

I already pointed out that the new release of WebCenter supports multi level navigation. To me, this is one of the best additions to this release. I already had a few customers that didn't like the idea of having only a single level of pages. We looked at different workarounds but they were all difficult to implement.

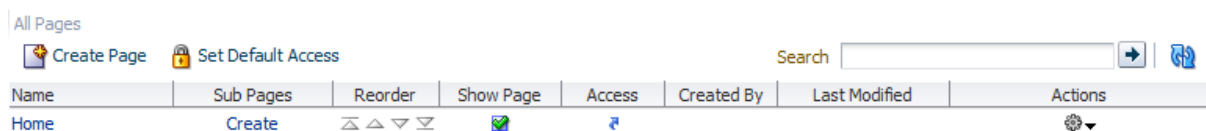
When I first saw the navigation models in this release I was very pleased with it. The navigation models are very flexible. You can create them both in JDeveloper as in runtime.

The concept of a navigation model is very cool. It enables administrators to create a navigation with more than just pages. You can for example also include taskflows or portlets in the navigation. When you click on a link to a portlet, a special render page will open that will only show that taskflow or portlet so no need to create a page with just a single component on it.

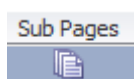
You can also add external links or links to external application. There is also the possibility to link to content from a content server or portal.

In this post I will point out some things you need to know about the new navigation model. As an example I just start with an empty Webcenter portal application.

When you go to the administration page, the first thing you see are the pages. By default you only have the Home page.

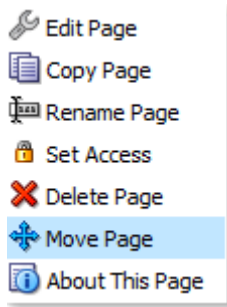


When a page has no children yet, you see a Create link in the Sub Pages column. Otherwise you will have a pages icon

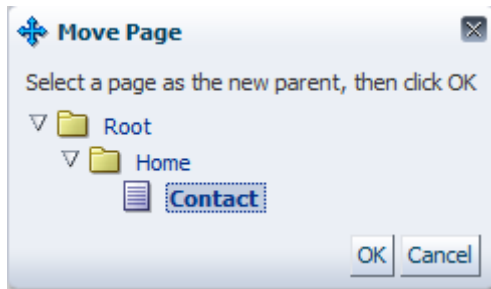


This way you can easily create child pages. Behind the screens, the pages will be added to the page hierarchy of the application. Each application has a single page hierarchy. In Webcenter you can find this file in the Web Content/oracle/webcenter/portalapp/pagehierarchy/ folder and is called pages.xml. When you add pages during runtime, these customizations are stored as layers in the MDS. The pages.xml from within JDeveloper will be merged with the customizations in the MDS repository.

If it of course also possible to move page. By moving a page you select the parent node were the page should be. This can be done by selecting the page in the resource manager and select Move from the action menu:



In the Move Page popup you can browse the page hierarchy and select the parent node:



In this example I first created a child page of the Home page called Contact. I then created a page that is a child of the Root and I want to make it a child of the Contact page. This means that when I press OK, I will have 3 levels.

You can go as deep as you want but make sure that the way you render the navigation is prepared for this.

You can also do the same in JDeveloper. When you open the pages.xml file you will see a nice interface to manage the page hierarchy:

#### Page Hierarchy

Drag ADF pages and drop them in the page hierarchy tree below.

Hierarchy: + X

- Root
  - Home

Id:\*

Title:

Path:

Visible

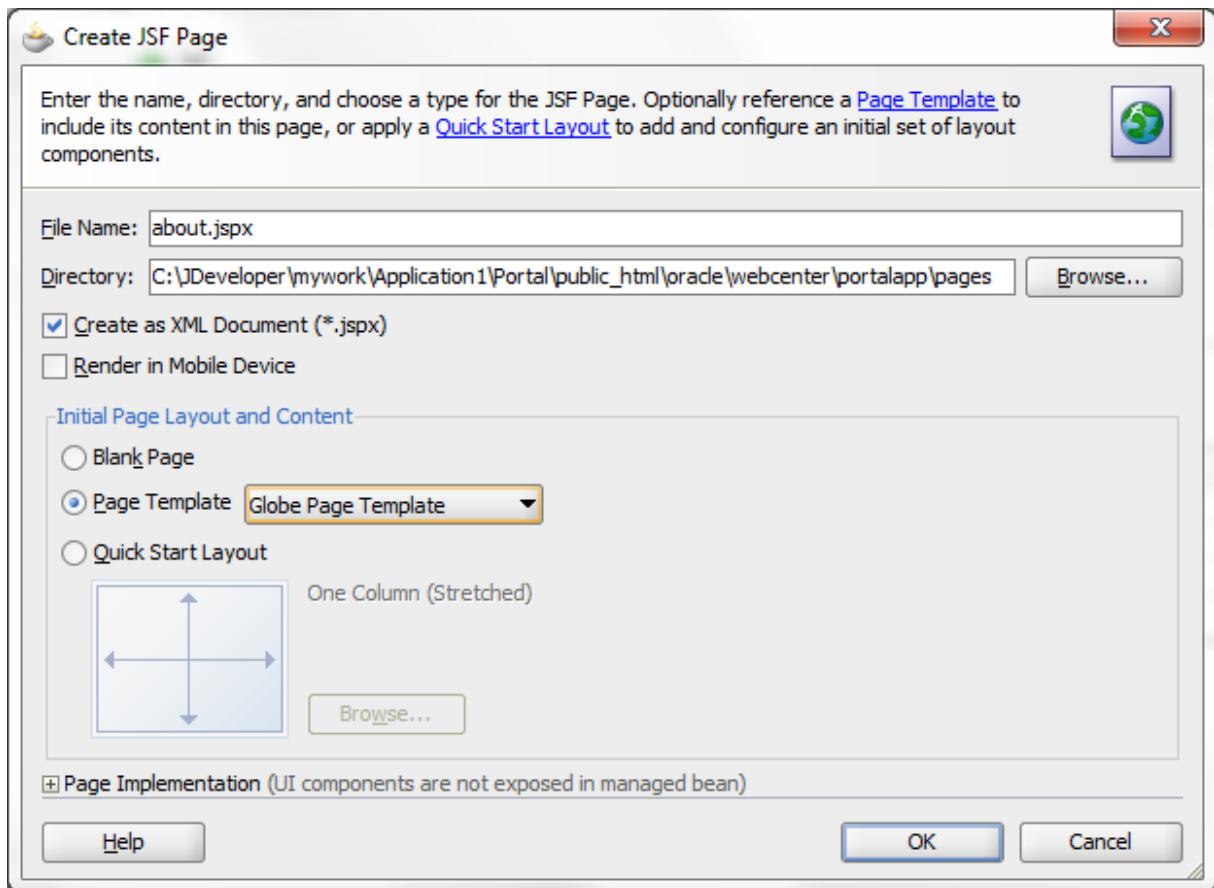
**Security** + X

Inherit Parent Security

Delegate Security

Role	Manage	Grant	Create	Delete	Update	Personalize	View
anonymous-role	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
authenticated-role	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Administrator	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Each level will have its own page hierarchy file. These are references in the pages.xml. For example we will create a new page in the pages folder:

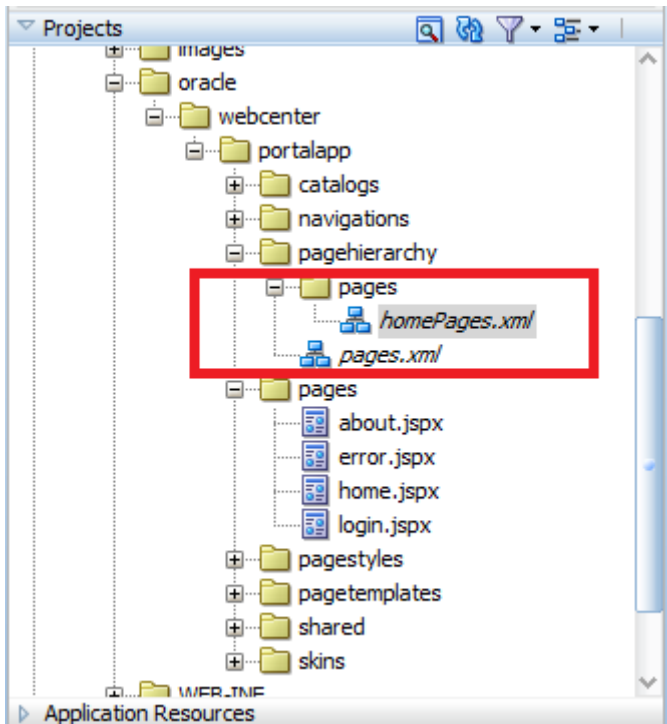


We name this About and use the Globe Page Template.

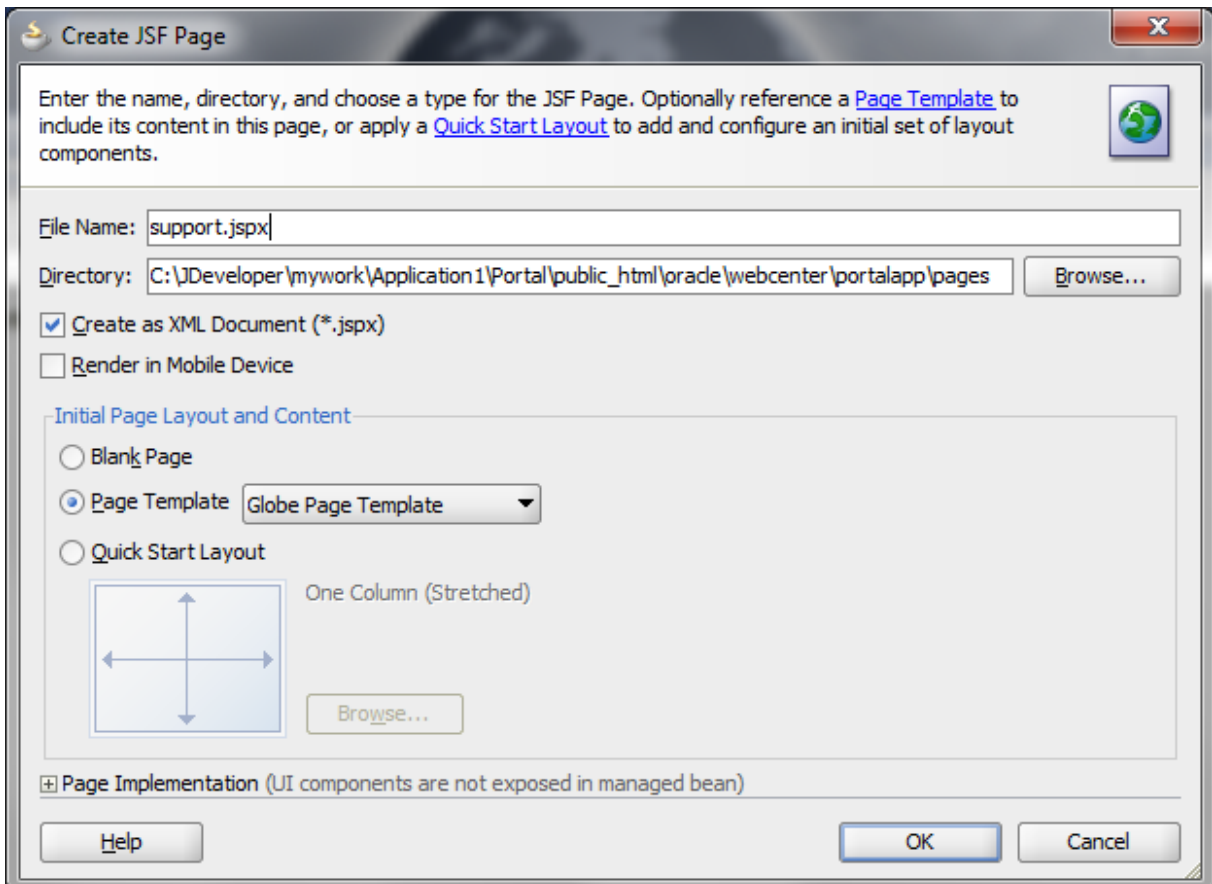
Once this page has been created we will add this as a child under the Home page.

Select the Home page in the pages.xml and press the Plus sign. Browse to the newly created about.jspx and select it.

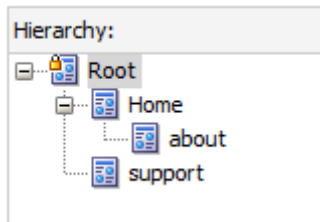
As soon as you select it, you will notice that a new folder under the pagehierarchy folder will be created with the homePages.xml file in it:



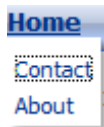
We will create a new page called support.jsp that also uses the Globe template and add this as a child under the root of the pages.xml hierarchy:



The page hierarchy should look like this now:



When you run the application and hover over the Home link you will see the subpages:



As you might notice, we also see the Contact page. This is because the page hierarchy is merged during runtime with the MDS customizations.

If the contact page is not showing you need to make sure the checkbox Show page is enabled on the page during runtime. When you add pages in runtime, this is disabled by default:

Name	Sub Pages	Reorder	Show Page	Access
Contact	Create	▲ ▲ ▼ ▼	<input type="checkbox"/>	↔

This shows how you can easily create and manage multi level portal sites.

In another blog post I will explain the navigation model in more depth.