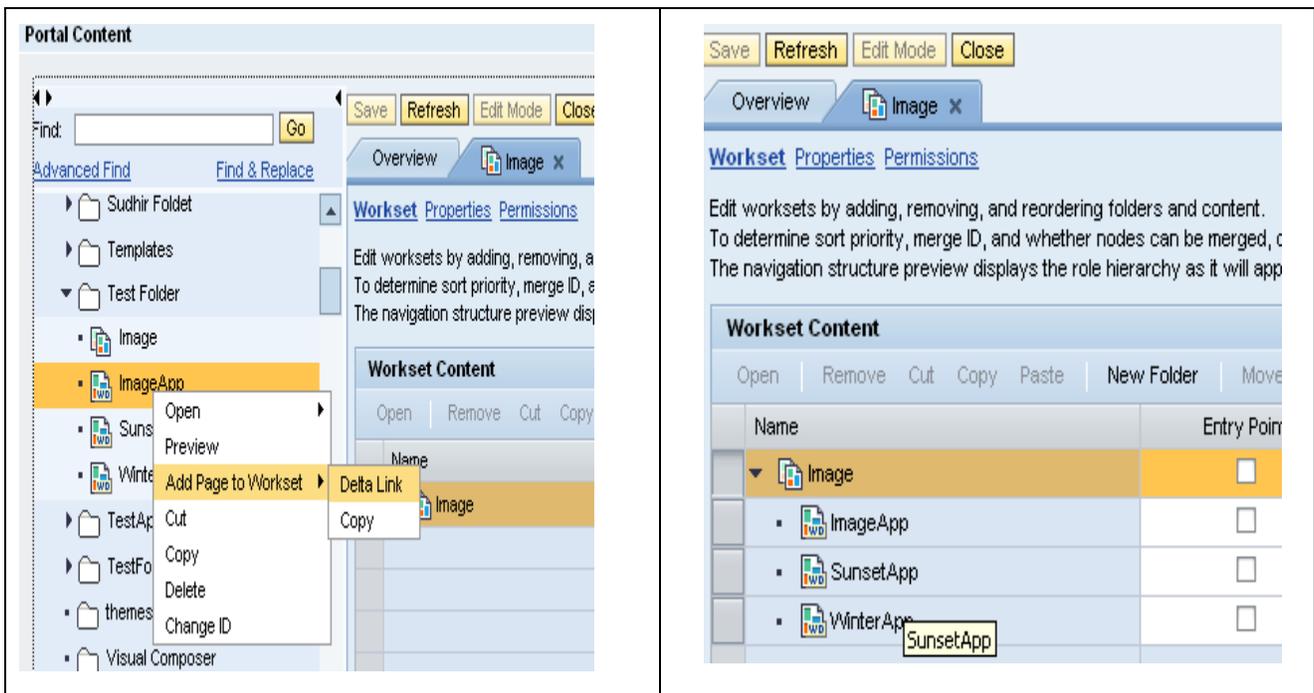
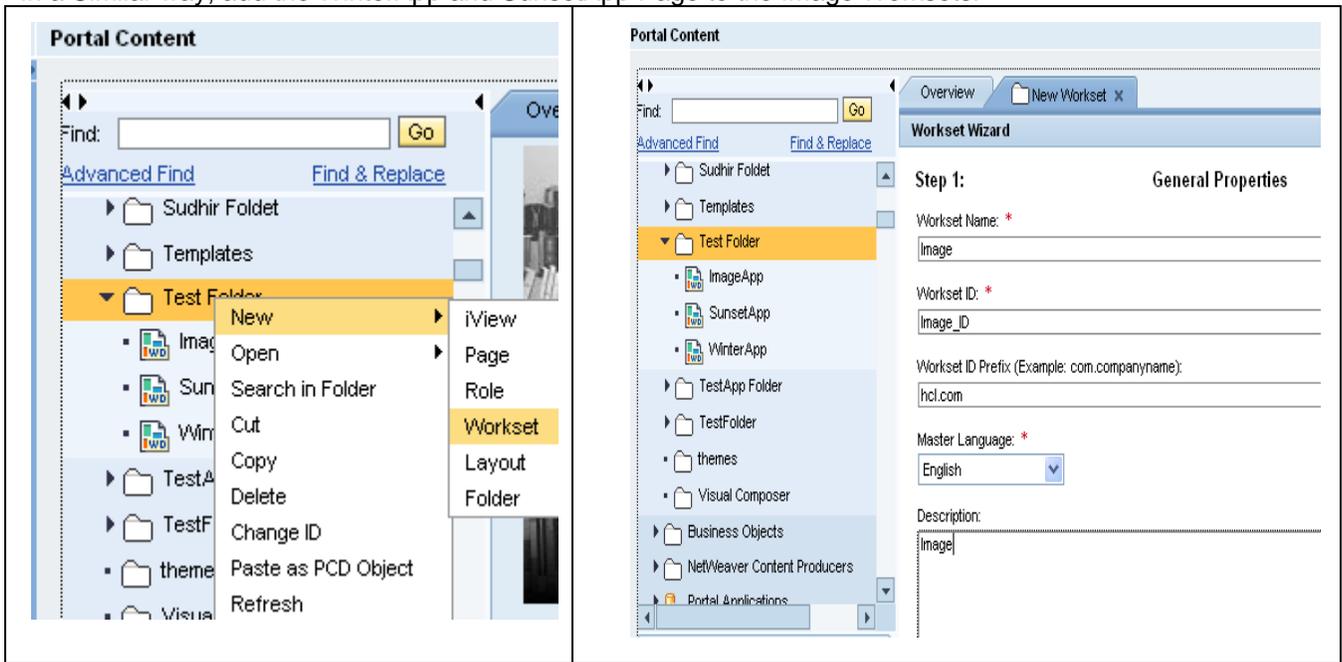
In the Similar way, copy the SunsetApp and WinterApp into the Test Folder.

If we want to see the preview of our Web Dynpro pages, right click on the ImageApp – preview.

These Pages can be directly added to Role or a Worksets.



In a Similar way, add the WinterApp and SunsetApp Page to the Image Worksets.



For more information on Object Based Navigation, please refer this link:

http://help.sap.com/saphelp_crm52/helpdata/EN/b6/a45b42bcbfd242e1000000a1550b0/frameset.htm

Now coming back to our project, to navigate from the Image App to SunsetApp or Winter App we need the URL (full path of PCD Object) of the Pages.

This URL can be get from the Quick Info or from the Location as shown below.

Name	Value
Changed By	Fistae
com.sap.portal.private.page.PersonalizationComponent	pcd:portal_content/every_user/general/personalization
Created By	Fistae
Created On	11/14/2008
Description	SunsetApp
ID	Sunset_App
ID Prefix	hcl.com
Last Date Changed	11/14/2008
Location	portal_content/hcl.com.Test_Folder/hcl.com.Sunset_Ap
Master Locale	en
Name	SunsetApp
Object is a Template	No

We can trigger the navigation using `WDPortalNavigation.navigateAbsolute ()`. This method performs navigation to an absolute defined navigation target and it takes the following parameters.

Navigation target: the absolute navigation URL of the Target. URL is the PCD Address of the Object (IViews or Page) with prefix "**ROLES: //**".

Navigation Mode: Mode in which the navigation has to be performed. The navigation modes are

- 1) Show in Place: Navigation target shown in same browser window.
- 2) Show external : target displayed in new browser window without portal framework i.e. just the views or page
- 3) Show external portal: target displayed in a new Browser window with portal framework.

History mode: how the navigation target occurs in the navigation history. There are three modes described below.

- 1) No history: the navigation target is not visible in the page history and we cannot use the back / forward to navigate back.
- 2) No Duplications: Navigation target is visible in the navigation history. Duplications are not allowed, if there is navigation to same target, and then it is displayed only once.
- 3) Allow Duplications: Navigation Target is visible in the navigation history and if the target is called more than once with different parameters, it is shown as different entries in the navigation history.

targetTitle: set the title of the page in the navigation history.

contextUrl: the Used navigation context(optional)

windowFeatures : Specifies the browser window features, such as window size and position .

windowName : set the name of the browser window , when the navigation target is called.

businessParameters : Url parameters which is used by the target IViews.

useSAPLauncher : Boolean value to specify whether the navigation target uses Sap launcher.

postParameters : defines whether the defined parameters are POST or GET Parameters.

Open the ImageCompView Java Editor; write the code for the Portal Navigation as shown below.

```

ImageCompView.java X
*           generic event object provided by framework
*/
/**end
public void onActionOnSelect (com.sap.tc.webdynpro.progmodel.api.IWDCustomEvent wdEvent )
{
    /**begin onActionOnSelect (ServerEvent)
        if ("Sunset".equals(wdContext.currentContextElement().getSelectedkey())) {
            WDPortalNavigation
                .navigateAbsolute(
                    "ROLES://portal_content/hcl.com.Test_Folder/hcl.com.Sunset_App",
                    WDPortalNavigationMode.SHOW_INPLACE, "", "",
                    WDPortalNavigationHistoryMode.NO_DUPLICATIONS, "",
                    "", "", "", false, true);
        } else {
            WDPortalNavigation
                .navigateAbsolute(
                    "ROLES://portal_content/hcl.com.Test_Folder/hcl.com.Winter_App",
                    WDPortalNavigationMode.SHOW_INPLACE, "", "",
                    WDPortalNavigationHistoryMode.NO_DUPLICATIONS, "",
                    "", "", "", false, true);
        }
    }
    /**end
}

```

In the Similar way, to navigate back to the Image App, we need the Correct Path of the Image App Page.

Follow the above mentioned procedure for getting the correct location of the Object.

In the OnAction Back Event Handler method, write the below code for both Sunset and Winter Component for navigating back to image component.

```

public void onActionBack (com.sap.tc.webdynpro.progmodel.api.IWDCustomEvent wdEvent )
{
    /**begin onActionBack (ServerEvent)
        WDPortalNavigation.navigateAbsolute(
            "ROLES://portal_content/hcl.com.Test_Folder/hcl.com.Image-App",
            WDPortalNavigationMode.SHOW_INPLACE,
            WDPortalNavigationHistoryMode.NO_DUPLICATIONS, "");
    }
    /**end
}

```

Related Content

[Portal Architecture](#)

[Web Dynpro Applications in Portal](#)

[Portal navigation](#)

For more information, visit the [Portal and Collaboration homepage](#).

Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.