Using CMIS query in a content query

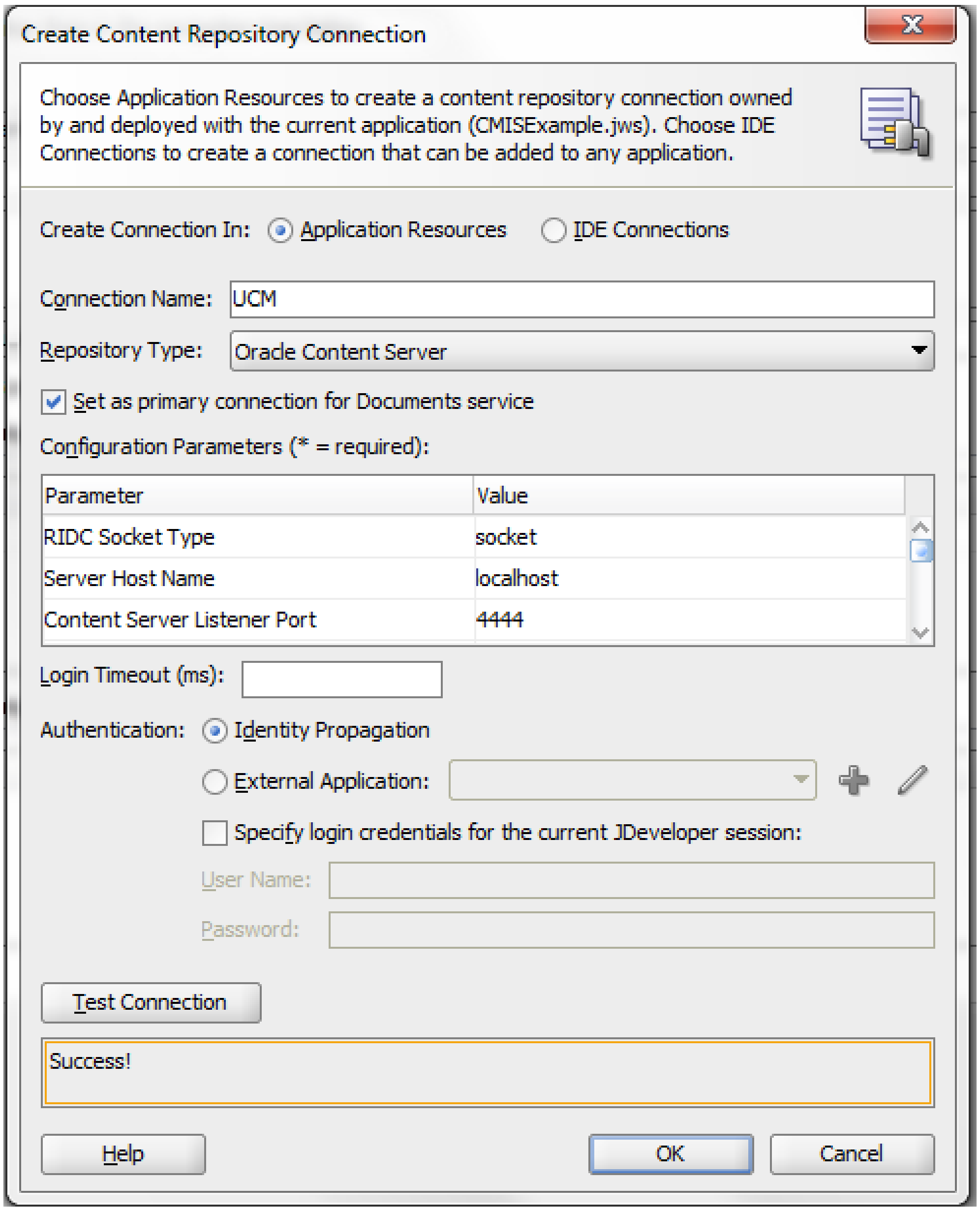
One of the new features in WebCenter PS3 is the new navigation model. It allows you build dynamic models . You can add all kinds of resources in the navigation model like external links, pages from your portal, taskflows, portals, content from a content server and so on.

When your portal is content driven, you definitly want the ability to add links to different items based upon some criteria instead of adding all the content manually. This can be achieved by adding a content query to your navigation model. A content query is based upon the CMIS standard. CMIS stands for Content Management Interoperability Services which is a standard for improving the interoperability between different content management servers. For example Alfresco, IBM, Microsoft, SAP,... support this standard.

CMIS has its own query language which looks a lot like the SQL language used in databases. At first sight a CMIS query can look somewhat difficult.

Before you can add a content query, you first need to create a connection to the content query. In JDeveloper right click the connections and select Content Repository from the New context menu.

Select Oracle content Server for the repository type and fill in the correct parameters for your server:


Create Content Repository Connection

Choose Application Resources to create a content repository connection owned by and deployed with the current application (CMISExample.jws). Choose IDE Connections to create a connection that can be added to any application.

Create Connection In: [✓ Application Resources] IDE Connections

Connection Name: UCM

Repository Type: Oracle Content Server

[✓] Set as primary connection for Documents service

Configuration Parameters (* = required):

Parameter	Value
RIDC Socket Type	socket
Server Host Name	localhost
Content Server Listener Port	4444

Login Timeout (ms):

Authentication: [✓ Identity Propagation] External Application:

Specify login credentials for the current JDeveloper session:

User Name:

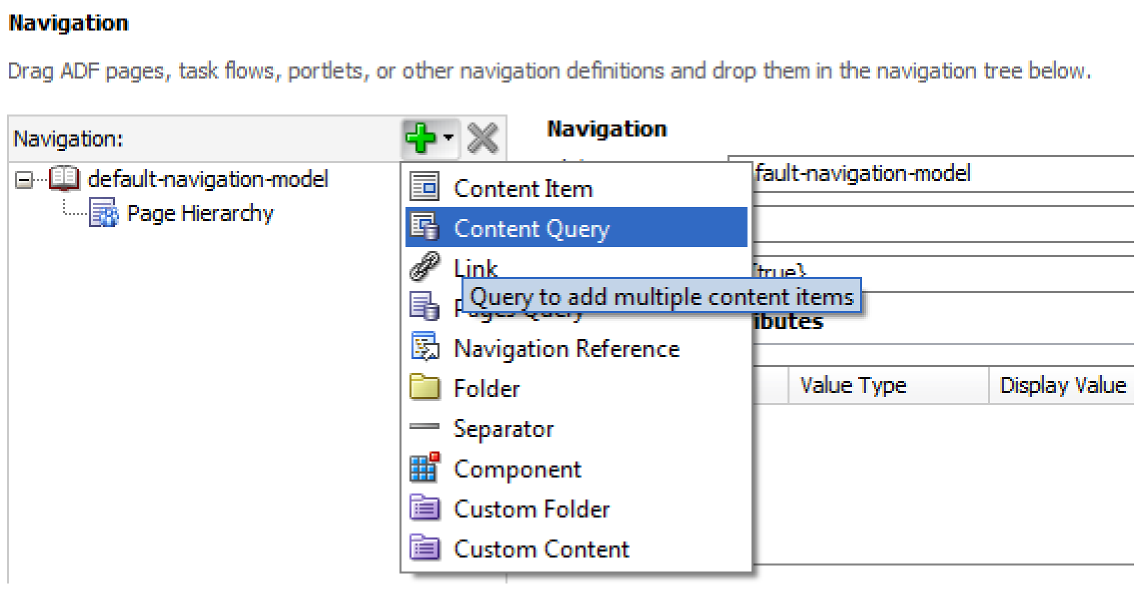
Password:

Test Connection

Success!

Before explaining the content query in detail we will first add it to a navigation model. Make sure you have created a WebCenter Portal application. Open the default-navigation-model.xml from the oracle/webcenter/portalapp/navigations folder.

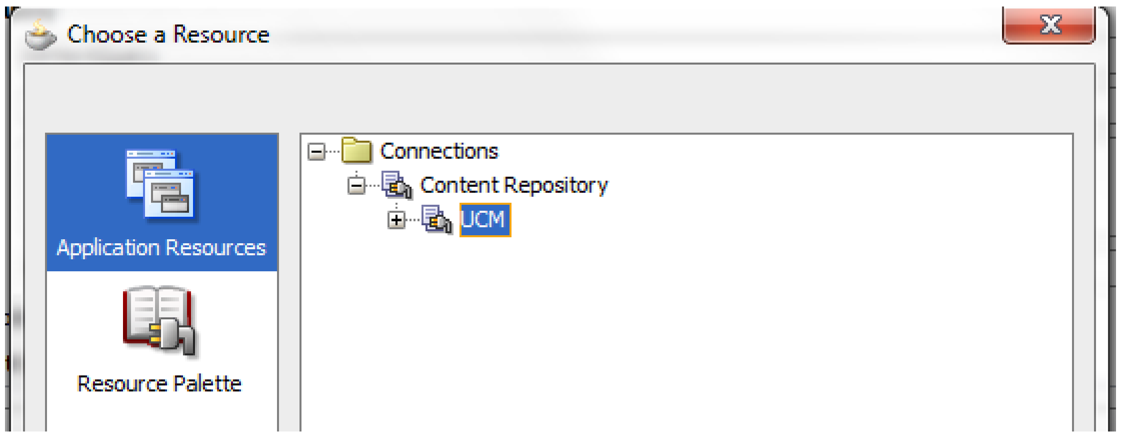
Select the default-navigation-model node and press the add button.


Navigation

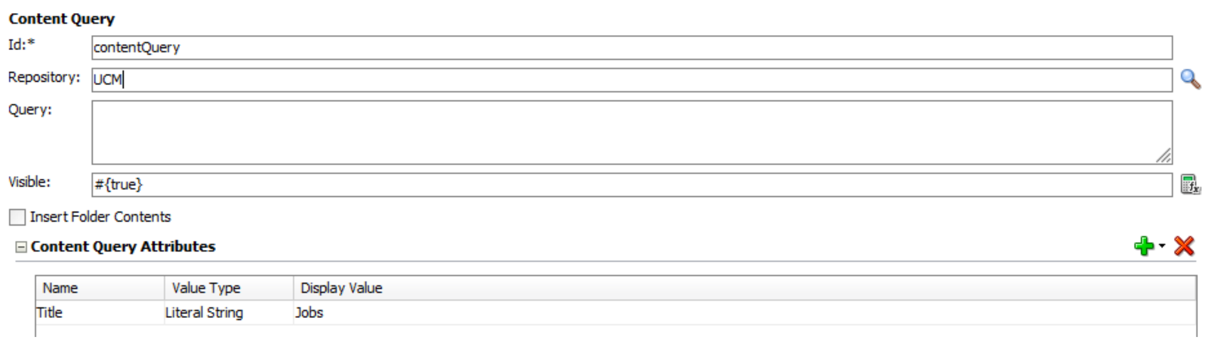
Drag ADF pages, task flows, portlets, or other navigation definitions and drop them in the navigation tree below.

Select the Content Query.

In the Repository field you need to select the content repository. Press the browse button so you can easily select the correct connection:



Specify a title which will be the root node for your query. The results in the query will be shown as children of this node.


Content Query

Id:* contentQuery

Repository: UCM

Query:

Visible: #{true}

Insert Folder Contents

Content Query Attributes

Name	Value Type	Display Value
Title	Literal String	Jobs

Now we can start explaining the query...

I will give a few examples of queries and explain them in detail:

Show all content in a specific folder

If you want to add all the items from a specific folder you need to use following query:

SELECT * from cmis:document where IN_TREE('/ucm/IDC:Folder/988901828852000401')

This query will return all the items from the folder with the specified ID. I still don't know if there is a query so we can use the path instead of the folder id. This query is not that good because when you migrate from a development environment to test or production environment, you have to make sure the ID's of the folders don0t change or the query will not work. A disadvantage of the CMIS query field is that you cannot use expression language to build dynamic queries...

Show all content with a specific term in the name

If you for example want to create a Jobs node in your portal and you want to include all the documents that has Job in its name than you would use following query:

SELECT * FROM cmis:document WHERE cmis:name LIKE Job%'

Using any metadata field in the query

You can use any metadata field in your query. Suppose you have create a custom metadata field that contains a productID and you want to show all the documents for that product:

SELECT * FROM ora:t:IDC:GlobalProfile WHERE ora:p:xProuctID = 15

Again... Because you can't use expression language the ProductID needs to be hardcoded and is not dynamic...

Also notice that that the query is case sensitive so if you have defined the ProductID field in your UCM as productId than you need to use xproductId instead of xProductID.

Here is a list of some fields available in the cmis:document object:

* cmis:createdBy is the name of the author which maps to dDocAuthor

* cmis:lastModifiedBy is the name of the user who made the last modification to the document. This field is mapped to dDocCreator.

* cmis:creationDate is the creation date which maps to the dCreateDate field

* cmis:lastModificationdate maps to dLastModifiedDate

* cmis:name maps to the name. dOriginalName when used in cmis:document ad dCollectionName when used in cmis:folder

* cmis:contentStreamMimeType contains the mime type which maps to dFormat

* cmis:objectId contains the id of the document which maps to dDocName

* cmis:objectTypeId contains the name of the UCM profile.