

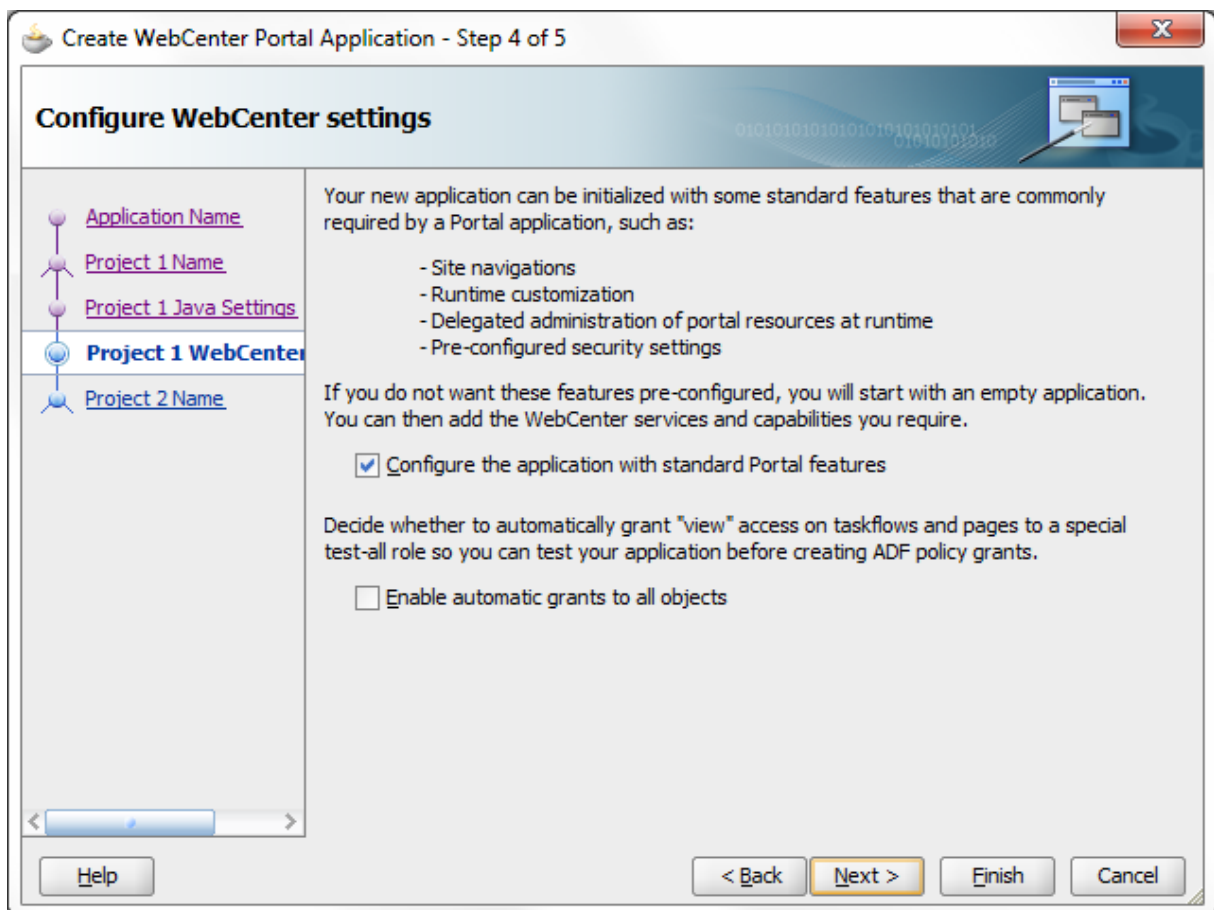
WebCenter 11.1.1.4

JDeveloper 11.1.1.4 has just been released and so are the WebCenter extensions. They come with a very big change in the WebCenter framework.

You will notice that when you create a new WebCenter portal application that is completely different from the application template in JDeveloper 11.1.1.3. In the previous version, the webcenter application template was a copy of the fusion web application with added webcenter libraries. In this version the template will create an out of the box portal with lots of features.

Therefore it is a nice idea to go over those features and changes so you can see what the new version of webcenter is all about.

First of all i will start with explaining the structure of the template. When you create the template you will notice that the final step is somewhat different. This is a special step to configure the WebCenter settings. In this step you can let JDeveloper create all the needed resources so you can easily start creating your portal. In fact, JDeveloper will create a readymade portal for you. This of course, only if you enable the **Configure the application with standard portal features**.



The template will also create a PortalWebAssets were you can store the assets for your web.

When you press finish, a lot of cool things happen behind the screen. You will see that a lot of files and folders are created. I will discuss most of them later on in this post but first we will explore the nice portal that we have just created.

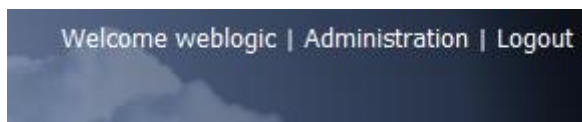
In JDeveloper, just right click the Portal project and press Run.



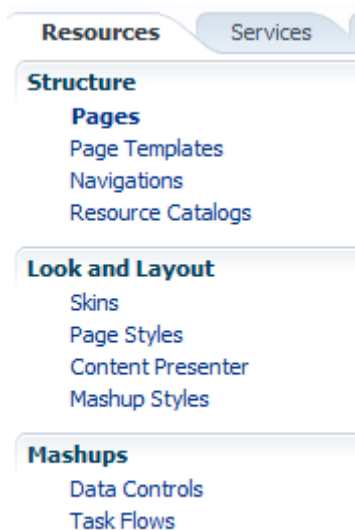
We haven't done anything and we have a nice page with login and navigation. All created by the template.

You can login with users that are defined in the weblogic server. By default the administrator is weblogic/weblogic1

When you log in to the portal, the login form will be replaced with following:



By clicking on the administration link you can start administer the portal and that's what is completely new in this release of webcenter. It is called the Resource Manager. In this manager you can manage all the resource for your portal.



All these resources can be managed during runtime. Without needing to create a single line of code you can create pages, templates, navigation models and so during runtime. This is a very big improvement over the previous release because we couldn't do anything during runtime beside creating pages and editing pages.

Some of these resources are new in this release of webcenter like navigations, mashup styles, content presenter,..

I will go over them one by one:

Pages

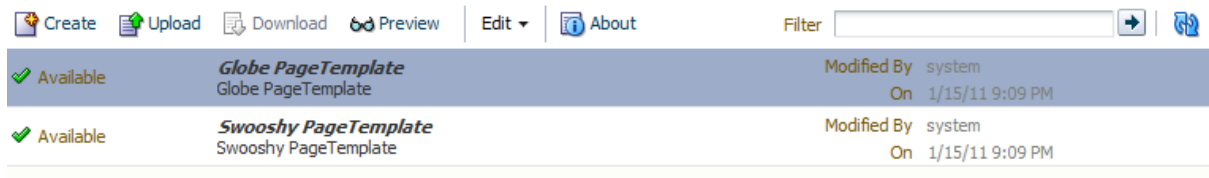
In the pages section of the resource manager you can manage the pages of your portal. You can create, edit, delete and set permissions on pages. When you open the actions menu of a page you can select Edit Page. This will bring you to the known composer. This way you can edit the page during runtime.



The composer has not changed much.

Page Templates

Page template dictates the look & feel of your pages. It defines the header, footer, navigation and regions where you can put the content, portlets and taskflows or other resources.



Now you can manage templates in runtime. You can even edit the template in runtime. You can also create a template in JDeveloper and import it in runtime so it is available for runtime creation of pages. By default templates created in JDeveloper can only be used in JDeveloper. When you right click such a template you can select **Create portal resource**. Then you can export it to an ear file which you can upload to a portal in runtime.

Navigations

Navigations are a cool new feature in Webcenter. They allow you to create different navigation models. A navigation model is a collection of links to resources. This can be pages, portlets, taskflows, content,... The navigation models will be used as a navigation for the users.

In this version it is possible to have **multi level navigation**. In webcenter 11.1.1.3, this was not possible. When you add a taskflow or portlet to a navigation, this will lead to a page with a template that only show 1 component. This way when you want to have a page with only that portlet or

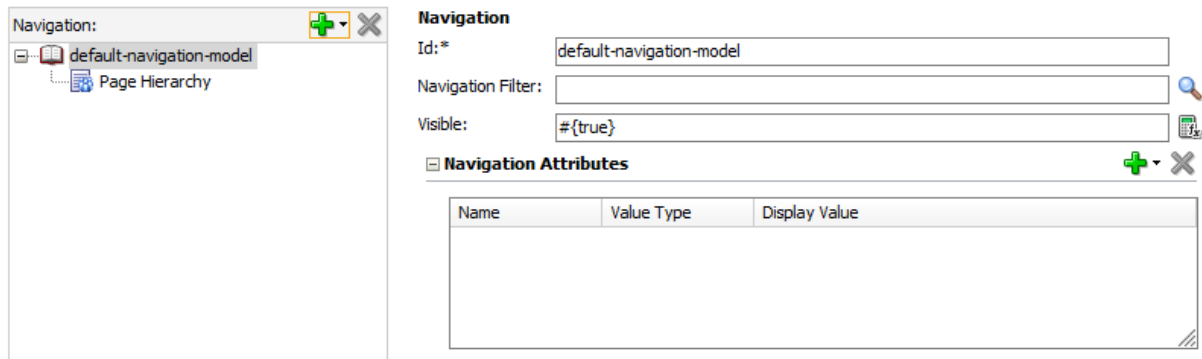
taskflow, you don't need to create it, you can create a link to the taskflow or portlet in your navigation model.

Navigation models can also be created and managed in runtime. The navigation models you create in JDeveloper are also made available in runtime. You don't need to export them.

The default navigation model is stored in `/oracle/webcenter/portalapp/navigations`. You can find it in `default-navigation-model.xml`: this is the navigation model used in the portal by default.

Navigation

Drag ADF pages, task flows, portlets, or other navigation definitions and drop them in the navigation tree below.



By default the navigation model only contains a page Hierarchy. You can only have a single page hierarchy in an application but each node will have its own page hierarchy file. There is only a single one that is the root.

When you create a complex page hierarchy for your application, you don't need to add the root, you can easily start with a child. For example when you have following page hierarchy:

- Company Portal (ROOT)
 - HR
 - Recruitment
 - Internal affairs
 - Finance
 - Payments
 - Invoices
 - Project
 - Project A
 - Project B

You could easily add the page hierarchy XML file for the HR page. By doing this, you will only add the HR pages to the navigation model.

Resource Catalogs

One of the best improvements of this release is the ease of use to manage resource catalog. In the previous version it was very hard to manage and maintain the resource catalog. In this release it is very easy. You can create resource catalogs in runtime, manage them, add resources to a catalog in

runtime,... No hassle anymore with the long paths of a taskflow. It is all made very simple to maintain. Both at runtime as in design time.

In JDeveloper the resource catalogs are stores in /oracle/webcenter/portalapp/catalogs. There you find two files:

- default-catalog: this is the default resource catalog
- catalog-registry: this is the library used to add resources to a catalog in runtime

When you open the default-catalog.xml you will notice you have a nice interface to manage the catalog.

Resource Catalog

Drag ADF pages, task flows, portlets, or other catalogs and drop them in the catalog tree below.

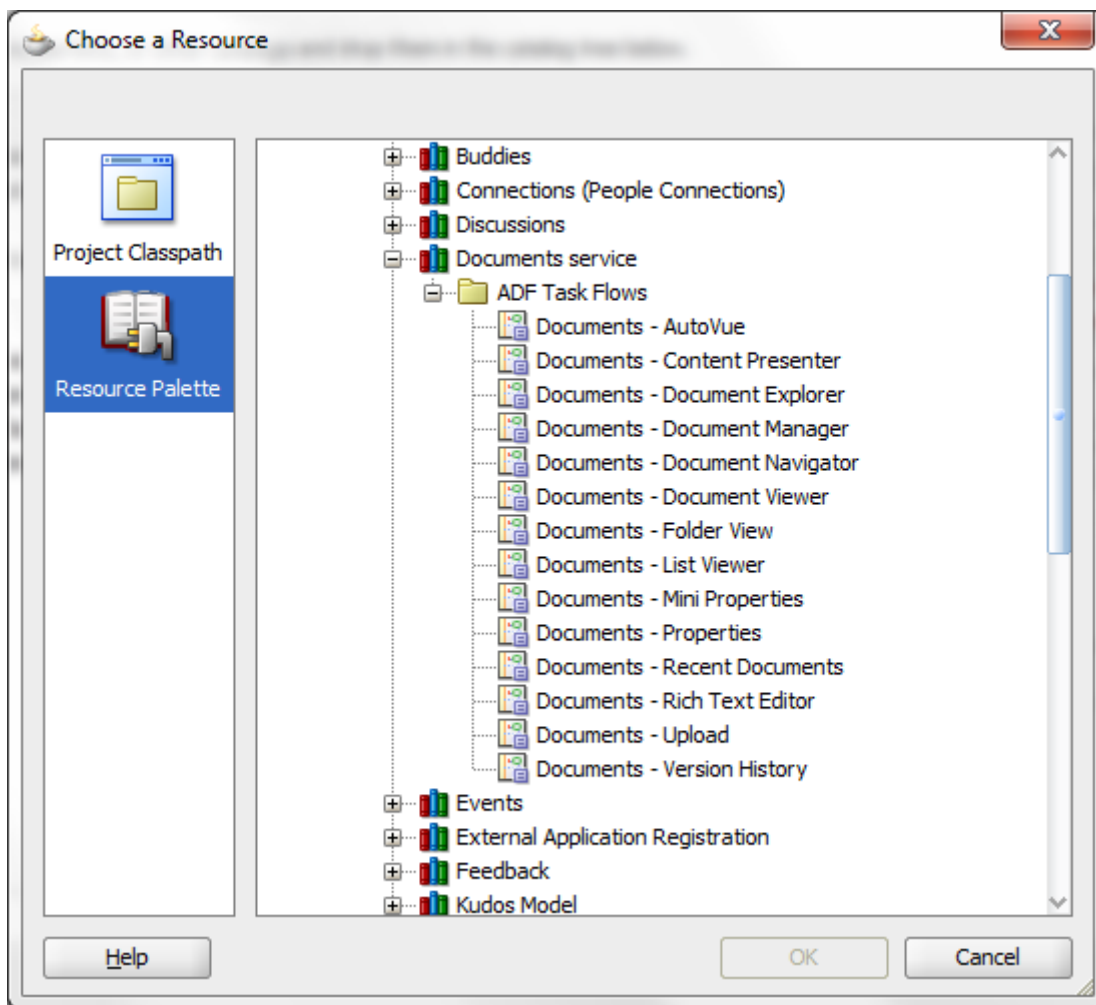
| Name | Value Type | Display Value |
|-------------|-----------------|--|
| Title | Resource Bundle | Default Portal Catalog |
| Description | Resource Bundle | Default resource catalog definition for WebCenter Portal ... |
| Subject | Resource Bundle | Portal, Catalog |
| IconURI | Literal String | /adf/webcenter/folderclosed_qualifier.png |

When you click on the plus sign you see what types of resources you can add:

A link is more a name for different types of resources. As you see, you can not choose from taskflow or portlet so at first sight you can't add a portlet or taskflow. They both are links.

When you add a link you need to provide the type of link:

You don't need to know the exact URL of your link because the browse button is context bound to the type. When you select taskflow as the type and press the browse button, you can select the taskflow from the resource palette:



When you select portlet, the browse button will show the registered providers and you can easily select the desired portlet.

These are some of the best and main new features of webcenter. There are dozens of others and it would be too big to discuss them all.

If you are interested in knowing more, I am currently busy writing a book about webcenter that is based upon this release. I hope it will be release somewhere near the summer.