Customizing and extending IBM Content Navigator

Introduction

Welcome to the Hands-on Lab for IBM Content Navigator 2.0.

This lab is intended for individuals that have a basic understanding of the end-user runtime feature of IBM Content Navigator and are now interested in learning more about customizing and extending the product to better fit their organization's environment. It provides attendees with an overview of many of the key administration features and extension points of IBM Content Navigator.

As this lab is being run from a VMware image on a notebook computer, performance will not be equivalent to the performance obtained by running IBM Content Navigator in an enterprise configuration.

Content Navigator Administration

This chapter introduces you to the Content Navigator administration.
The Content Navigator administration is used to manage all Content Navigator features, including managing the connections to your enterprise ECM repositories, role specific views called desktops, menus, and labels.

Step 1 – Login to Windows

To begin, log in at the Windows operating system level using the **Administrator** account and the password **FileNetP8123**. To keep things simple for you, all other passwords on all levels are **IBMFileNetP8** in this image.

Step 2 – Start the Content Navigator Administration

Double-click on the Content Navigator Administration Internet Explorer icon on the desktop.

Step 3 – The Administration

You should now be looking at the IBM Content Navigator 2.0 Administration desktop. Login to the application using the following credentials:

User - filenetadmin, Password - IBMFileNetP8

Maximize the VMware image by selecting View > Full Screen from the VMware menu

Maximize the browser to enable it to be viewed more easily.

The Administration interface consists of six major areas:

Desktops

Repositories

Plugins

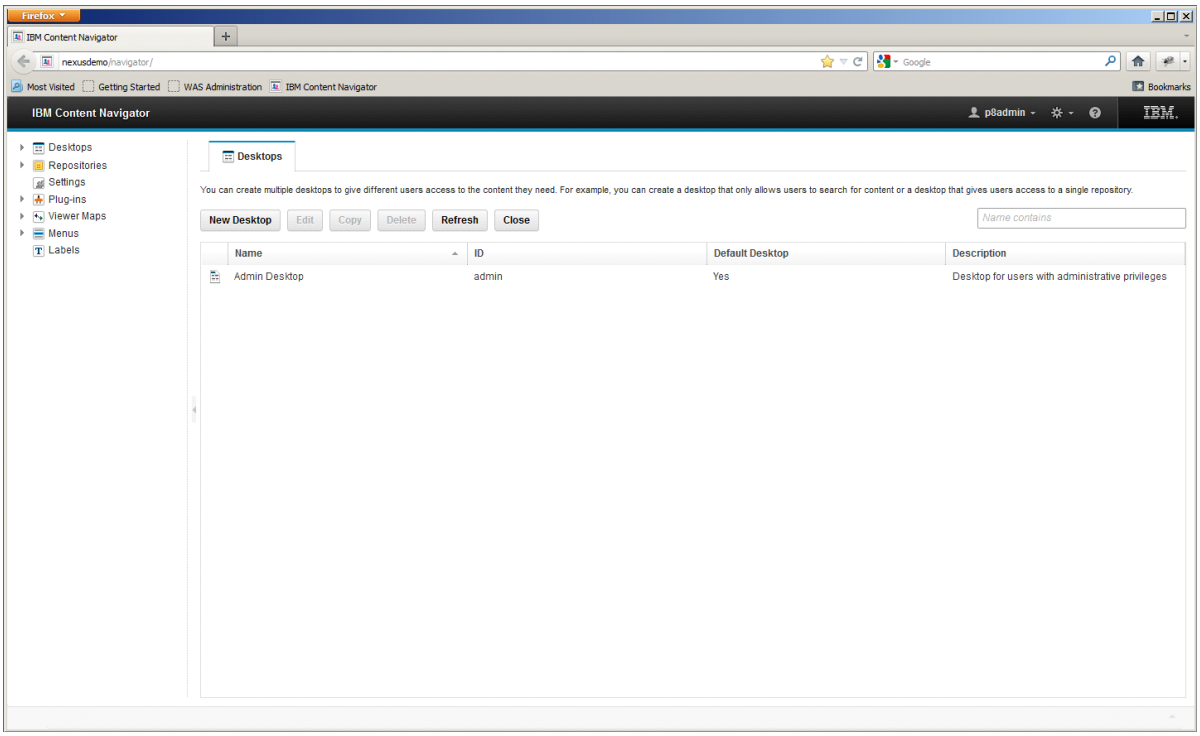
Viewer Maps

Menus

Labels

Each section is reviewed in detail below. Note that the screen shots in this lab manual may not exactly match what you see in your VMware image, but for the purposes of this overview small differences do not matter.

Desktops



First, a quick introduction to the concept of Desktops is necessary. You can think of a desktop as a role specific layout. An organization may have many desktops, one for each business unit. Each business unit may have it's own repository instances, unique workflows, and unique use-cases. A desktop allows the administrator to define what the user of that desktop will see when they log in to the application.

The Desktops panel is used to create and manage desktops. A desktop has the following configuration tabs:

General: The General tab is used to specify general settings such as the desktop name, whether the desktop is the overall default, whether users can specify security information, workflow connection point, etc.

Repositories: The repositories tab is used to select which repositories are assigned to the desktop. This is also where you specify which is the default repository. The default repository is the repository that the user is logging in to when they first access Content Navigator.

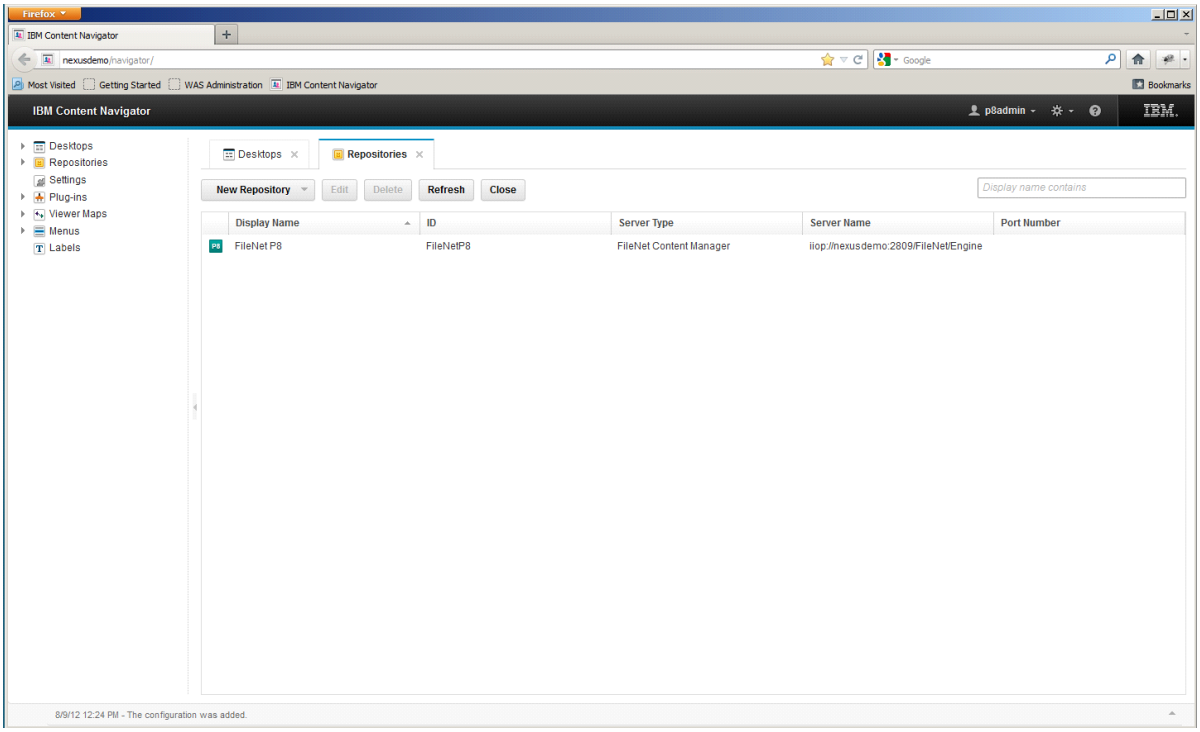
Appearance: The appearance tab allows the administrator to customize various visual settings including the application name that will show on the login page and top banner, custom URL's for password rules or help, and what features are available to the desktop. A feature is essentially a section of the application that exposes a particular set of functions. For example, Browse is a feature that has a tree and a content list and allows users to browse repositories. The administrator

can select which features are available to the desktop and if there are multiple repositories, which repository is the default repository for certain features such as Search, Browse, or Teamspaces.

Menus: Provides the administrator the ability to customize the menus available to the users of the desktop. If custom menus have been created then they can be selected here and assigned to the desktop.

Workflows: Administrators can assign one or more Application Spaces to the desktop here. This is where you can choose which workflows are assigned to the desktop.

Repositories



Repositories are critical to the application as these represent the content management systems to which Content Navigator is connected. Administrators can use this section of the administration to define exactly which repositories are available to the application. After repositories are registered here, they can then be associated with one or more desktops in the Desktops tab we covered in section 2.1.

Content Navigator supports four different types of repositories. Each type of repository has it's own set of tabs for configuring various aspects of the connection to those repositories:

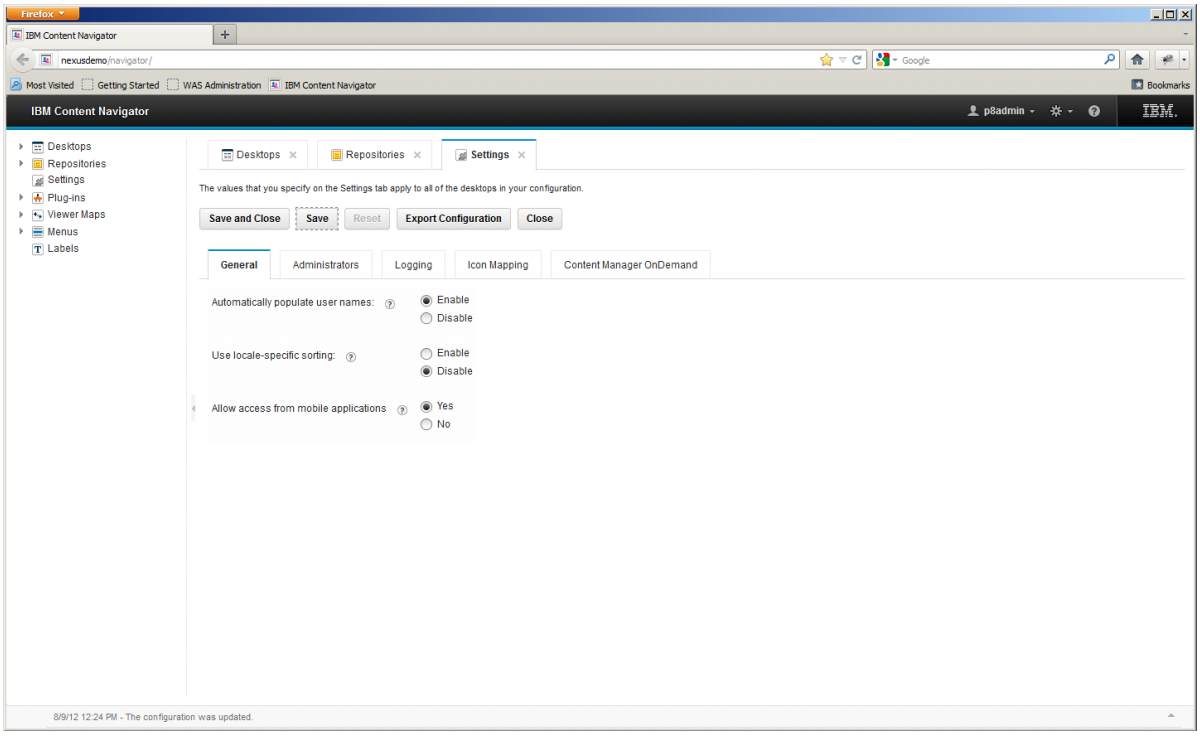
FileNet Content Manager: For FileNet CM repositories, you must create a repository definition for each Object store that you need to access. The General tab allows the administrator to enter basic connection information. After successfully connecting with an administrator account, the other tabs become available. Take a look at each of them and familiarize yourself with the settings.

Content Manager: For Content Manager you create one repository definition for each Library Server.

Content Manager OnDemand: For CMOD, you create one repository for each CMOD Library Server.

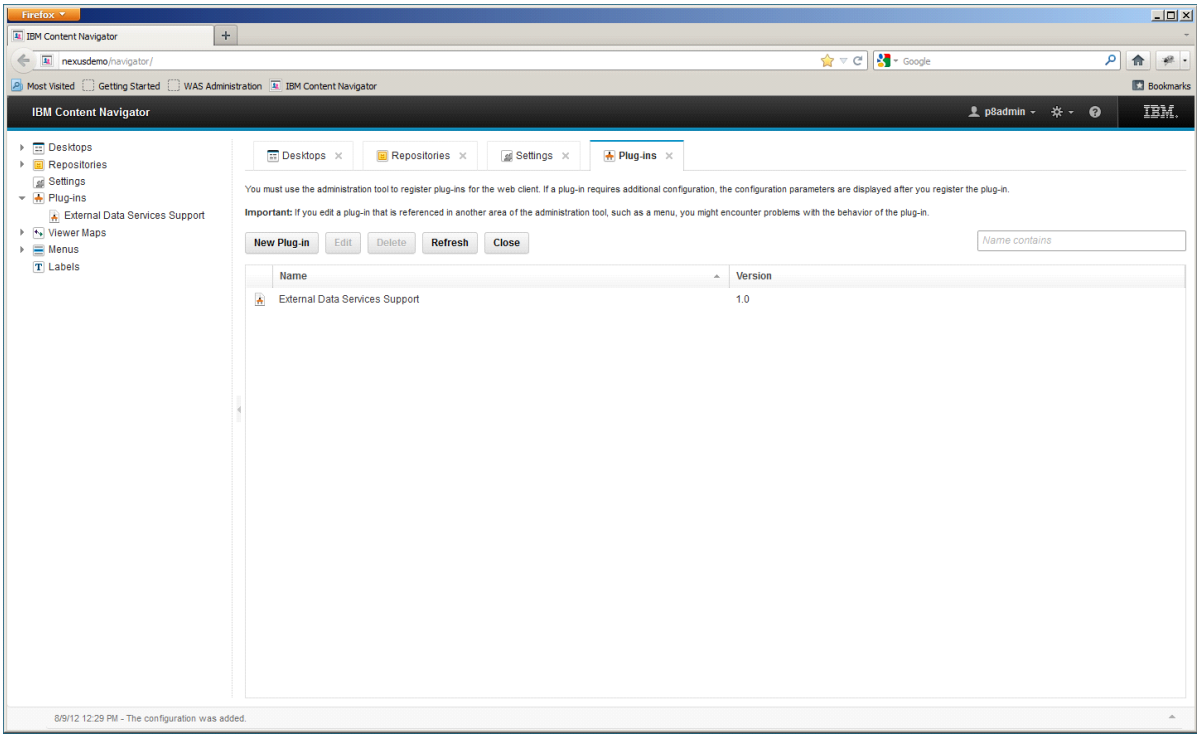
Content Management Interoperability Services: Content Navigator V2.0 ships with a technology preview connector to CMIS compliant repositories.

Settings



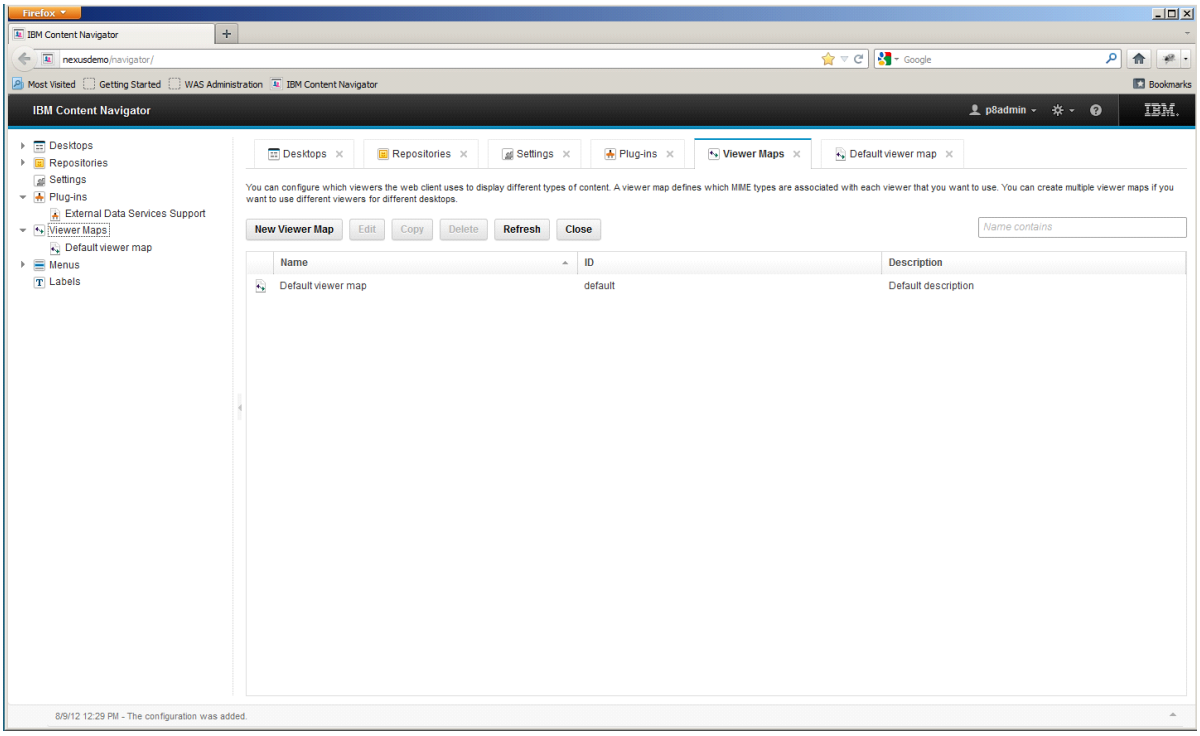
The Settings panel is used to control general settings for the entire application, independent of a specific repository or desktop. Take a look at each tab and familiarize yourself with the options available. Of particular note is the switch that allows an organization to disable mobile access from the Content Navigator iPad application and the Icon Mapping tab which allows an organization to override icons used in Content Navigator with their own custom icons. The Icon Mapping is a prime example of how Content Navigator can be customized without writing any actual code.

Plug-ins



The Plug-ins panel is where Administrators can add custom extensions to Content Navigator. We'll discuss Plug-ins in a bit more detail later in this lab.

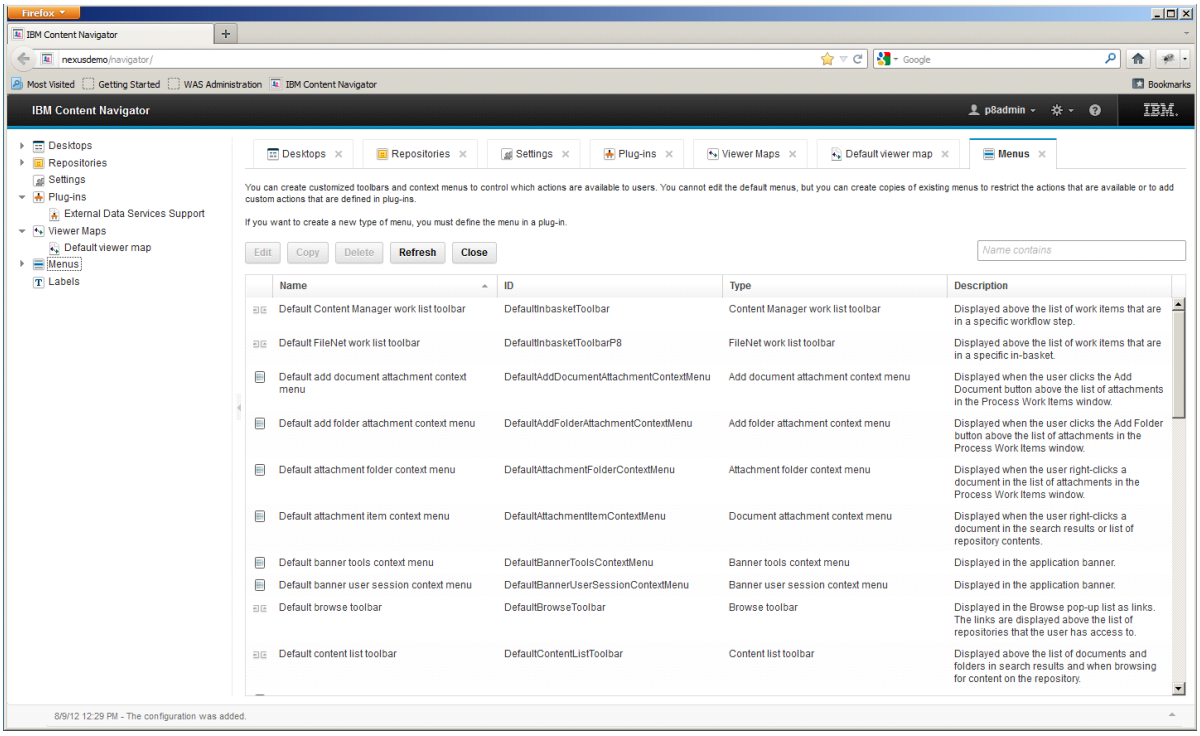
Viewer Maps



The Viewer Maps panel is where Administrators can customize how end-users will interact with the actual content stored in to the content repositories. To be a bit more specific, Viewer Maps control what viewers are launched when a user clicks on a document in any of the various

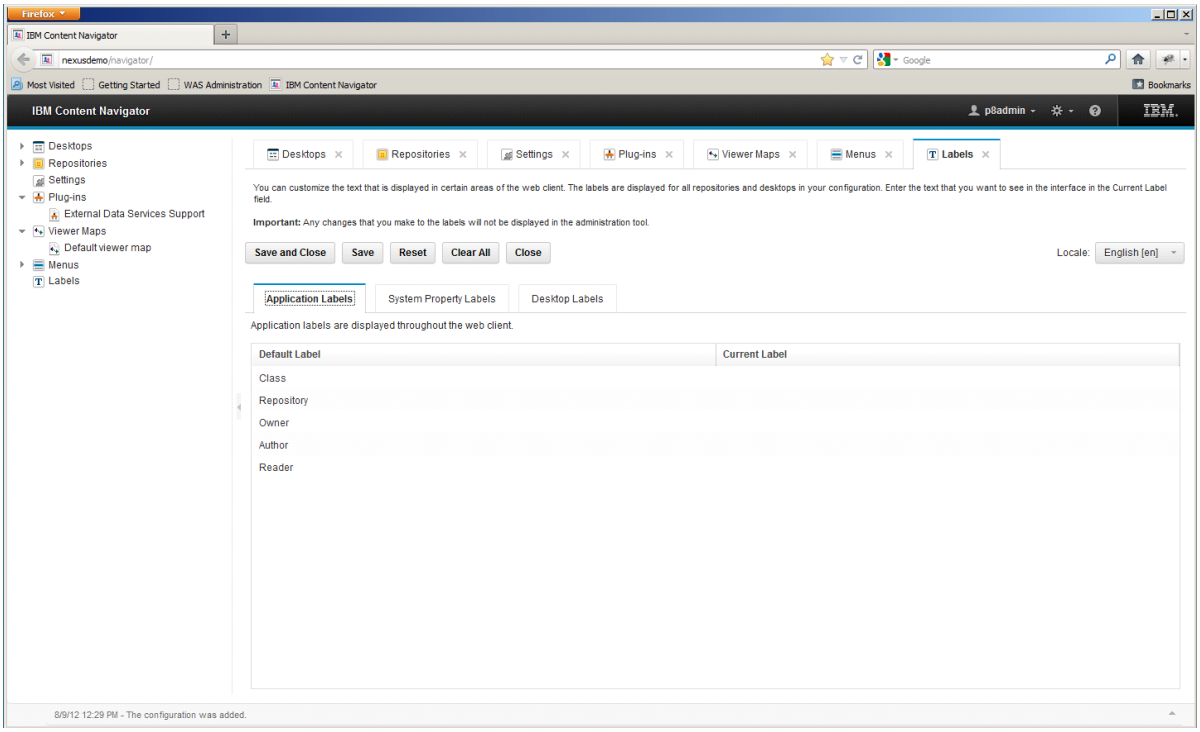
features in Content Navigator. Administrators can specify unique viewers per repository type per mimetype. The viewers are listed in the map in order of precedence so generally the top most entries are the most specific and then the last entries are more general with the intention being that the last entries are the general catch-all setting for any content type not already covered by a viewer definition higher up in the list. This is also where Administrators could register custom viewers that may have been created as a Plug-in by services organizations, or business partners, or even customers themselves.

Menus



The Menus panel is where Administrators can customize the menu definitions for the application. The default menus used by Content Navigator are all registered here and are not editable. If an administrator wishes to override a default menu, then you would simply create a copy of the default menu, rename it to something unique, and then make any changes desired. Once the new menu definition has been saved, it can then be associated with a particular desktop in the Desktops section. Plug-in developers can also create custom menus programmatically that would be registered in the Menus panel after a plug-in has been deployed.

Labels



The Labels panel is where Administrators can change many of the default labels used by Content Navigator in the application. For example, you can override application wide labels such as Class to call it Document Type or Object class. Additionally, you can override many System Property labels and you can also override specific labels for each Desktop. Again, this is another example of how Content Navigator can be customized without any actual development effort.

Lab Activity 1 – Creating a New Desktop Definition

In this activity, you will walk through how to create a new desktop definition using the Content Navigator administration.

Click on the Desktops link on the left hand navigation menu in the Content Navigator administration.

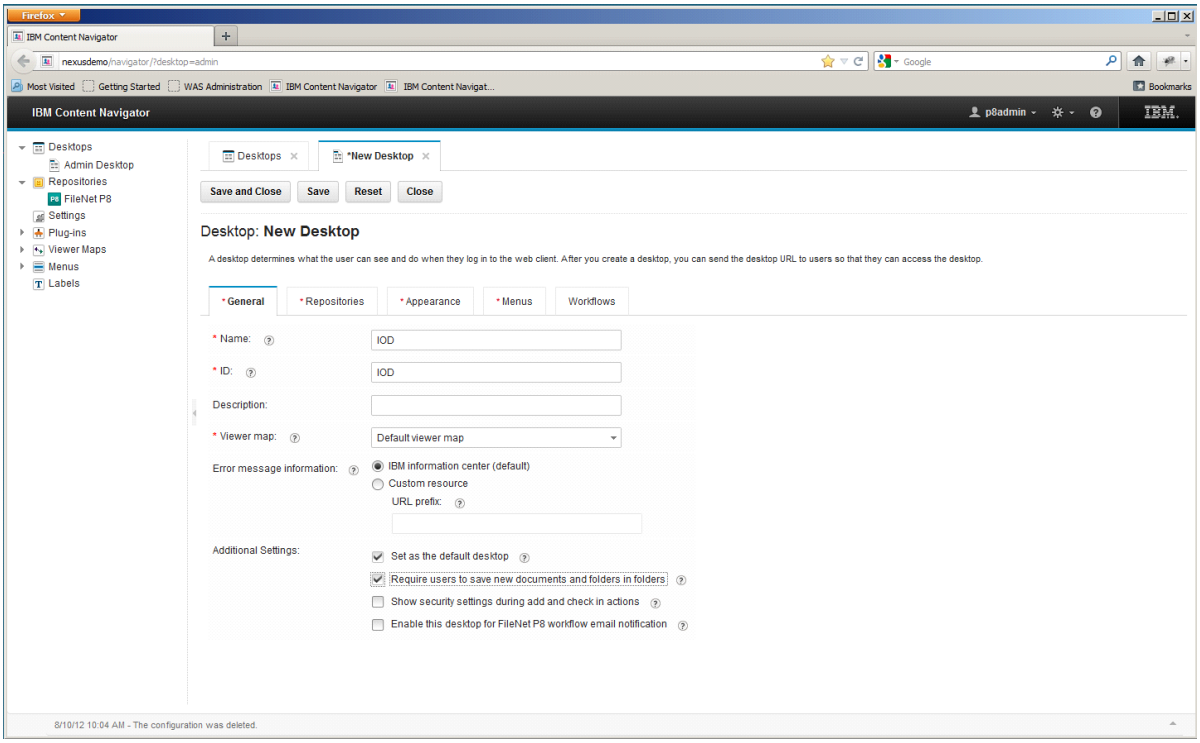
Click the New Desktop button.

On the General tab:

Specify a name of “ECM” in the Name text entry box.

Check the box next to “Set as the default desktop”

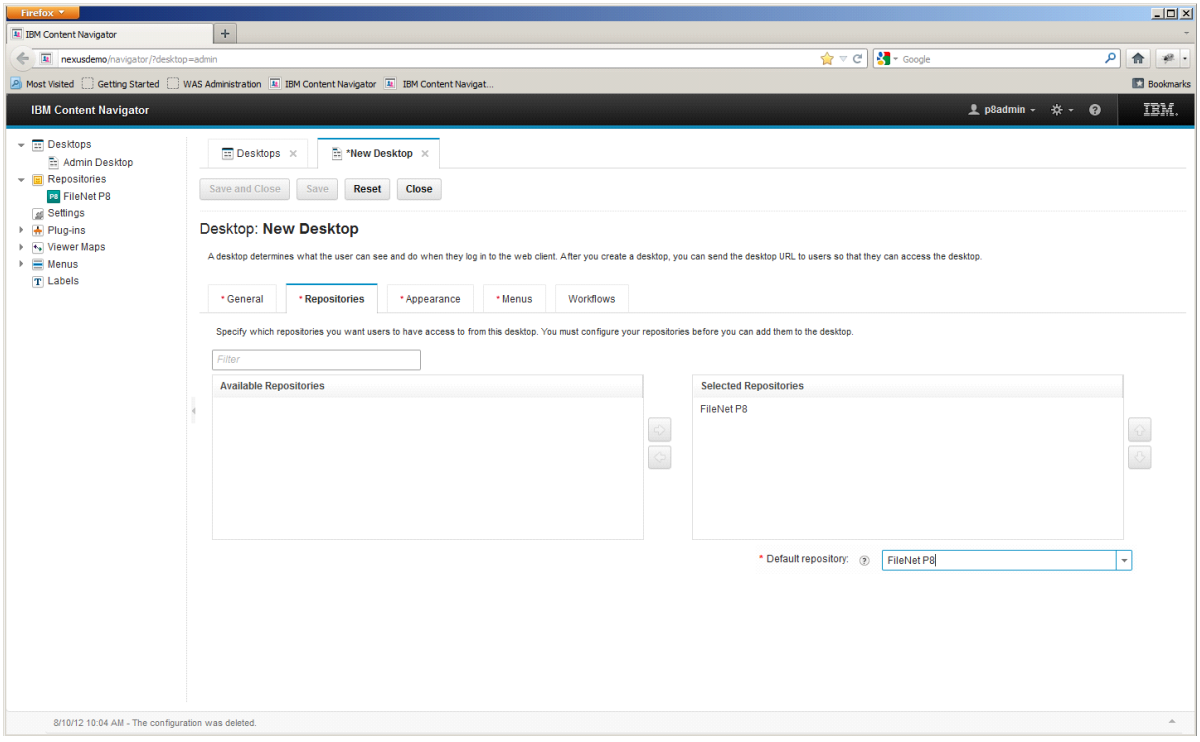
Check the box next to “Require users to save new documents and folders in folders”. This setting means that users cannot create documents in an unfiled state.



Select the Repositories tab:

In the Available Repositories tab select ECM and click the Right arrow icon in the middle of the page to move the repository definition to the Selected Repositories selection box.

In the Default Repository drop down select the ECM repository definition.

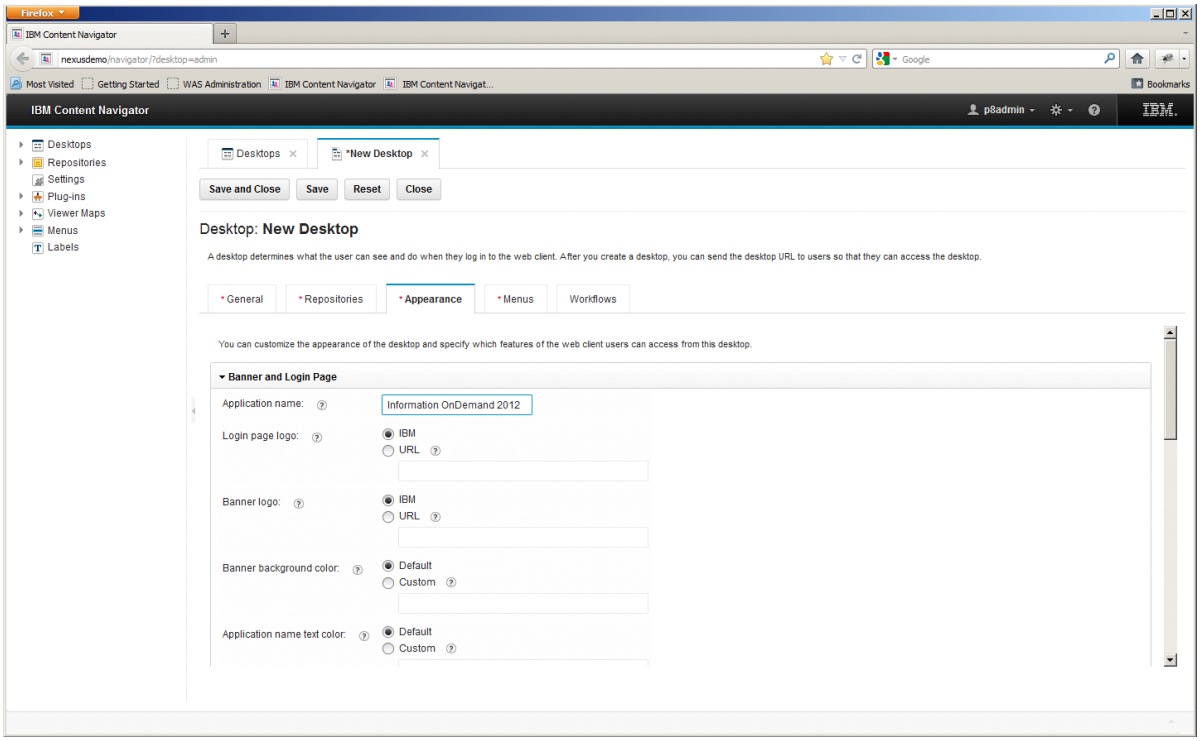


Select the Appearance tab:

For Application Name, enter “Enterprise Content Management”

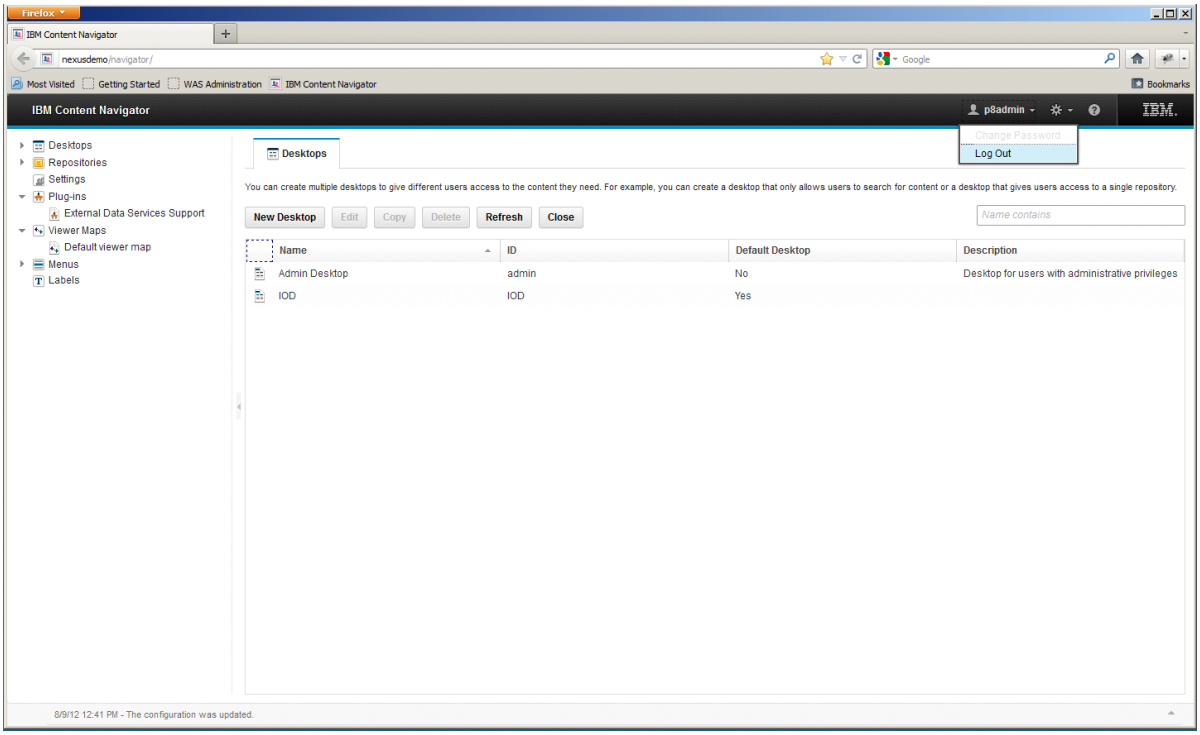
Click the twisty icon next to the Banner and Login Page label to collapse the section.

Note that the Layout section has Favorites, Browse, and Search as the Selected Features. Let’s leave this setting as-is.

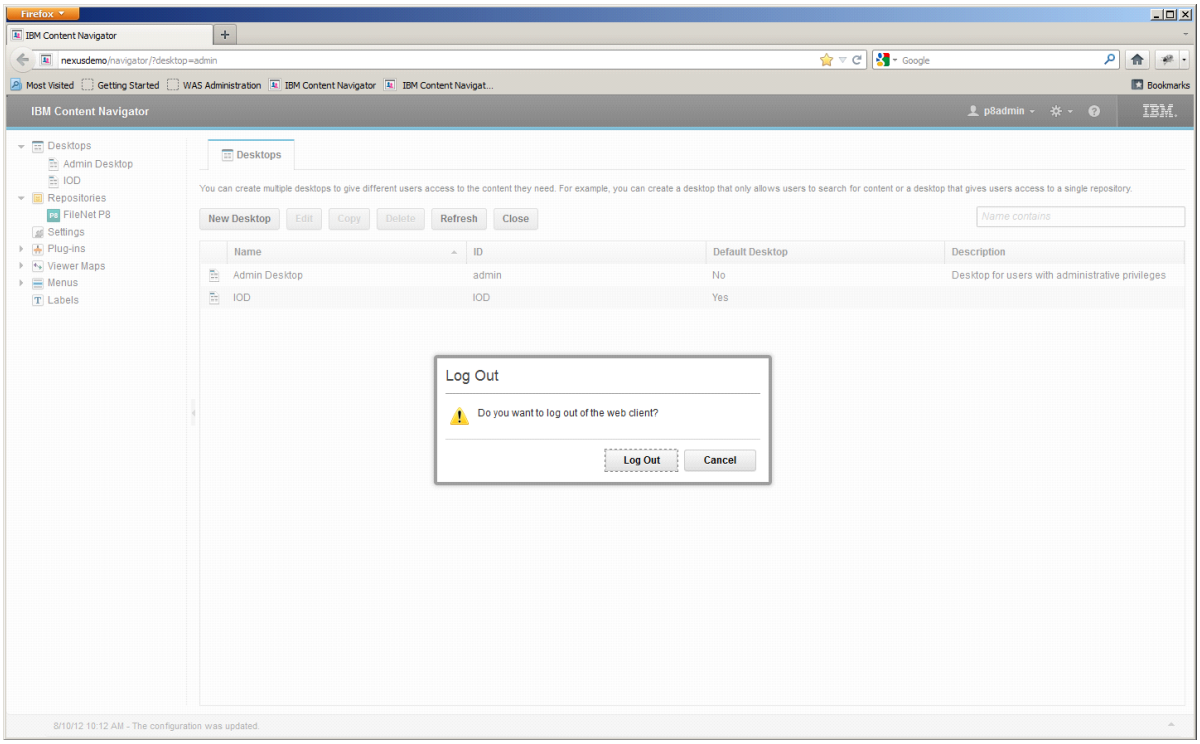


Click the Save and Close button.

Click the down arrow next to the username in the upper right hand corner of the application and select Log Out.



Click the Log Out button when prompted.



Close Internet Explorer.

Open Internet Explorer from the quick launch bar and enter http://Base-Win2k8x64:9080/navigator and click Enter.

Note that the application name is Enterprise Content Management that matches the name you specified in step 5a.

Login as filenetadmin / IBMFileNetP8

Congratulations, you've successfully created a new default desktop. This concludes Lab Activity 1.

Lab Activity 2 – Customizing the Login Page

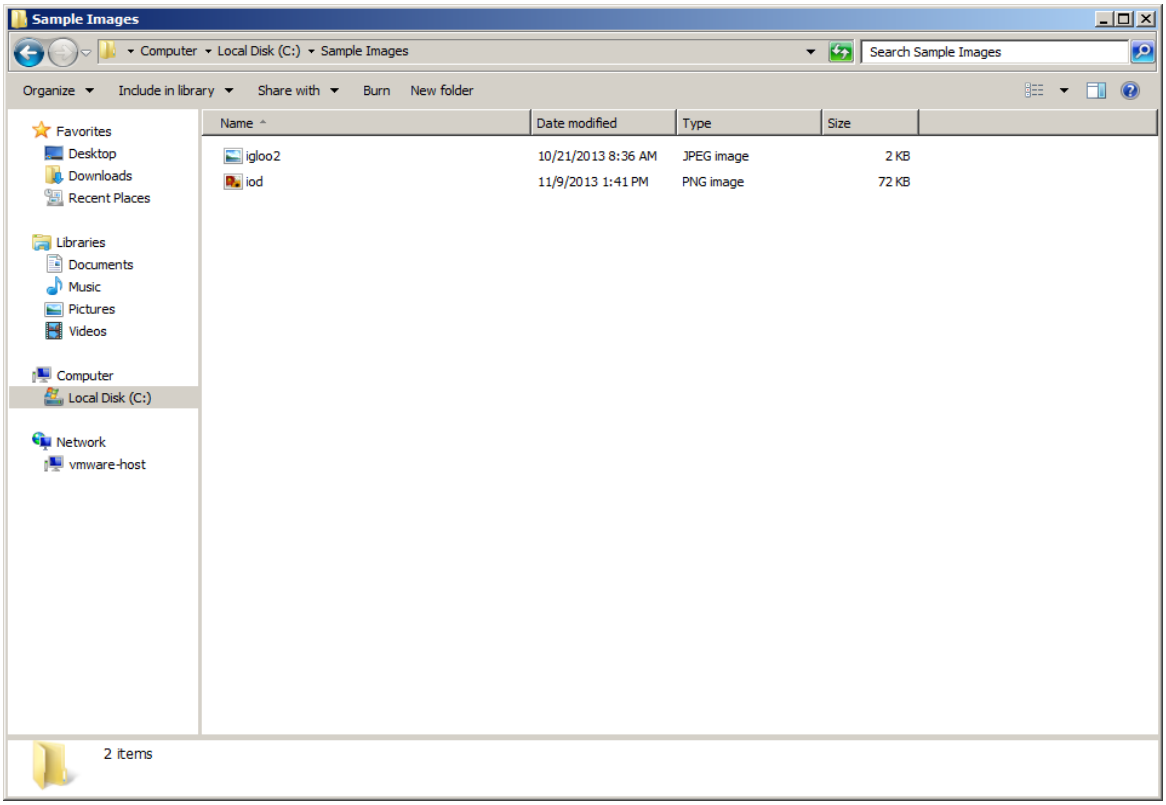
In the previous chapter you walked through the basic Content Navigator administration and familiarized yourself with all of the administration features. You also walked through a small exercise to create a simple Desktop.
In this chapter, we'll now take a look at how you can customize the desktop even further by changing the colors, logos, and URL's that the user will see when working with Content Navigator without having to write any code whatsoever.
Before we start with the Content Navigator administration steps, we'll first take a look at some sample image files and HTML files we'll use in this exercise.

Step 1 – Open Windows Explorer

Open Windows Explorer by clicking on right most icon in the Quick launch bar.

Step 2 – Open the Sample Images directory

Double click on the C:\Sample Images directory. Note the iod.png and igloo2.jpg files that are there. We'll be using this file to customize the login page logo later in the Navigator administration.



Step 3 – Copy an image to the WorkplaceXT images directory

In this step we'll copy the image over to a directory in a web application so that the image file will be URL addressable.

Right click on the file and select Copy.

On the left hand browse tree, expand the directory tree to C:\Program Files\IBM\WebSphere\AppServer\profiles\AppSrv01\installedApps\P8Node01Cell\WorkplaceXT.ear\web_client.war\images.

Right click in the images directory and choose Paste to add the iod.png and igloo2.jpg files to the directory.

Step 4 – Copy an html file to the WorkplaceXT images directory

This HTML file will be used to create a custom login help area on the ECM Desktop's login page.

In Windows Explorer, select the C:\Sample HTML Files directory.

Right click on the Login.html file and choose Copy.

On the left hand browse tree, expand the directory tree to C:\Program Files\IBM\WebSphere\AppServer\profiles\AppSrv01\installedApps\P8Node01Cell\WorkplaceXT.ear\web_client.war\images.

Right click in the images directory and choose Paste to add the Login.html file to the directory.

Step 5 – Open the Content Navigator Administration

Double-click on the Content Navigator Administration icon on the desktop. Login using the filenetadmin / IBMFileNetP8 user and password.

Step 6 – Open the ECM Desktop

After logging in to administration you should automatically be placed on the Desktops panel. Double click on the ECM Desktop.

Step 7 – Select the Appearance Tab

Click on the Appearance tab.

Step 8 – Update the Appearance Tab

In this step we'll be making changes to the desktop's Banner and Login Page settings.

Next to the Login Page Logo label, change the radio button to URL.

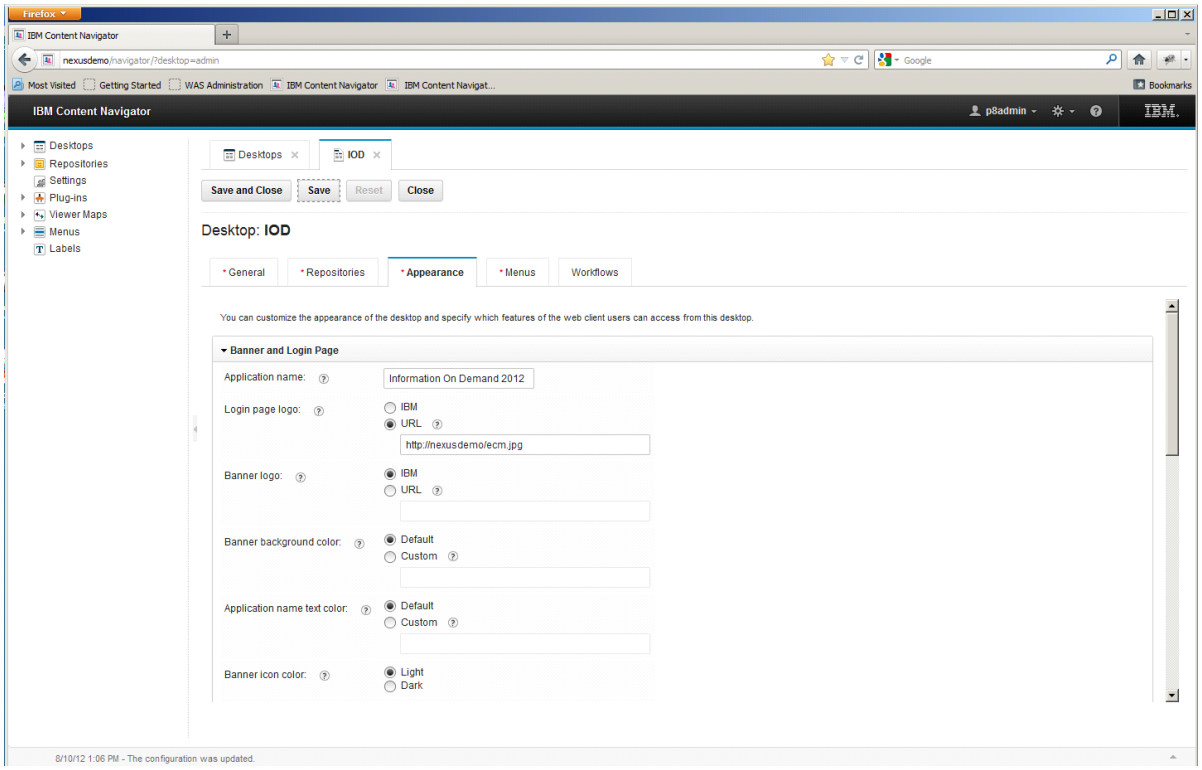
In the URL text box, enter http://Base-Win2k8x64:9080/WorkplaceXT/images/ecm.jpg

Scroll down a bit and change the radio button next to Banner logo label to URL.

In the URL text box, enter http://Base-Win2k8x64:9080/WorkplaceXT/images/igloo2.jpg

Scroll down a bit and change the radio button next to Login page content to URL.

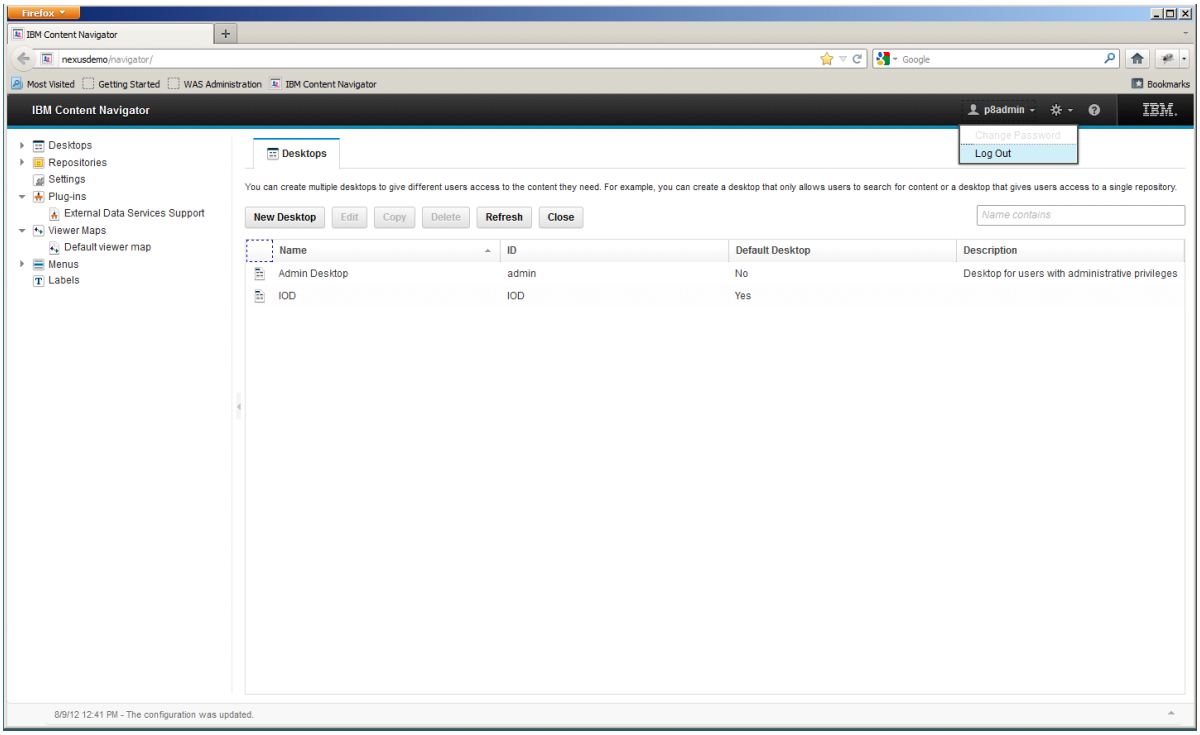
In the URL text box, enter http://Base-Win2k8x64:9080/WorkplaceXT/images/Login.html



Click Save and Close.

Step 9 – Logout and Reopen the Browser

Click the down arrow next to the username in the upper right hand corner of the application and select **Log Out**.



Click the **Log Out** button when prompted.

Close Internet Explorer.

Step 10 – Reload Content Navigator

Reopen Internet Explorer by clicking on the Internet Explorer icon in the Quick launch bar.

Enter **http://Base-Win2k8x64:9080/navigator?desktop=**ECM and hit Enter.

Note that your login page now has a section on the left that allows the user to get help with logging in to the application. Also note that the application now has a new icon in the bottom right corner of the login area.

Lab Activity 3 – Custom Menus

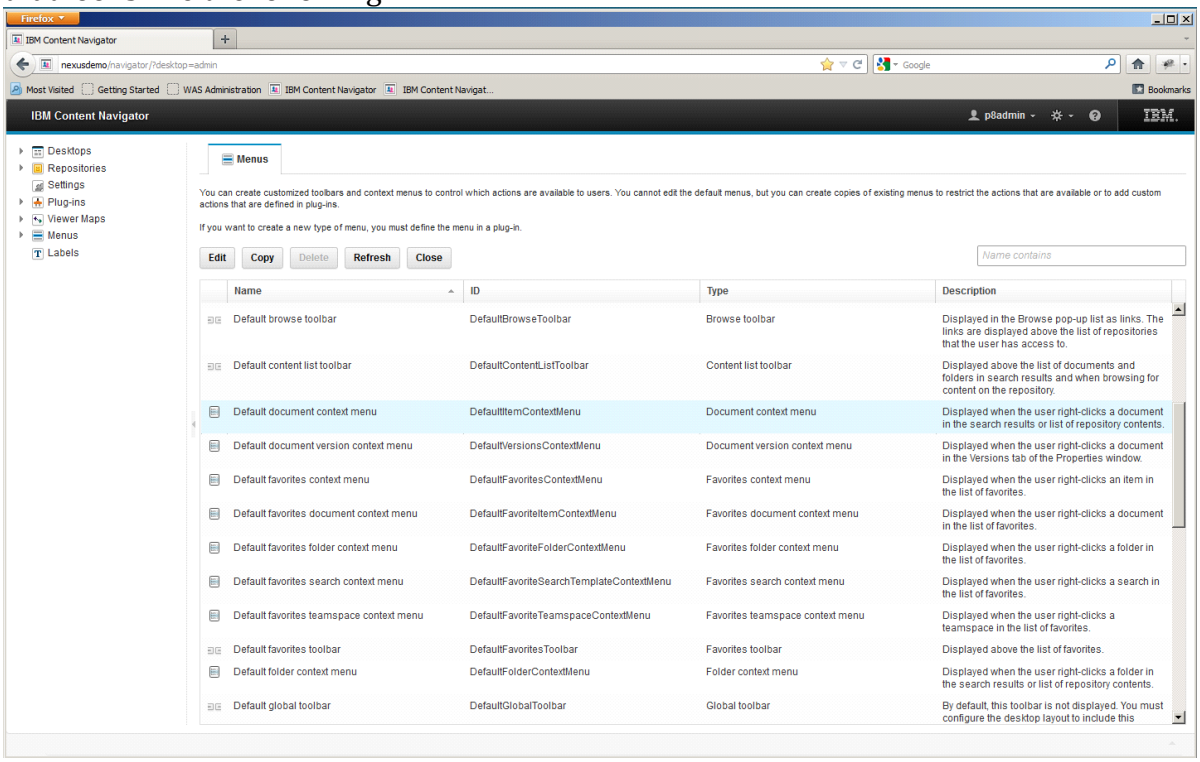
In this next Activity, we'll walkthrough how easily you can customize the menus that your users would see in a desktop, again without any development effort. Specifically, we're going to change the default menu that users of the ECM desktop will see when they right-click on a document in the content list so that they cannot Preview documents or download them as PDF. The reason that we're making this change is to reduce the load placed on the Content Navigator server in a production scenario by preventing these users from using those features.

Step 1 – Open the Content Navigator Administration

Double-click on the Content Navigator Administration icon on the desktop and login using **filenetadmin** as the user and **IBMFileNetP8** for the password.

Step 2 – Select the Menus Panel

On the left hand side of the screen, click on the **Menus** option in the tree. You should see a screen that looks like the following.



Step 3 – Copy an existing Menu

In this step we'll copy an existing menu and use it to create a new menu definition.

Scroll down and select the **Default document context menu**.

Right click on the **Default document context menu** and select **Copy**.

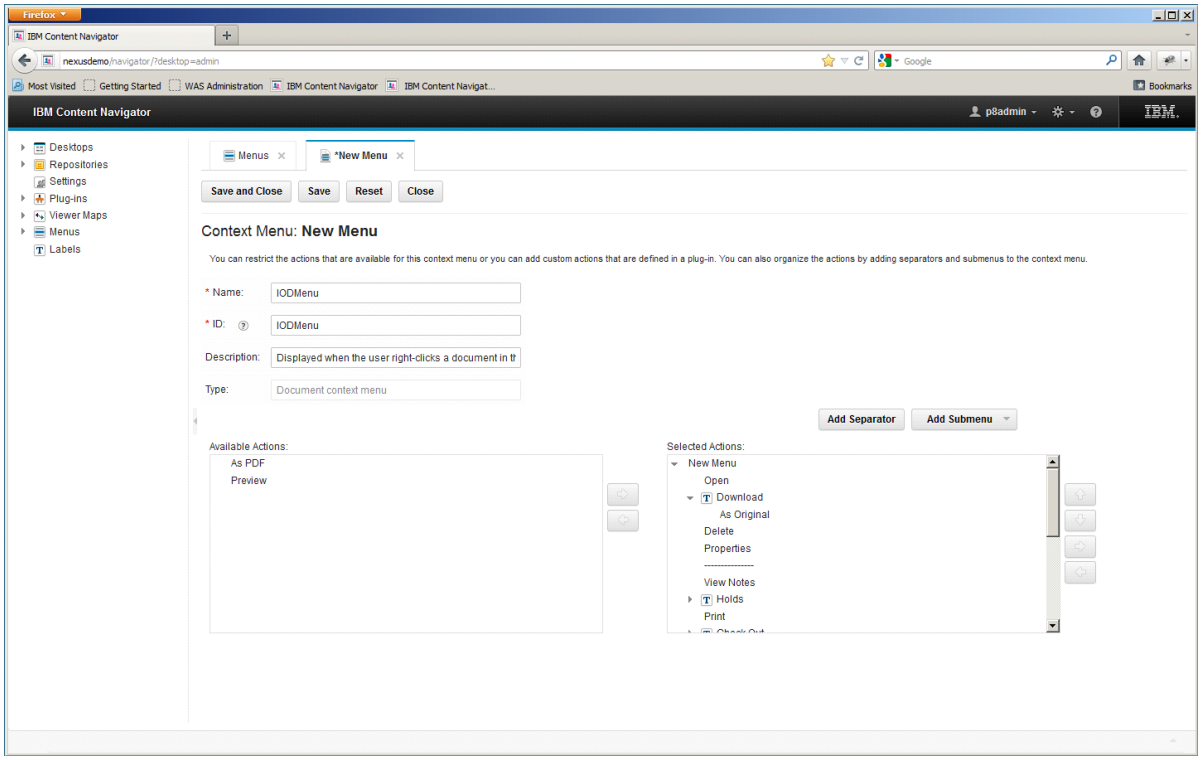
Step 4 – Create a new Menu

In this step we'll be making changes to the desktop's Banner and Login Page settings.

In the **Name** field enter **ECMCustomMenu**.

In the **Selected Actions** window on the right hand side of the screen, select **Preview** and then click the left arrow to move that to the **Available Actions** list.

Next, in the **Selected Actions** window on the right hand side of the screen, expand **Download** and select the **As PDF** option and then click the left arrow to move it to the **Available Actions** list.



Click the **Save and Close** button.

Step 5 – Assign the Menu to the Desktop

In the left hand navigation tree, expand **Desktops** and select the **ECM** desktop.

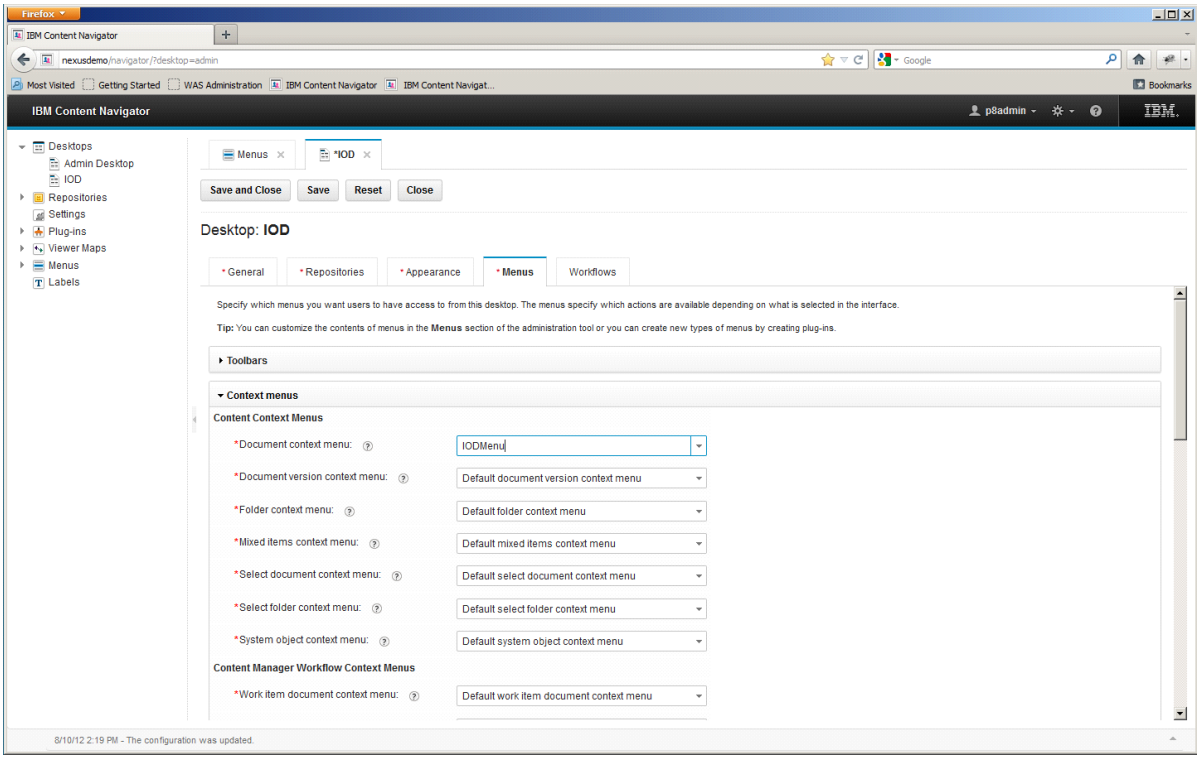
Click on the **Menus** tab.

Click the down arrow next to the **Toolbars** heading to collapse the **Toolbars** section.

In the **Context menus** section click the drop down next to the **Document context menu** label.

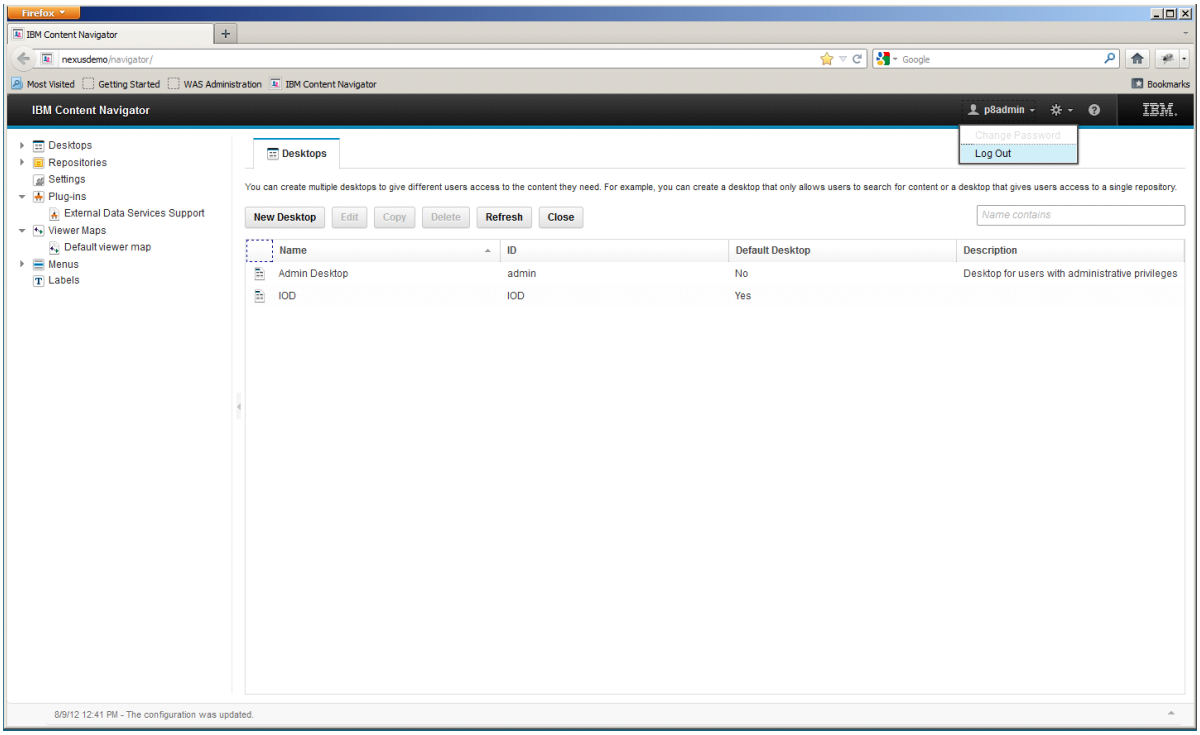
In the drop down list, select **ECMCustomMenu**.

Click the **Save and Close** button.



Step 6 – Logout and Reopen the Browser

Click the down arrow next to the username in the upper right hand corner of the application and select Log Out.



Click the **Log Out** button when prompted.

Close IE.

Step 7 – Reload Content Navigator

Reopen IE by clicking on the IE icon in the Quick launch bar.

Enter **http://Base-Win2k8x64:9080/navigator?desktop=**ECM and hit Enter.

Enter **filenetadmin** as the user and IBMFileNetP8 as the password.

After logging in, select the **Browse** icon on the left hand side of the screen.

Expand the **Invoices** folder.

Select any document in the list and then right click on the document.

Note that your menu does not have the option to **Preview** or **Download As PDF**.

Lab Activity 4 – Custom Labels

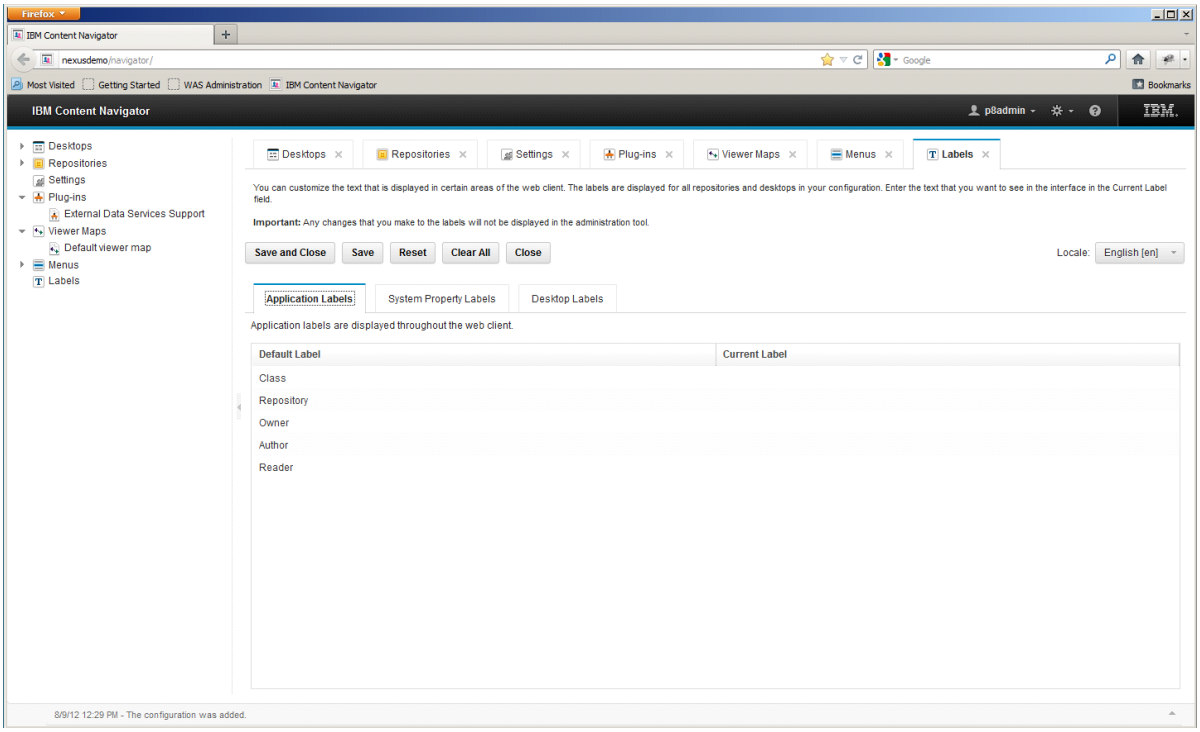
In this next Activity, we'll walkthrough how easily you can change the default Application labels used throughout Content Navigator. Again, this can be done without any development effort. Sensing a trend yet? The use-case here is that a customer that only has FileNet Content Manager may want to use the labels of Object class and Object store instead of Class and Repository. This activity will walk you through how to make this change and test it in the application.

Step 1 – Open the Content Navigator Administration

Double-click on the Content Navigator Administration icon on the desktop and login using **filenetadmin** as the user and **IBMFileNetP8** for the password.

Step 2 – Select the Labels Panel

On the left hand side of the screen, click on the **Labels** option in the tree. You should see a screen that looks like the following.



Step 3 – Edit the Application Labels

Here we'll actually update the **Application** labels called **Class** and **Repository** to new values.

Find the **Class label** in the **Default Label** column. Click in the space next to that label in the **Current Label** column.

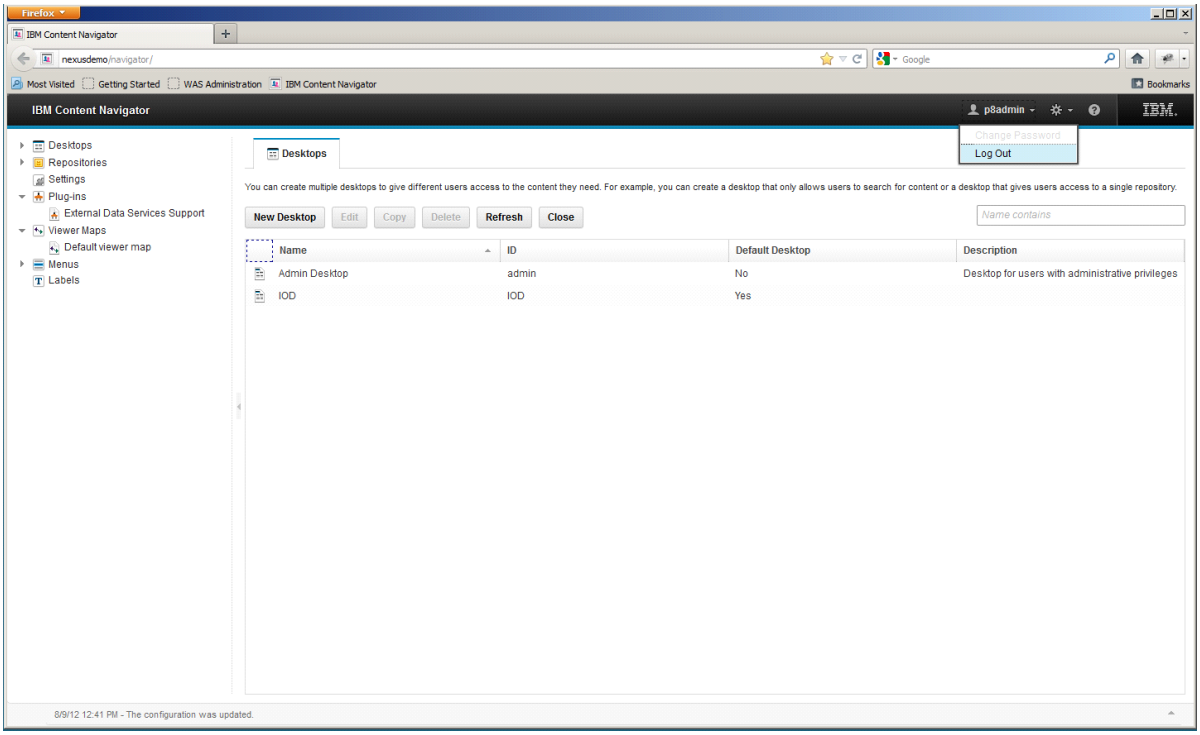
Enter **Object Class** as the new label.

Repeat steps 1 and 2 for **Repository**, this time using a new label of **Object Store**.

Click the **Save and Close** button.

Step 4 – Logout and Reopen the Browser

Click the down arrow next to the username in the upper right hand corner of the application and select **Log Out**.



Click the **Log Out** button when prompted.

Close IE.

Step 5 – Reload Content Navigator

Reopen IE by clicking on the IE icon in the Quick launch bar.

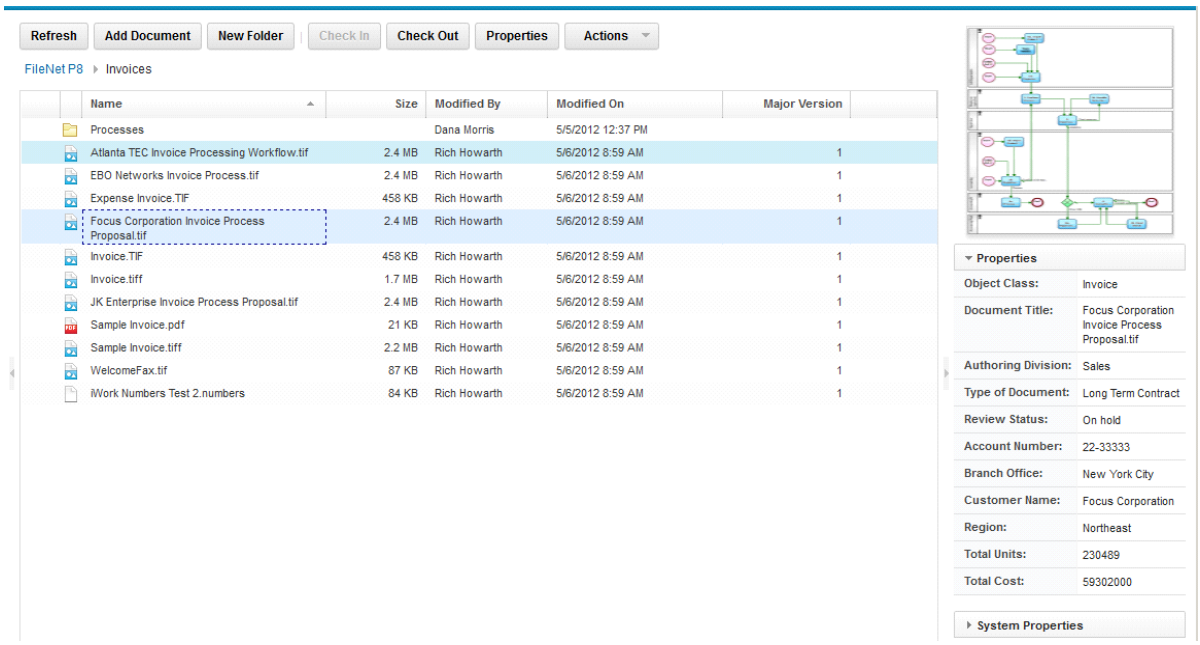
Enter **http://Base-Win2k8x64:9080/navigator?desktop=**ECM and hit Enter.

Enter **filenetadmin** as the user and **IBMFileNetP8** as the password.

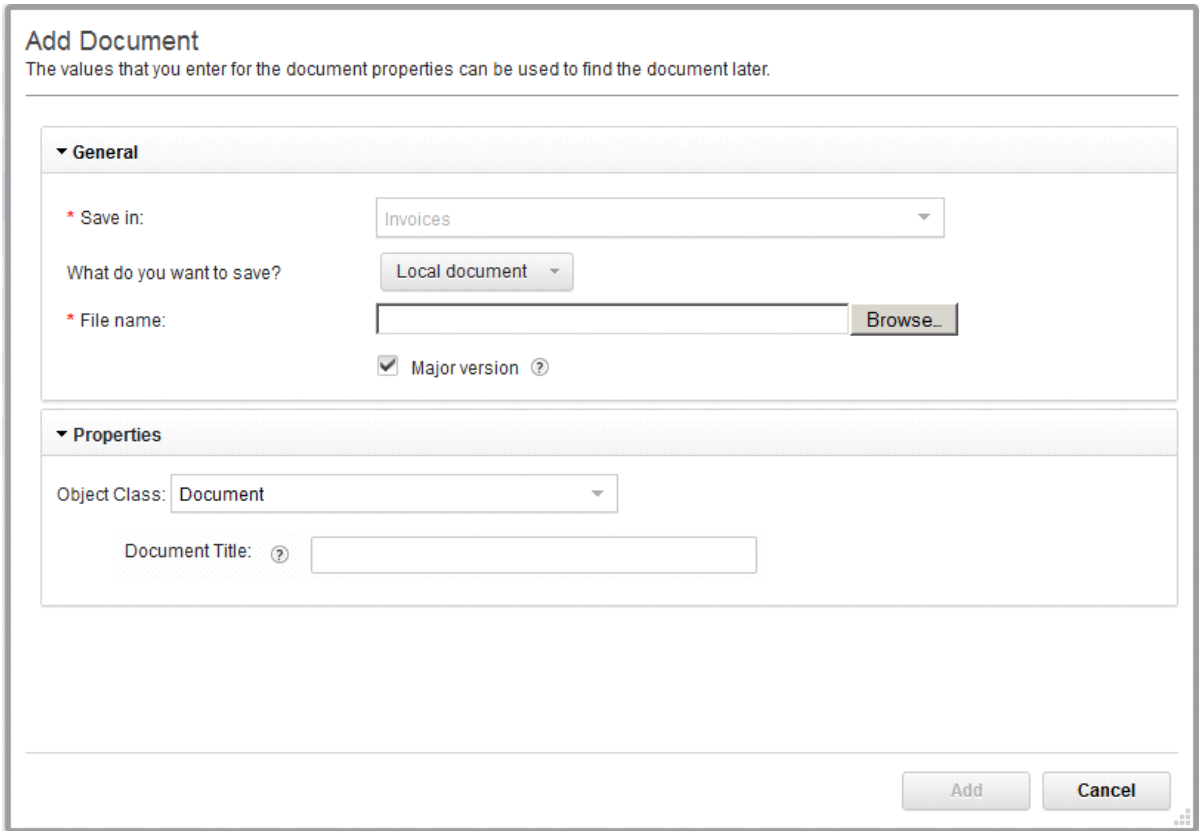
After logging in, select the **Browse** icon on the left hand side of the screen.

Expand the **Invoices** folder.

Select any document in the list. Notice that the label now says **Object Class** in the **Properties** section on the right hand side of the Browser.



Next click the **Add Document** button. Notice that the label here also now says **Object Class**.



Lab Activity 5 –External Data Services

In this chapter we introduce a usage scenario leveraging the capabilities that are provided by the External Data Service (EDS) which enables IBM Content Navigator to load data dynamically from external sources such as database tables, JSON files, or any other source of data you have in your enterprise. Some examples of the how you can integrate the data include managing property behavior by prefilling some property values, populating choice lists, setting a field as required, and validating property values input by the user in real-time.

An Example Scenario

The insurance company handles property, casualty, and auto policies. All documents relating to policies are stored in the insurance company's Enterprise Content Management (ECM) repository.
Inbound correspondence is scanned, indexed, and stored in a folder-based filing system by document type within a policy number of a unique client number.
The filing of inbound documentation is dependent upon the assignment of correct values to the inbound document. An invalid or misspelled word means that the document will not be filed correctly. Incorrectly filed documents do not enter the appropriate business process leading to delays and additional costs.
In this scenario we'll use external data services to manage the consistency of data entered during the post scanning indexing phase of in bound document capture.
There are several use cases that can be met by leveraging EDS capabilities:

Provide a choice list for all the estimators for a car insurance claim. This can be a list of people stored in a database table external to the IBM ECM system.

Provide a dependent choice list of categories belonging to one main category. This is to limit the available categories underneath each main category. Some example categories might be regions, with the subcategories being cities with offices.

Validate entered claim numbers by format, e.g. if they contain the exact number and type of characters.

Hide input property field to enter a custom text for the reason an insurance claim was opened if none of the provided ones out of the choice list would make sense.

Sample Project Description

In the next few sections we will be creating a updating the sample EDS implementation that is provided with IBM Content Navigator. The sample EDS implementation can be found under the path: **C:\Program Files\IBM\WebSphere\AppServer\profiles\AppSrv01\installedApps\P8Node01Cell\sampleEDSService_war.ear**

The sample EDS service provided with Content Navigator is a simple web application that

contains two Java servlet classes. The two servlet classes are **UpdateObjectTypeServlet** and **GetObjectTypesServlet**. The **GetObjectTypesServlet** provides an HTTP GET service to determine the list of object types supported by this EDS implementation. The **UpdateObjectTypeServlet** provides the HTTP POST service that is used to obtain external data and dynamically change, validate and enrich metadata in real-time.
The rest of the files that are contained in the sample are JavaScript Object Notation (JSON) files holding the data returned by this EDS implementation. **ObjectTypes.json** is the key file that describes which FiletNet P8 Object classes in the IBM ECM repository should the EDS service be expected to work with. The other JSON files are sample JSON files that match to some of the Object class names in P8. These files contain the sample data that will manipulate the property values shown in Content Navigator.
For the rest of this chapter, we'll focus on a simple scenario where you will add data validation and external choice lists to the Presentation object class. The Presentation class will be validating and adding data based on session information. Please keep in mind that the sample EDS service is set up to provide EDS using "hard-coded" JSON files. In a more real-world scenario, such as the one covered in the Redbook SG248055 "Customizing and Extending Content Navigator", the services would more typically be a remote web service that returns data programmatically based on calls to a relational database or 3rdParty system.
For all of the subsequent walkthroughs, we'll be making changes to the sample EDS service which is already deployed on the image. You can find the JSON files for this service in the following directory:
C:\Program Files\IBM\WebSphere\AppServer\profiles\AppSrv01\installedApps\P8Node01Cell\sampleEDSService_war.ear\sampleEDSService.war\WEB-INF\classes

Data Validation

In this section we'll walk through an example of adding property validation.

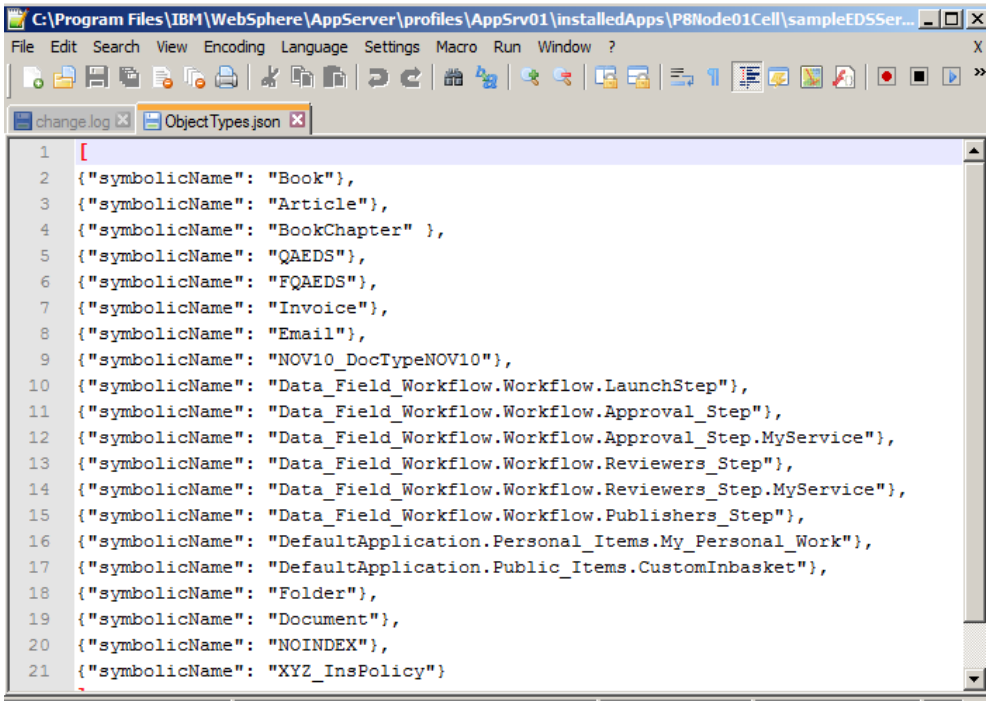
Step 1 – Update ObjectTypes.json

First, we need to update the sample ObjectTypes.json file to know that it should expect to return information anytime the Presentation object class is referenced in the Content Navigator application.

Open Windows Explorer

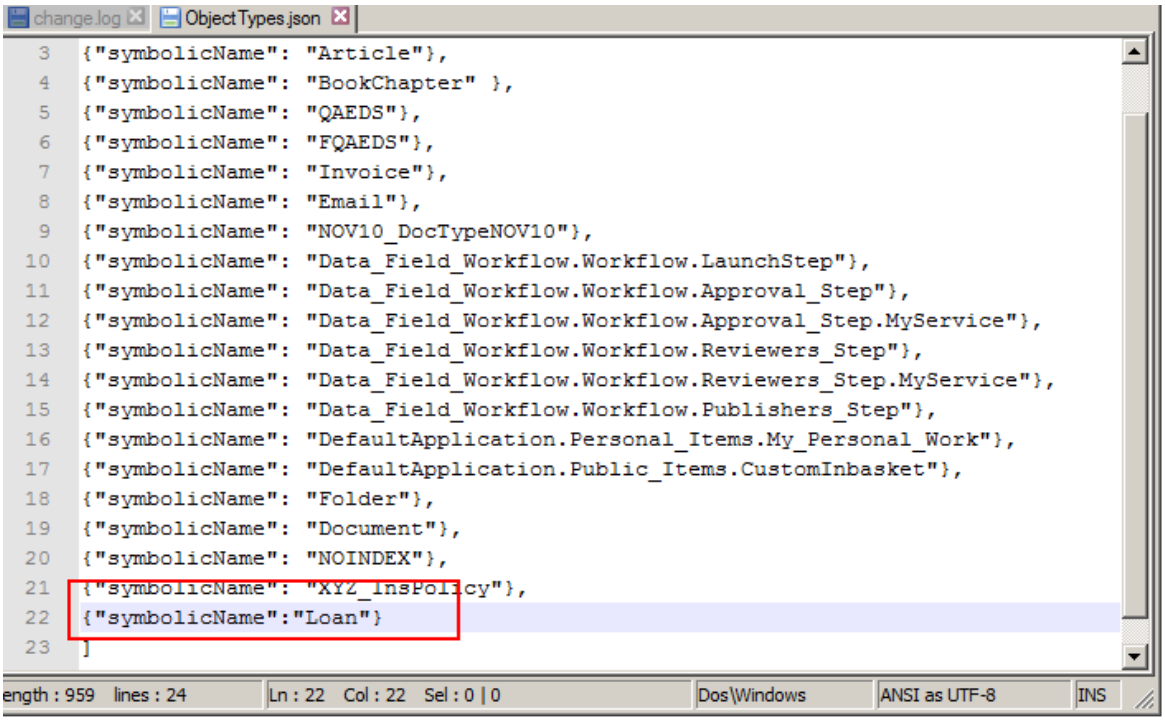
Navigate to the deployed EDS service directory at **C:\Program Files\IBM\WebSphere\AppServer\profiles\AppSrv01\installedApps\P8Node01Cell\sampleEDSService_war.ear\sampleEDSService.war\WEB-INF\classes**

Open the **ObjectTypes.json** file in Notepad. It should look something like the following.



Now find the last line with a value of **{"symbolicName": "Document"}** and append a comma on the end and then hit Enter.

Add the text **{"symbolicName": "Loan"}** to the file. The file should now look like the following screen.



Save and close the file.

You've now updated the EDS service to instruct the sample Java servlet to respond to any request from Content Navigator for data for the Object class with the symbolic name of "**Loan**".

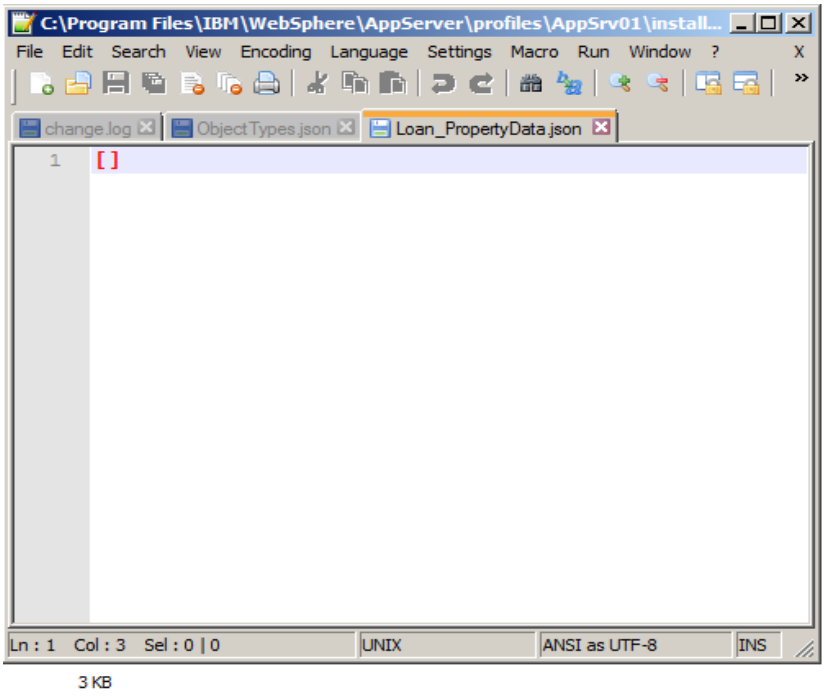
Step 2 – Add Validation for the Session Number

Next, we'll add data validation for the **LoanNumber** property of the **Loan** class.

Open Windows Explorer

Navigate to the deployed EDS service directory at **C:\Program Files\IBM\WebSphere\AppServer\profiles\AppSrv01\installedApps\P8Node01Cell\sampleEDSService_war.ear\sampleEDSService.war\WEB-INF\classes**

Open the Loan_**PropertyData.json** file in Notepad. It should look something like the following.



After the **[** character hit Enter to insert a new line.

Add the following text to the file.

{

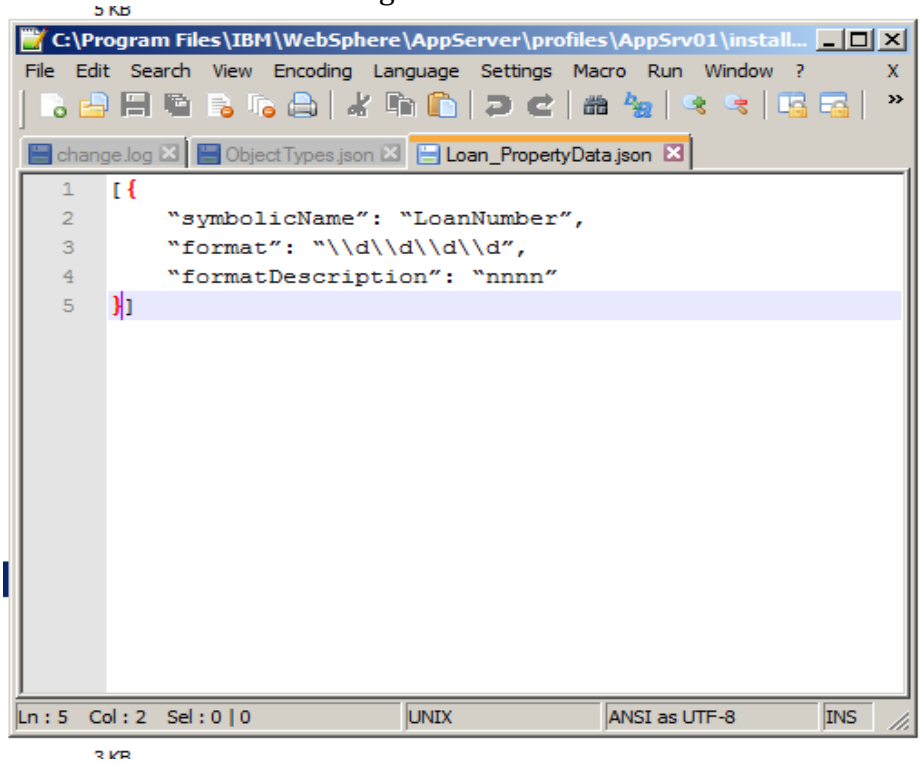
"symbolicName": "LoanNumber",

"format": "\\d\\d\\d\\d",

"formatDescription": "nnnn"

}

The file should now look like the following screen.



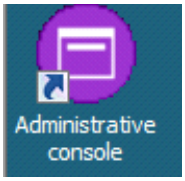
Save and close the file.

What you've just done is to add some very simple validation for the **LoanNumber** property of the **Loan** class. In our example, Loan numbers must be 4 digits long so we've added a regular expression to validate that the input value is exactly 4 digits long with no spaces, special characters, or text characters allowed. The **formatDescription** is the text that will be displayed to the user when they input an invalid value. That is basically the value that will be displayed to help the user know what format is expected.

Step 3 – Restart the sampleEDSService Web Application

Next, you'll log in to the WebSphere Administration console to restart the deployed sampleEDSService application. This will then force the application to reload the JSON files so that they will be applied when we next log in to Content Navigator.

Double click on the **Administrative Console** icon on the desktop to launch the WebSphere Application Server administration.

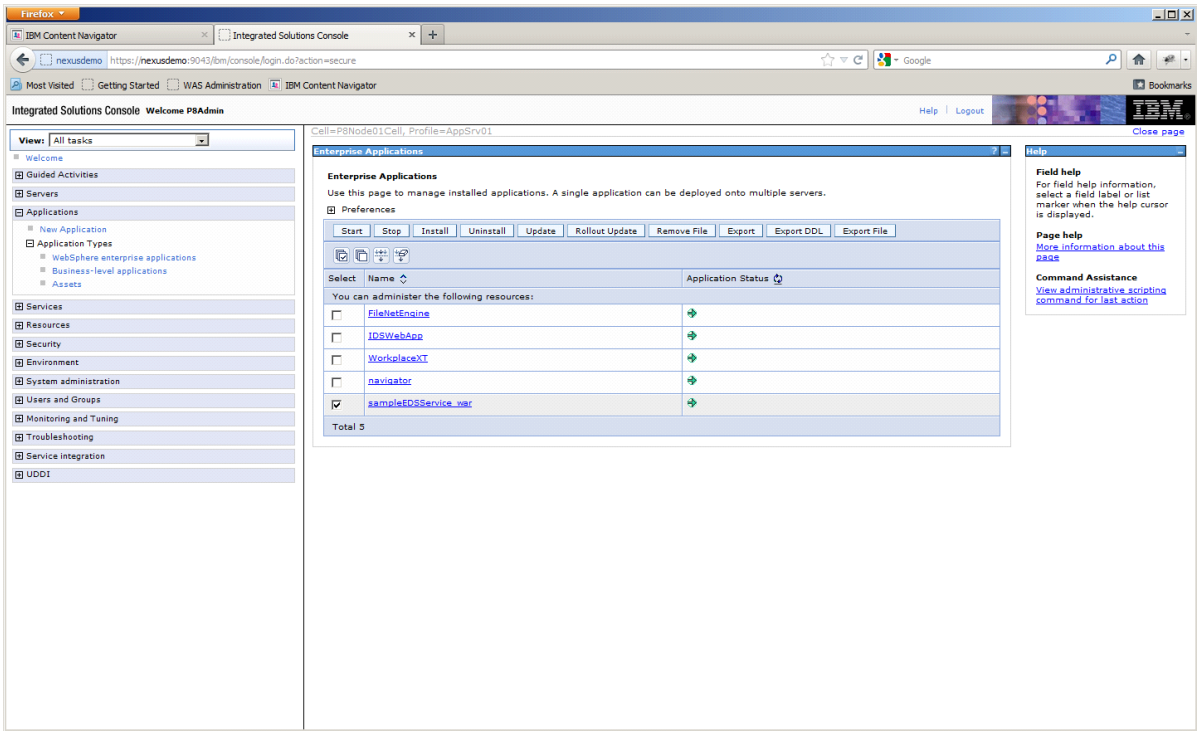


Enter **p8admin** as the user and **IBMFileNetP8** for the password.

Expand **Applications** and then **Application Types**. Click on the **WebSphere enterprise applications** link.

Select the **sampleEDSService** application and then click the **Stop** button.

Select the **sampleEDSService** application and then click the **Start** button.



Click the **Logout** link in the upper right corner of the browser window.

Close IE

Step 4 – Test the EDS Service

Next we'll log in to Content Navigator and test out the changes we've just made by trying to add a document using the Presentation class.

Open IE by clicking on the icon in the Windows quick launch bar.

Enter **http://Base-Win2k8x64:9080/navigator?desktop=ECM** as the URL and hit **Enter**.

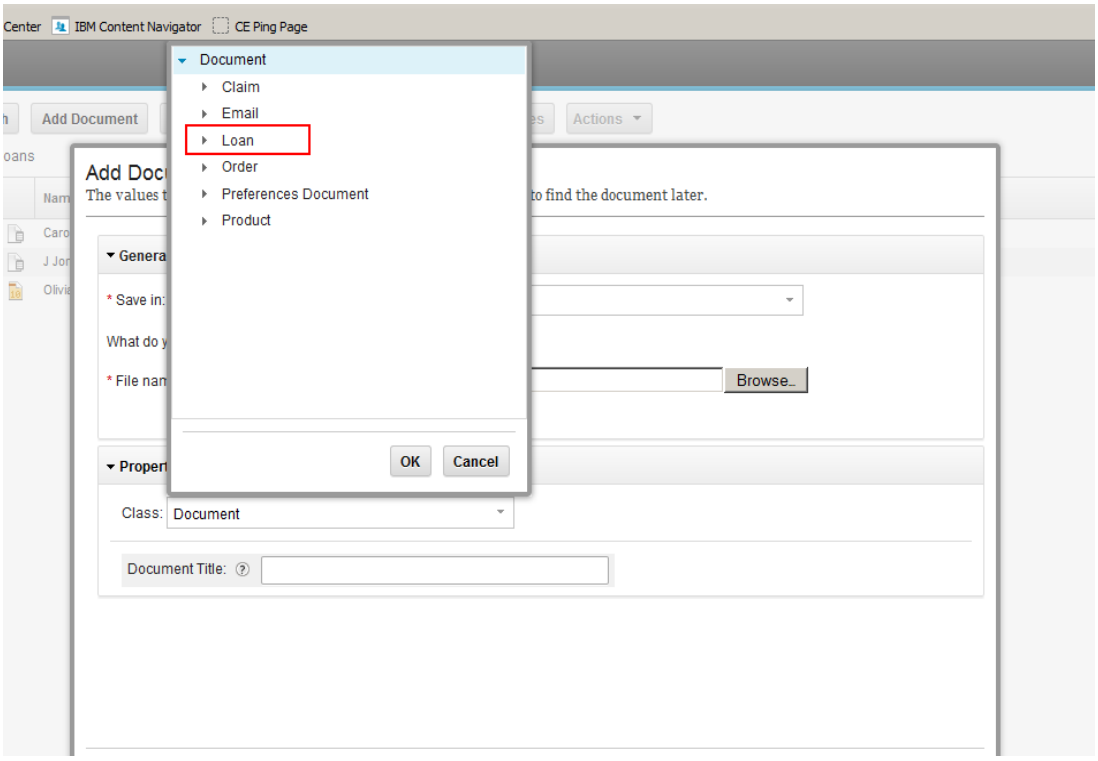
Log in using **filenetadmin** for the user and **IBMFileNetP8** for the password.

Click on the **Browse** icon on the left hand side of the screen.

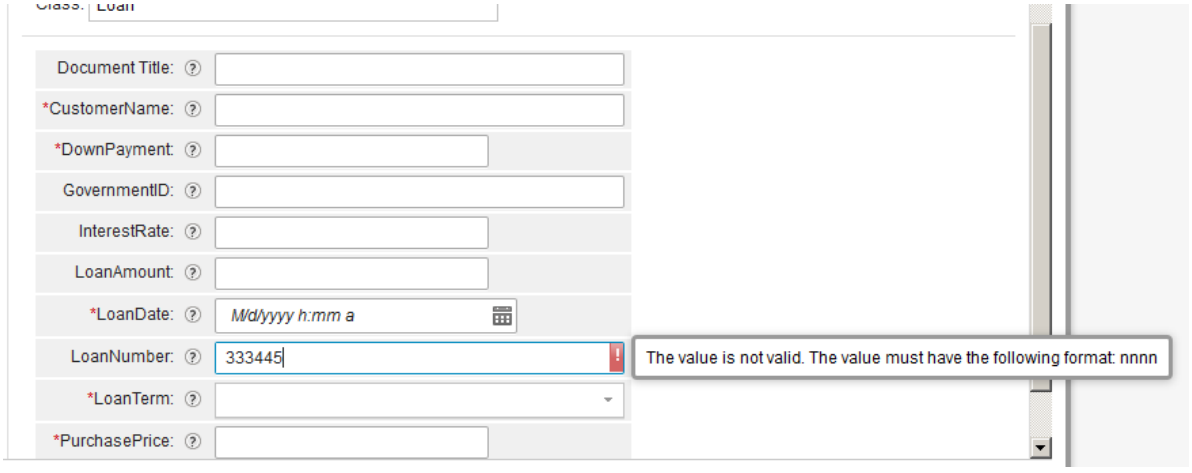
Open the **Loans** folder.

Click the **Add Document** button.

In the Object Class drop down select **Loan** and click OK.



Next click on the Loan **Number** text box. Enter the letter **f** or any other text character. You should see the field change to being marked with a red * and the help text should tell you that the value you have entered is invalid.



Try correcting the data but putting in 5 digits. You should see that an error occurs because there are too many digits.

Close IE.

Choice Lists

In this section we'll walk through an example of adding external choice lists. Choice lists are a great way to help prevent indexing errors because they limit the choices available to the user and they are pre-configured so that users cannot make any mistakes typing in data.

Step 1 – Add a Choice List for the Loan Term

Next, we'll add a choice list for the LoanTerm property of the Loan class.

Open Windows Explorer

Navigate to the deployed EDS service directory at **C:\Program Files\IBM\WebSphere\AppServer\profiles\AppSrv01\installedApps\P8Node01Cell\sampleEDSService_war.ear\sampleEDSService.war\WEB-INF\classes**

Open the Loan**_PropertyData.json** file in Notepad.

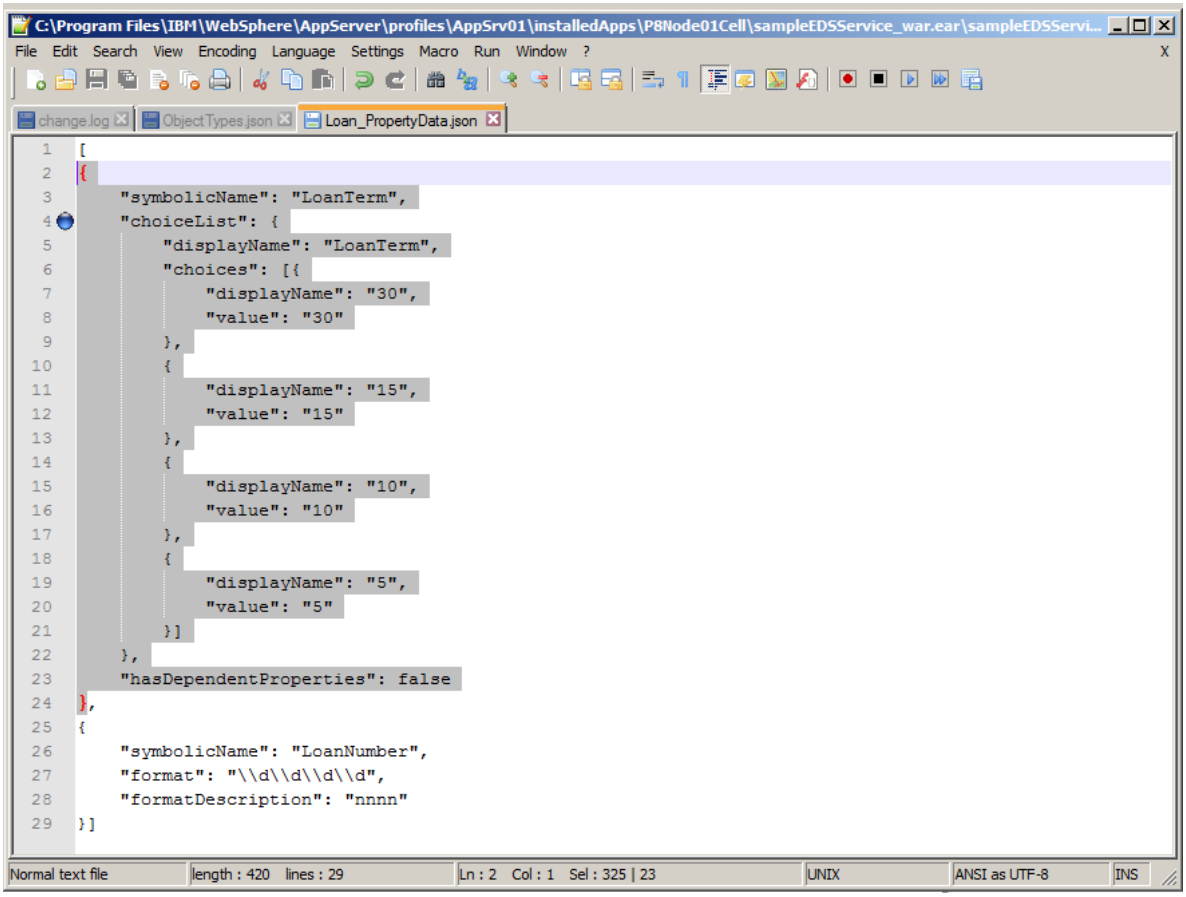
After the **[** character hit Enter to insert a new line.

Add the following text to the file.

```
{
        "symbolicName": "LoanTerm",
        "choiceList": {
                "displayName": "LoanTerm",
                "choices": [{
                        "displayName": "30",
                        "value": "30"
                },
                {
                        "displayName": "15",
                        "value": "15"
                },
                {
                        "displayName": "10",
                        "value": "10"
                },
                {
                        "displayName": "5",
                        "value": "5"
```

```
            }]
        },
        "hasDependentProperties": false
    },
```

The file should now look like the following screen.



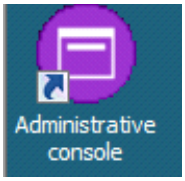
Save and close the file.

What you've just done is to add a very simple list of 4 choices from which the user can select a single value from to assign to the **LoanTerm** property.

Step 2 – Restart the sampleEDSService Web Application

Next, you'll log in to the WebSphere Administration console to restart the deployed sampleEDSService application. This will then force the application to reload the JSON files so that they will be applied when we next log in to Content Navigator.

Double click on the **Administrative Console** icon on the desktop to launch the WebSphere Application Server administration.

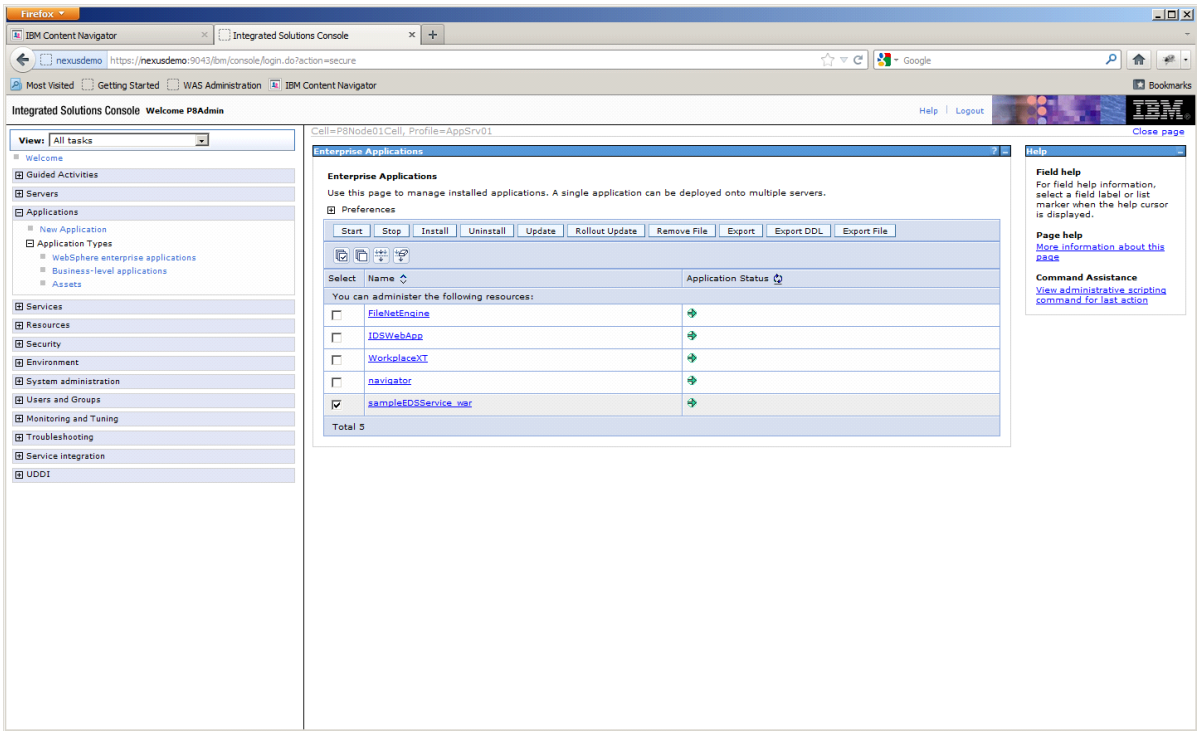


Enter **p8admin** as the user and **IBMFileNetP8** for the password.

Expand **Applications** and then **Application Types**. Click on the **WebSphere enterprise applications** link.

Select the **sampleEDSService** application and then click the **Stop** button.

Select the **sampleEDSService** application and then click the **Start** button.



Click the **Logout** link in the upper right corner of the browser window.

Close IE

Step 3 – Test the EDS Service

Next we'll log in to Content Navigator and test out the changes we've just made by trying to add a document using the Presentation class.

Open IE by clicking on the icon in the Windows quick launch bar.

Enter **http://Base-Win2k8x64:9080/navigator?desktop=ECM** as the URL and hit **Enter**.

Log in using **filenetadmin** for the user and **IBMFileNetP8** for the password.

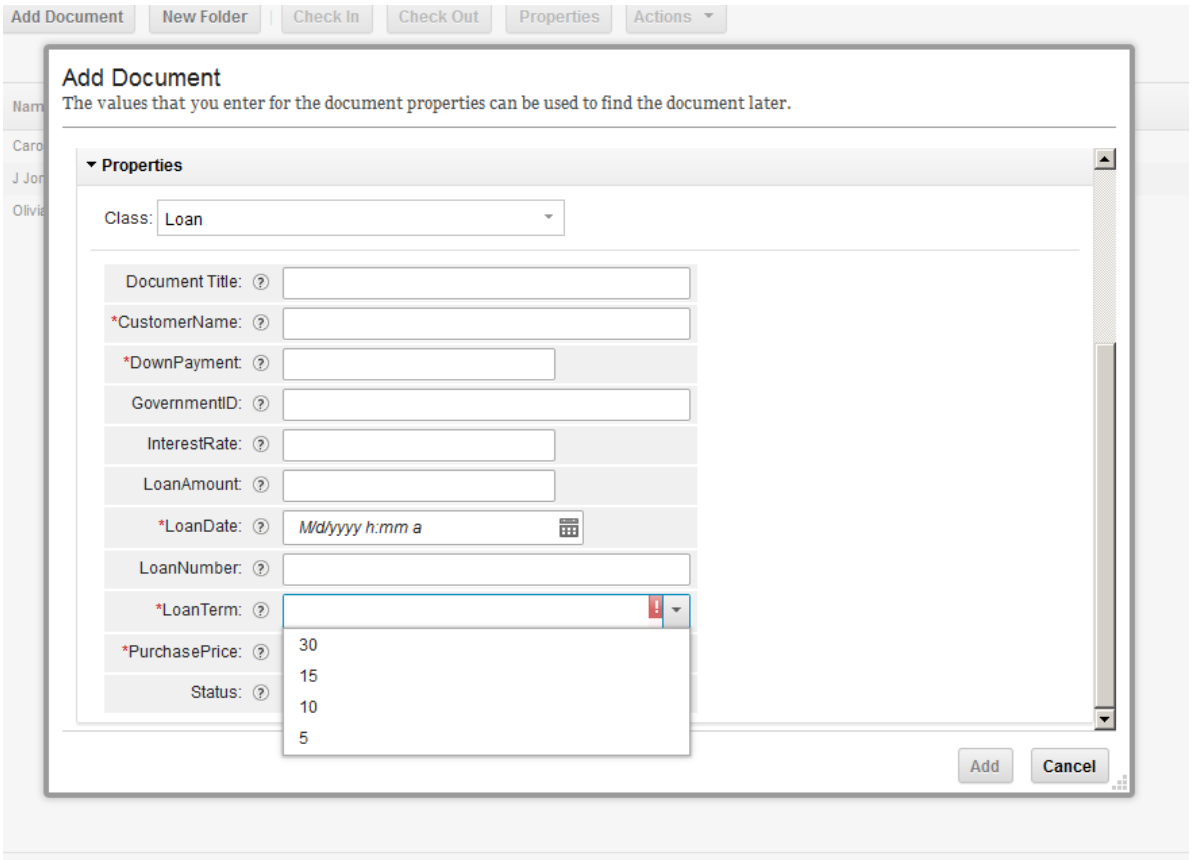
Click on the **Browse** icon on the left hand side of the screen.

Open the **Loans** folder.

Click the **Add Document** button.

In the Object Class drop down select **Loan** and click OK.

Next click on the Loan **Term** property. This property should now have a down arrow allowing you to select from the 4 choices you entered in step 1.



Close IE.

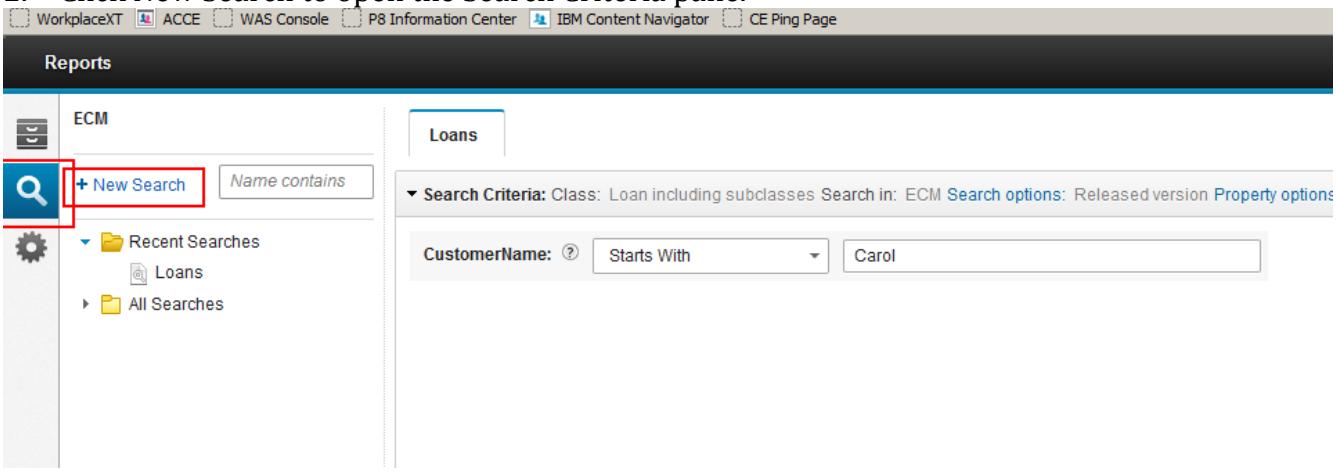
Dependent Choice Lists

In this section we're not going to specifically add any additional EDS customizations. For an example of how to set up a Choice List dependency, you can open up the Invoice_PropertyData.json file in the . **C:\Program Files\IBM\WebSphere\AppServer\profiles\AppSrv01\installedApps\P8Node01Cell\sampleEDSService.ear\sampleEDSService.war\WEB-INF\classes** directory. The specific example of dependent choice lists is between the **Region** and **BranchOffice** properties. When the user selects a **Region**, the **BranchOffice** choice list is updated to reflect choices that are valid for the given region. Try adding a document to the **Invoice** object class to see an example of this powerful capability in action.
There is also additional information in the Content Navigator programming guide that details the other capabilities of Content Navigator's EDS implementation.

Lab Activity 5 – Search

To create a new search, complete the following steps:

1. Click the Search view icon (it looks like a magnifying glass) if you do not already have the Search Criteria pane open.
2. Click New Search to open the Search Criteria pane.



3. Click Class at the top left of the Search Criteria pane.
If you select a specific document class, you can sort through all of the content in the repository but limit the search to a certain document class. For example, if you want to search and find only loans, you select Loan from the document class list; the search results show only documents that are a part of the Loan document class.

Lab Activity 6 – Developing Plug-ins

A Content Navigator plug-in allows you extend the product to meet any specific business needs. By creating a plug-ins, you can add functionality directly into the Content Navigator user interface, you can extend existing functionality, or you can completely rework the application.
A Content Navigator plug-in must implement a set of abstract Java classes that provide Content Navigator with information on the functionality provided by the plug-in as well as how that functionality should be integrated with the base product.
The code for a plug-in is packaged as a single JAR file that is then registered with Content Navigator through the Content Navigator administration. After the plug-in is registered with Content Navigator, the functionality provided by the plug-in will be available from within the base product.
A plug-in can extend and enhance virtually any aspect of the Content Navigator experience. You can create a plug-in to provide any of the following extensions:

A custom action (ex: Preview or Download as PDF)

An additional menu (ex: IBM Enterprise Records records declaration)

Entirely new features (ex: IBM Content Analytics)

Custom services (ex: IBM Content Analytics)

Additional viewers (ex: several IBM Business Partners provide custom viewers that can be integrated)

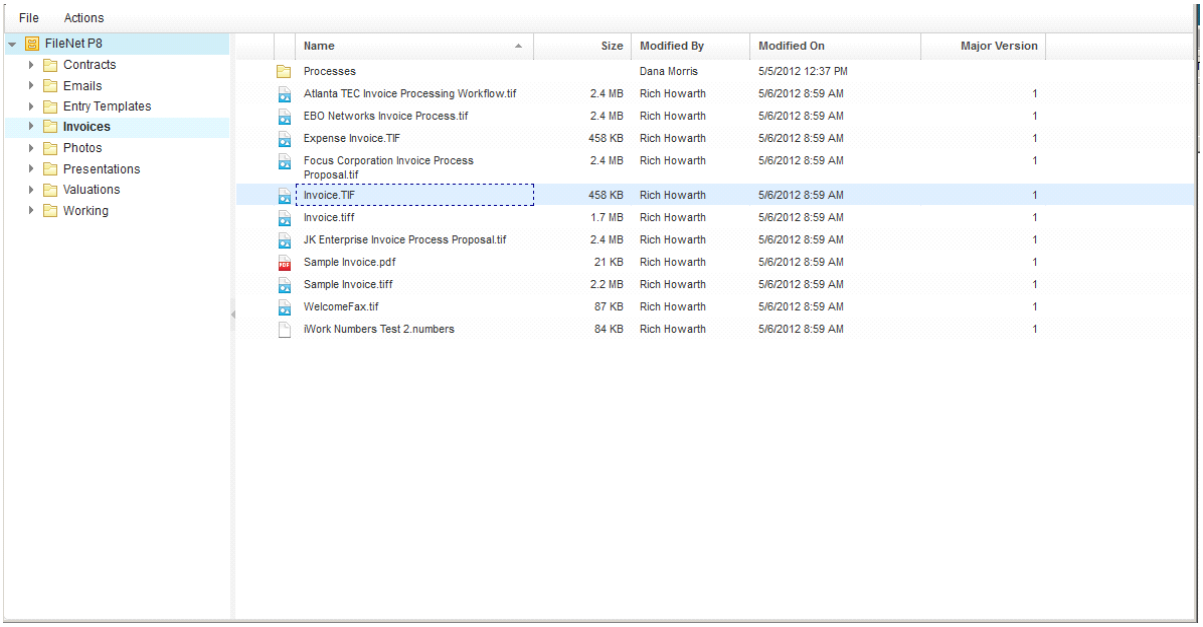
An entirely new layout

Intercept and modify the requests or responses sent (ex: External Data Services)

Custom widgets (ex: the Content Navigator tree control or add document dialog are visual widgets)

Configuration screens for setting up the plugin specific information in the Content Navigator administration

A plugin JAR file is self-contained and can be as simple as a single new action to containing everything needed for an entirely new set of services, widgets, and layout. The contents of a plug-in are always defined through the Plugin class that belongs to every plug-in project.
For more details on any of the above methods of extension, refer to the IBM Content Navigator Programming Guide or the Customizing and Extending IBM Content Navigator redbook.
For the remainder of this exercise, you'll walk through an example of creating and developing a new Plug-in to deploying it in Content Navigator for testing. This particular plugin will create a new Microsoft Windows Explorer style layout that will look like the following



1.1 Creating a new Eclipse Project

The first step is to create a new project in Eclipse. Eclipse is an open-source Java and web development Integrated Development Environment (IDE). We'll be using Eclipse for much of the remainder of this lab.

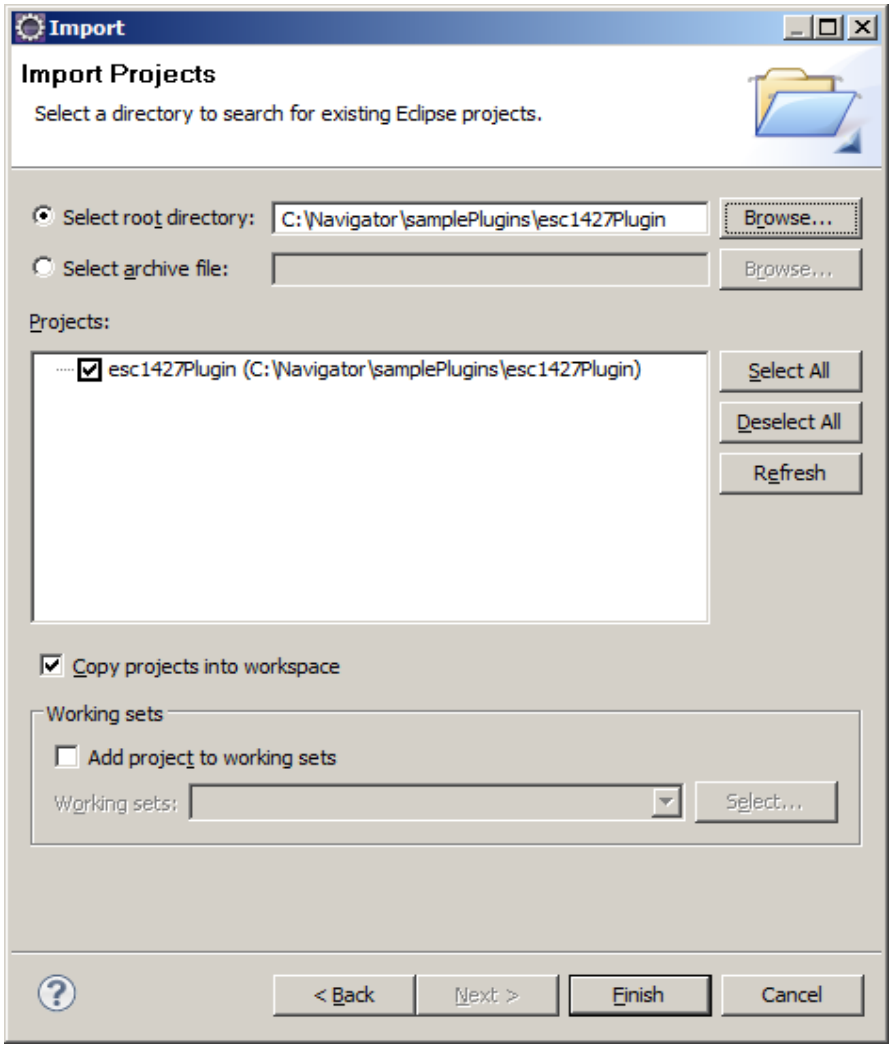
Step 1 – Open the Eclipse IDE

Double-click on the **Eclipse** icon on the desktop.

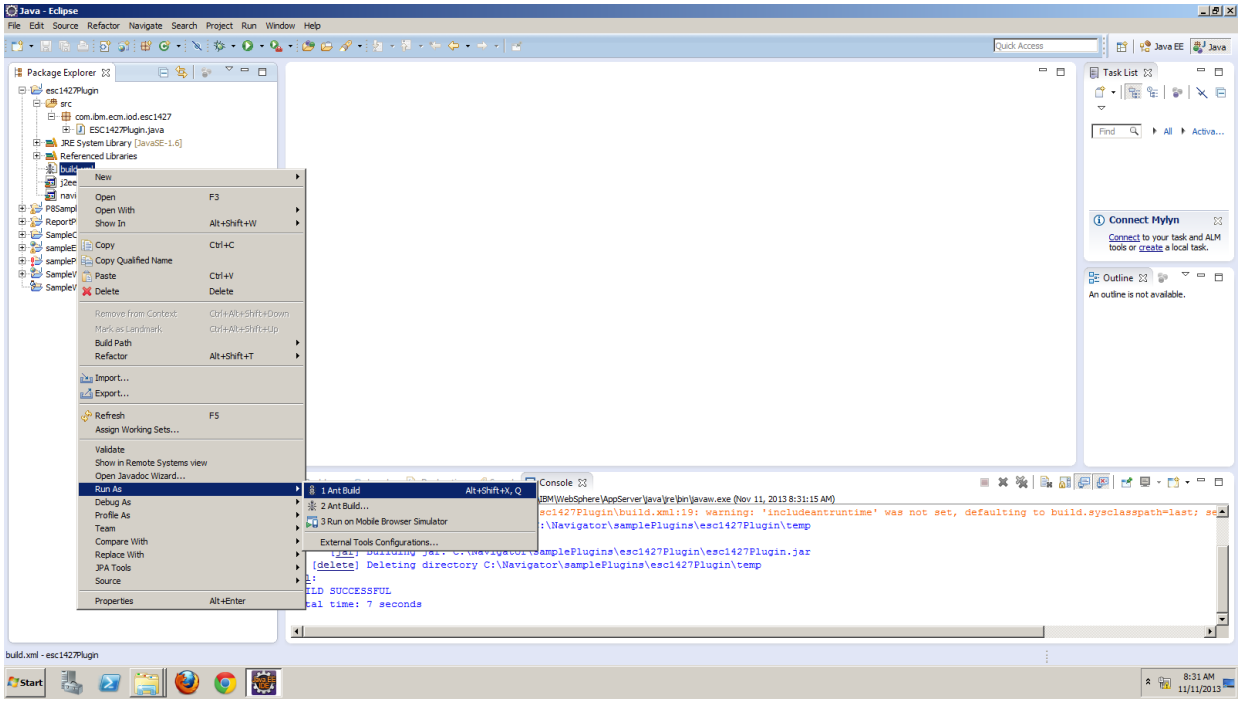
Step 2 – Create a new Java Project

Next, we'll import an existing project that will set up the basics for us.

1. Click on **File->Import**
2. Expand the General folder and click on **Existing Projects into Workspace**
3. On the next screen click the "**Browse**" button and select the **esc1427Plugin**. (The plugin is in C:\Navigator\sampleplugsin folder)



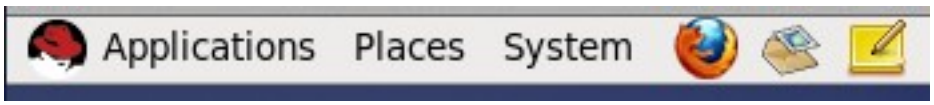
4. Click **Finish**.
5. Expand the **src** folder and the subfolders.
6. Build the plugin jar file by right clicking on build.xml → Run As → Ant build



Step 3 – Register the plug-in in IBM Content Navigator

In IBM Content Navigator 2.0.2 the plug-in administration now supports the ability to provide a project directory and plug-in class to the class loader. This feature allows changes to your plug-in to reflect immediately in IBM Content Navigator, so you can actively test your plug-in code during development. In previous releases, you had to package your plug-in as a JAR file prior to registering it in IBM Content Navigator. For this exercise, we will register this new plug-in now.

1. Click on the Firefox icon in the top bar:

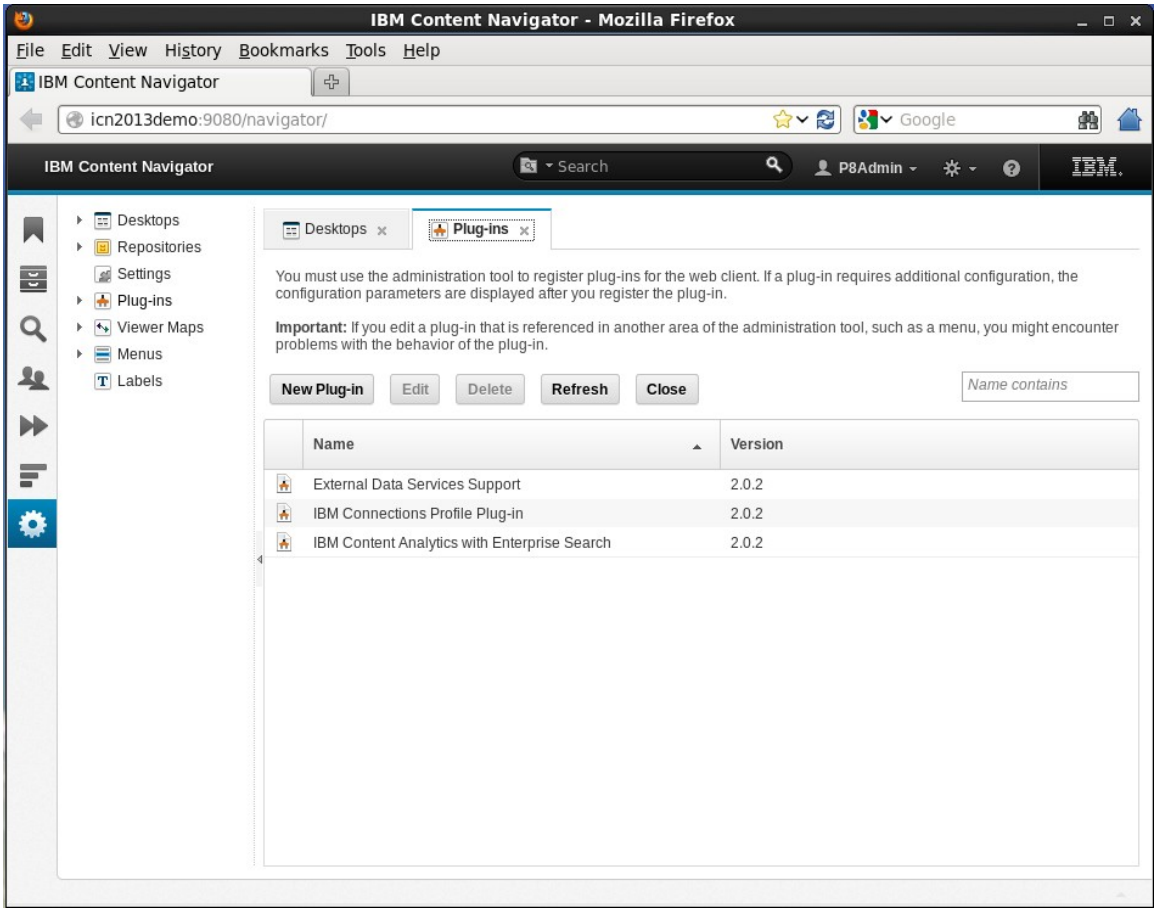


2. In FireFox, select “Bookmarks” and click on “IBM Content Navigator”

3. Login to IBM Content Navigator with user “**filenetadmin**” and password “**IBMFileNetP8**”.

4. Click on the “**Administration**” feature, which is the “gear” icon at the bottom of the navigation bar on the left side of the IBM Content Navigator User Interface.

5. Click on “**Plugins**”:

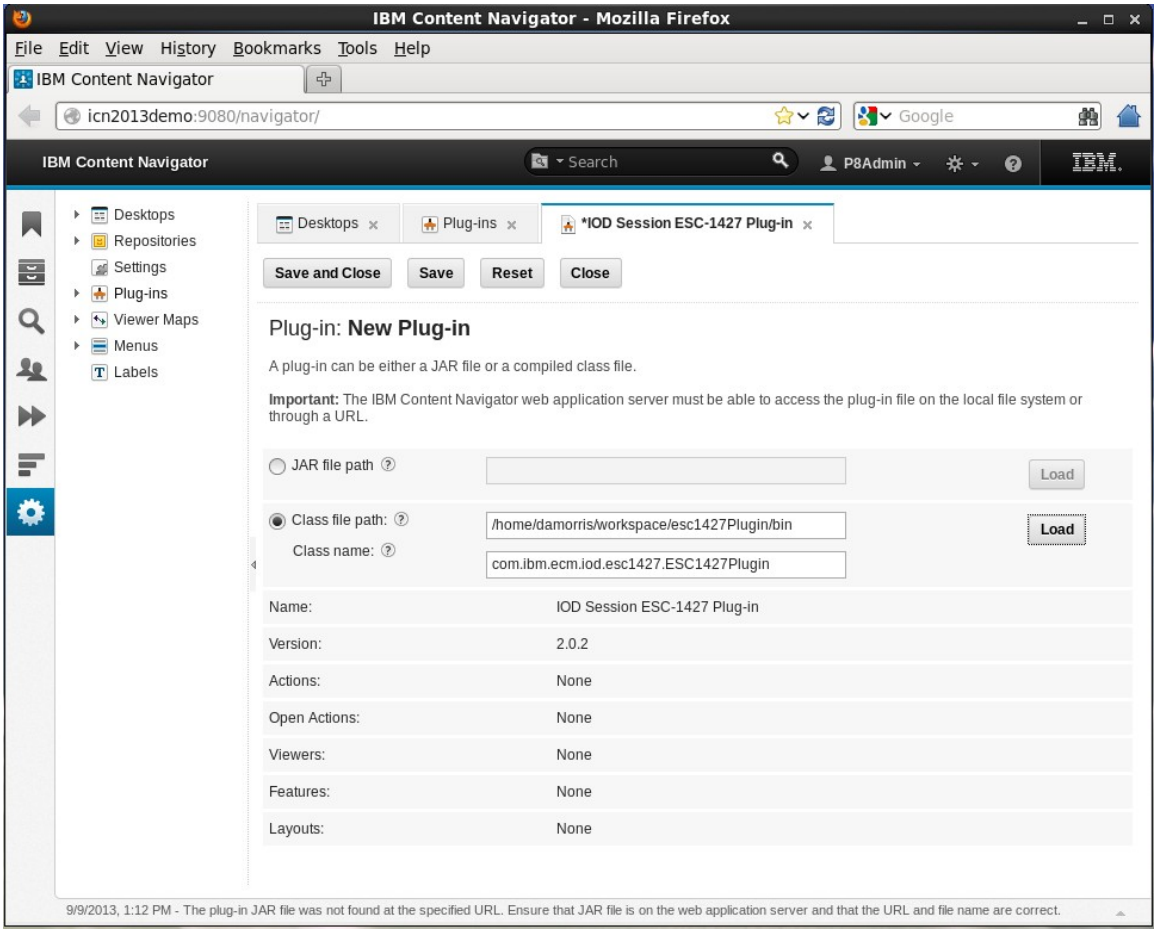


6. Click on "**New Plug-in**"

7. On the "**New Plug-in**" screen, select the radio option called "**Class file path**".

8. Enter "**C:\Users\Administrator\workspace\esc1427Plugin\bin**" for the "Class file path" and "**com.ibm.ecm.iod.esc1427.ESC1427Plugin**" for the "Class name".

9. Click the "**Load**" button:

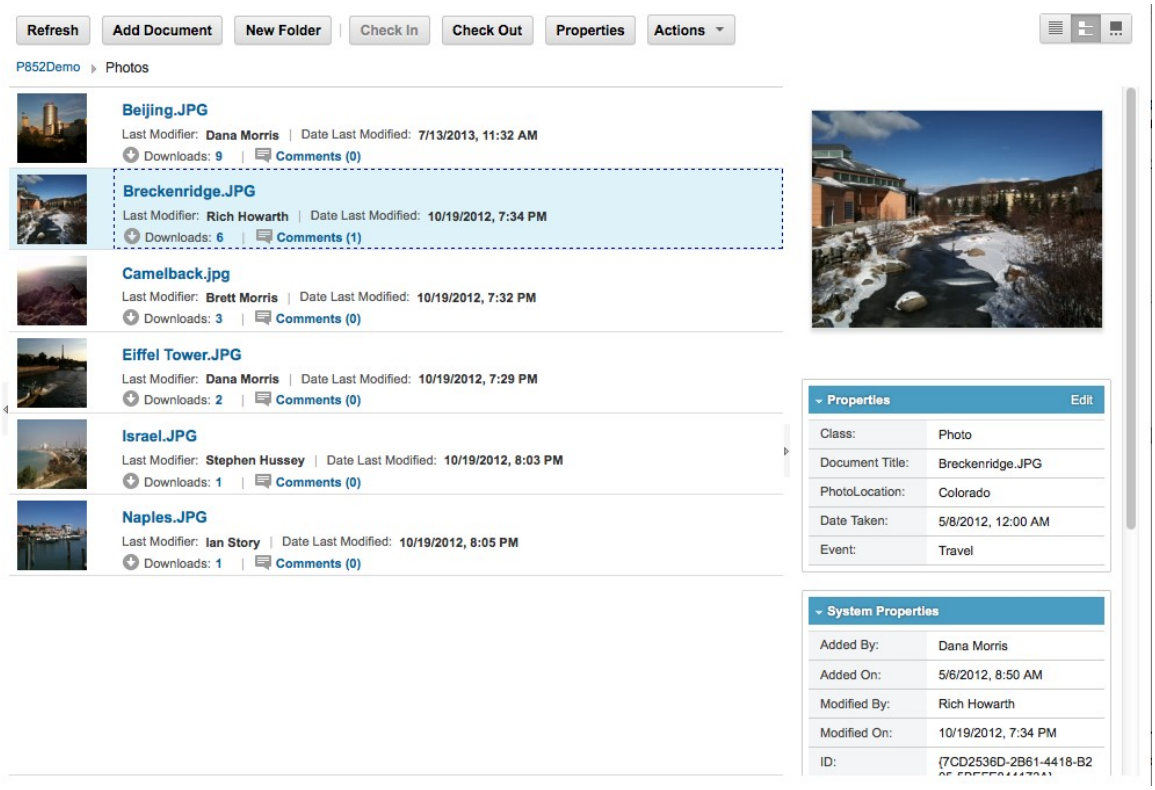


10. Click "**Save and Close**".

1.2 ContentList and Document information custom formatting

In the next step, we'll be importing various project files with the goal of enabling custom formatting of a property in both the ContentList and the Document information area. The ContentList is a DOJO Widget in the IBM Content Navigator toolkit, responsible for displaying lists of objects. It is used extensively throughout the IBM Content Navigator web application and provides a significant number of customization options. In this exercise we will demonstrate a simple customization that will format numbers in a property as currency.

The document information view is a module in the ContentList that is used to display property information when you select an object in the ContentList. The Document information DOJO widget provides a mechanism for customizing property display, similar to the ContentList. In this exercise we will apply the "currency" formatting to the property in this view as well. Here is an example of the ContentList and Document information area in IBM Content Navigator:



The following steps will walk you through importing the sample files as well as describing what purpose each file / code snippet serves. At each step you should review the code to better understand how the plugin comes together.

Step 1 – The Plugin Java Files

The PluginResponseFilter Java files that will enable our custom formatters in the ContentList and Document Information area. A PluginResponseFilter is an extension that allows customization of the JSON response from the IBM Content Navigator server. In this case, we want to modify the structure of the ContentList to apply a "**decorator**" to the "**TotalCost**" property display. A decorator is a JavaScript method that alters the display of a property within the ContentList. In addition, our response filter will apply a custom "**propertyFormatter**" to the property displays in the document information area. The "propertyFormatter" is JavaScript method that formats the display of a value in the document information property grid.

The three Java files will enable the response filters for the search, open folder and open class actions in IBM Content Navigator.

The **esc1427Plugin.js** file is the base JavaScript file that will be loaded when IBM Content Navigator loads the plug-in. You can use this to JavaScript file to apply any global changes (such as a style override) or load any JavaScript classes that need to be available throughout the session. In this case, we're adding the JavaScript class to load our decorator JavaScript, which provides global methods that will be used within the ContentList to format our custom property.

The newly added files and the enhanced ESC1427Plugin.java file will enable two new response filters in the IBM Content Navigator application and these will insert additional entries in the JSON response from the server that will tell the IBM Content Navigator DOJO widgets to leverage the custom esc1427Decorator.js (for the ContentList) and the esc1427PropertyFormatter.js (for the Document information area) when displaying the "**TotalCost**" property.

Step 2 – Validating the plug-in

Next, we'll validate the plug-in is working.

Login to the admin desktop: http://localhost:9080/navigator?desktop=admin

1. Click on the "**Administration**" feature, which is the gear icon at the bottom of the navigation bar on the left side of the IBM Content Navigator User Interface. Credentials are filenetadmin/IBMFileNetP8

2. Click on "**Plugins**"

3. Select the "**IOD Session ESC-1427 Plug-in**" and select **Edit**

4. Click the "**Load**" button to make sure the plugin changes are reloaded

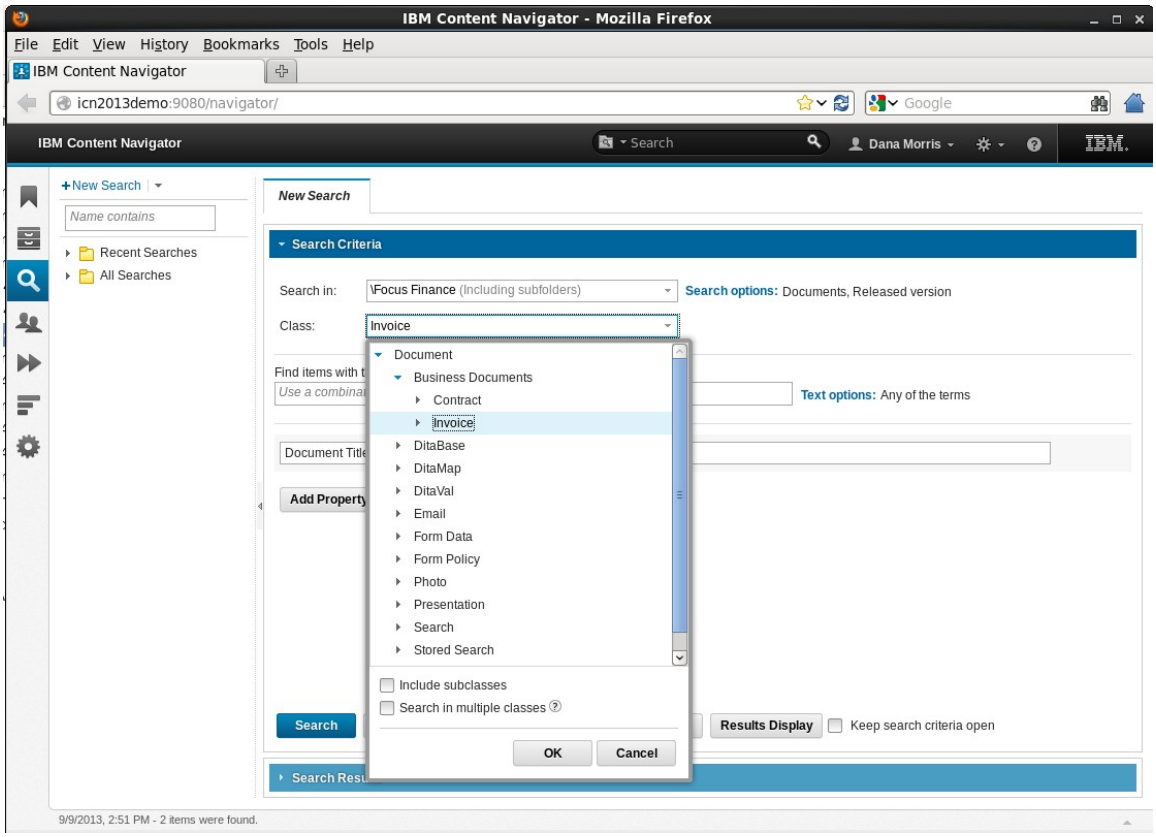
5. Log out of Content Navigator admin desktop and login to the ECM desktop

6. http://localhost:9080/navigator?desktop=ECM and click the browser refresh button to reload the page. Log back in using the user "**filenetadmin**" and password "**IBMFileNetP8**"

7. Select the "**Search**" feature by clicking on the magnifying glass icon in the navigation bar on the left side of the user interface.

8. Click on "**New Search**" located at the top of the search selector area.

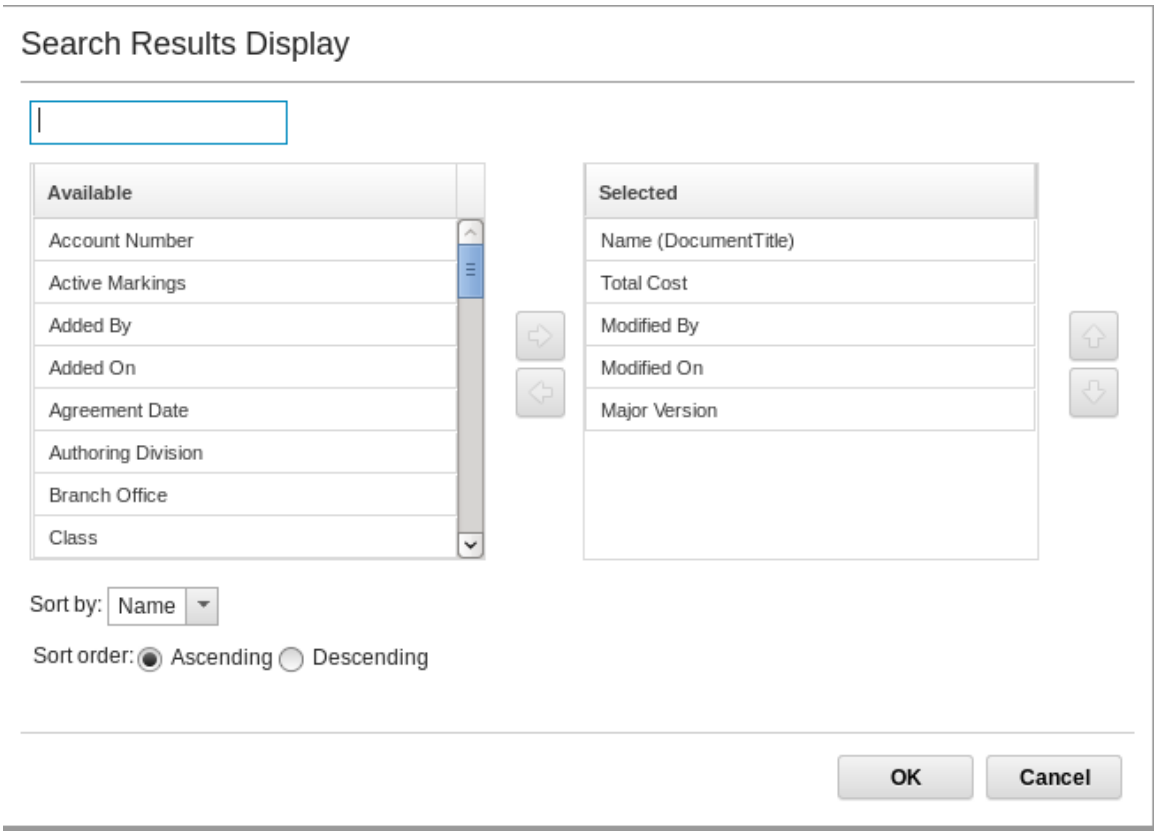
9. On the new search tab, open the class selector and navigate to the "**Invoice**" class located under Document ->



10. Select the **Invoice** class and select "**Is Not Empty**" as the search option for the Document Title property.

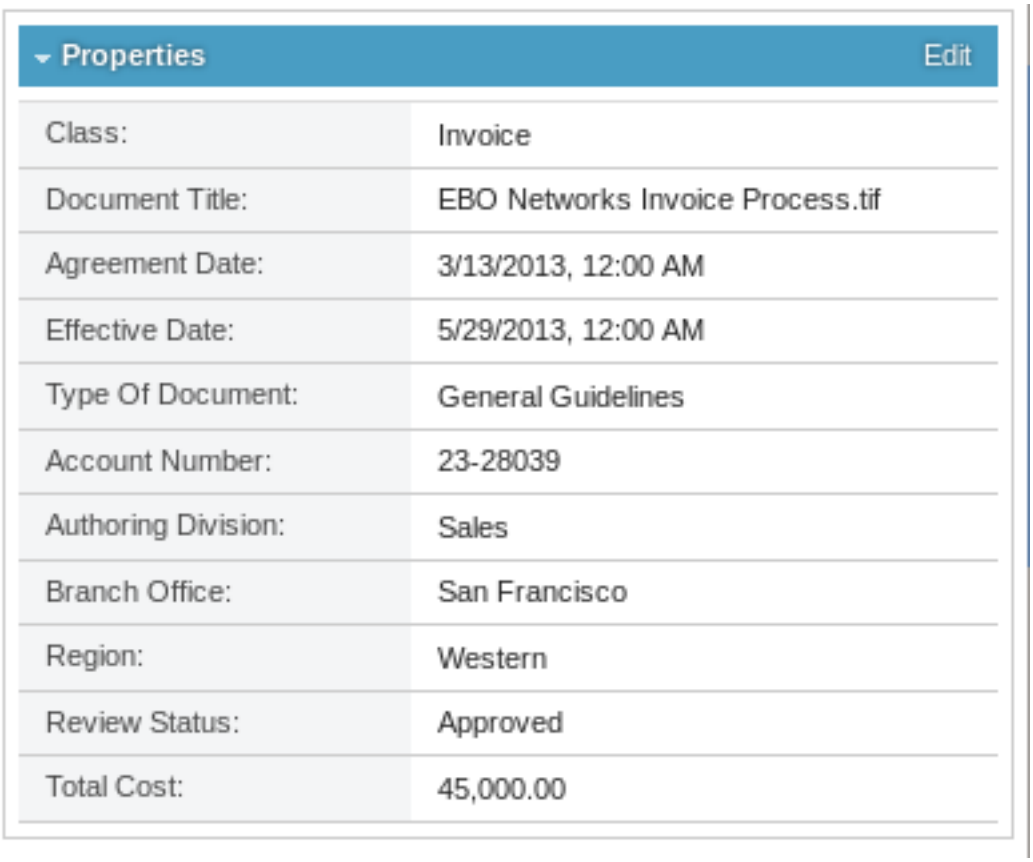
11. Click on the "**Results Display**" button.

12. Add the "**TotalCost**" property to the results display list:



13. Click "**Search**"

    In the search results you should see the Total Cost property is formatted as "currency" instead of a plain integer. If you select the document you will also see the value is formatting in the document information properties display:



1.3 Adding a custom property editor

In this step, we'll make further changes to our custom plugin to create a custom property editor for the "Photographer" property of the "Photo" class. Custom property editors can be used to control user input while adding and editing documents. This feature allows for much more granular control, by providing a mechanism for overriding the default IBM Content Navigator property editors. In this particular example, we will add a button next to the "Photographer" property that launches a user lookup dialog. The user will be forced to use this new dialog for finding and adding a user name to the "Photographer" attribute.

Step 1 – Import the Property Editor JavaScript class

1. Right-click the **com.ibm.ecm.iod.esc1427.WebContent.esc1427Dojo** folder in your project and then right click and select **Import** from the menu.

2. On the next screen, choose **File System**

3. Browse to the **"C:\Navigator\samplePlugins\esc1427Plugin_lab_6_3"** directory. Click OK and then check the box next to **esc1427PropertyEditor.js**.

   Additional Details: Importing this file will add the custom property editor DOJO widget to the plug-in. In this case, our custom property editor is an extension of the IBM Content Navigator ValidationTextBox DOJO widget, providing a button next to the standard input text area for a user lookup.

4. Click **Finish**.

Step 2 – Import the Property Editor template

1. Right-click the **com.ibm.ecm.iod.esc1427.WebContent.esc1427Dojo.templates** folder in your project and then right click and select **Import** from the menu.

2. On the next screen, choose **File System**

3. Browse to the **"C:\Navigator\samplePlugins\esc1427Plugin_lab_6_3"** directory. Click OK and then check the box next to **ESC1427PropertyEditor.html**.

   Additional Details: A template is an HTML fragment that provides the base HTML layout for a DOJO widget. In this case, our template adds the button and the style classes necessary to place the button next to the input box in the browser.

4. Click **Finish**.

Step 3 – Import the new plug-in Style Sheet

1. Right-click the **com.ibm.ecm.iod.esc1427.WebContent** folder in your project and then right click and select **Import** from the menu.

5. On the next screen, choose **File System**

6. Browse to the **"C:\Navigator\samplePlugins\esc1427Plugin_lab_6_3"** directory. Click OK and then check the box next to **esc1427Plugin.css**.

   Additional Details: Here we're adding the a set of styles necessary to position the button next to the text input in the browser.

7. Click **Finish**.

Step 4 – Overwrite the Plugin and OpenClass response filter Java class

In this step we'll also overwrite the existing Plugin Java class as well as our OpenClass response filter Java class. The new versions will enable the newly added style sheet as well as the additional logic to update the OpenClass JSON response to ensure the new custom property editor is enabled for the "Photographer" property.

1. Right-click the **com.ibm.ecm.iod.esc1427** folder in your project and then right click and select **Import** from the menu.

2. On the next screen, choose **File System**

3. Browse to the **"C:\Navigator\samplePlugins\esc1427Plugin_lab_6_3"** directory. Click OK and then check the box next to **ESC1427Plugin.java** and **ESC1427OpenClassResponseFilter.java**.

4. Click **Finish.**

Step 5 – Validate the plug-in

In this step we'll validate the plug-in is enabled by adding a new Photo and leveraging using the new custom property editor to select the photographer.

1. Open Firefox and to IBM Content Navigator again (ECM desktop) using user "**filenetadmin**" and password "**IBMFileNetP8**".

2. Click on the "**Administration**" feature, which is the gear icon at the bottom of the navigation bar on the left side of the IBM Content Navigator User Interface.

3. Click on "**Plugins**"

4. Select the "**IOD Session ESC-1427 Plug-in**" and select **Edit**

5. Click the **Load** button to make sure the plugin changes are reloaded and click **Save and Close**

6. Log out of Content Navigator and click the browser refresh button to reload the page. Log back in using the user "**filenetadmin**" and password "**IBMFileNetP8**"

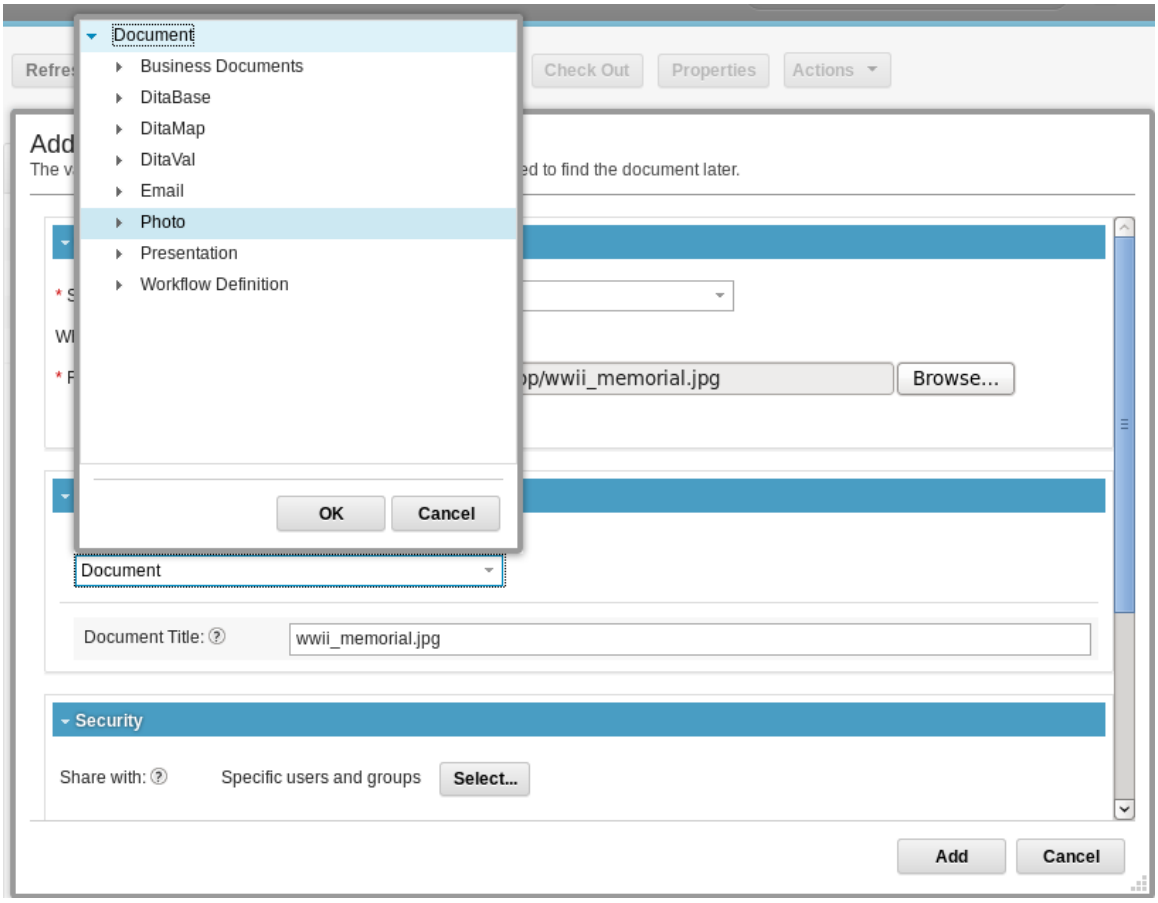
7. Select the **Browse** feature by clicking on the second icon in the navigation bar on the left side of the user interface.

8. Click on the folder named "**Photos**"

9. Click "**Add Document**"

10. Click the **Browse** button and select the file named "**WWII-Memorial020.jpg**" under C:\Users\Administrator\Downloads folder:

11. Click the "**Class**" selector and select the class "**Photo**".



After selecting the class you should see a button called “Lookup” next to the “Photographer” property. You can now click this button to select a user and add the document!

1.4 Creating the “Windows Explorer” style browse

In this step, we’ll make the final changes to our custom plugin to create a new browse feature that will provide users with a “windows explorer-like” view into the Enterprise Content Management repository. To accomplish this we will extend the existing IBM Content Navigator browse feature and add the file menu bar, similar to what you would see in a Windows program. Next, we will alter the default menus to remove actions relating to features not available in Browse. Finally we will create a new desktop that will enable our new feature and it’s menus.

Step 1 – Import the Java classes and overwrite the existing ESC1427Plugin Java class

In this step we will import the new classes that will define our custom Browse feature as well as the customized versions of the existing IBM Content Navigator “Item” menus.

1. Right-click the **com.ibm.ecm.iod.esc1427** folder in your project and then right click and select **Import** from the menu.

2. On the next screen, choose **File System**

3. Browse to the **"C:\Navigator\samplePlugins\esc1427Plugin_lab_6_4"** directory. Click OK and then check the box next to **ESC1427Plugin.java, ESC1427Feature.java, ESC1427FolderContextMenu.java, ESC1427ItemContextMenu.java, ESC1427MixItemsContextMenu.java and ESC1427SystemItemContextMenu.java**.

   Additional Details: Plug-ins have the ability to define custom menu types and toolbar types as well as the ability to define menu and toolbars. In this case, we're providing custom menus of existing IBM Content Navigator menu types. These custom menus remove some actions that will not be available in this simplified view of the repository.

4. Click **Finish.**

Step 2 – Overwrite the existing Style Sheet

In this step we will add new styles to the plugin style sheet to customize the look and feel of the browse panel.

1. Right-click the **com.ibm.ecm.iod.esc1427.WebContent** folder in your project and then right click and select **Import** from the menu.

2. On the next screen, choose **File System**

3. Browse to the **"C:\Navigator\samplePlugins\esc1427Plugin_lab_6_4"** directory. Click OK and then check the box next to **esc1427Plugin.css**.

4. Click **Finish.**

Step 3 – Add the new Browse widget

1. Right-click the **com.ibm.ecm.iod.esc1427.WebContent.esc1427Dojo** folder in your project and then right click and select **Import** from the menu.

2. On the next screen, choose **File System**

3. Browse to the **"C:\Navigator\samplePlugins\esc1427Plugin_lab_6_4"** directory. Click OK and then check the box next to **ESC1427BrowsePane.js**.

   Additional Details: The ESC1427BrowsePane is an extension on the existing IBM Content Navigator BrowsePane. The extension is necessary to add the menubar (similar to what you see in a Windows program) and alter the custom modules applied to the ContentList widget.

4. Click **Finish.**

Step 4 – Add the new Browse html template

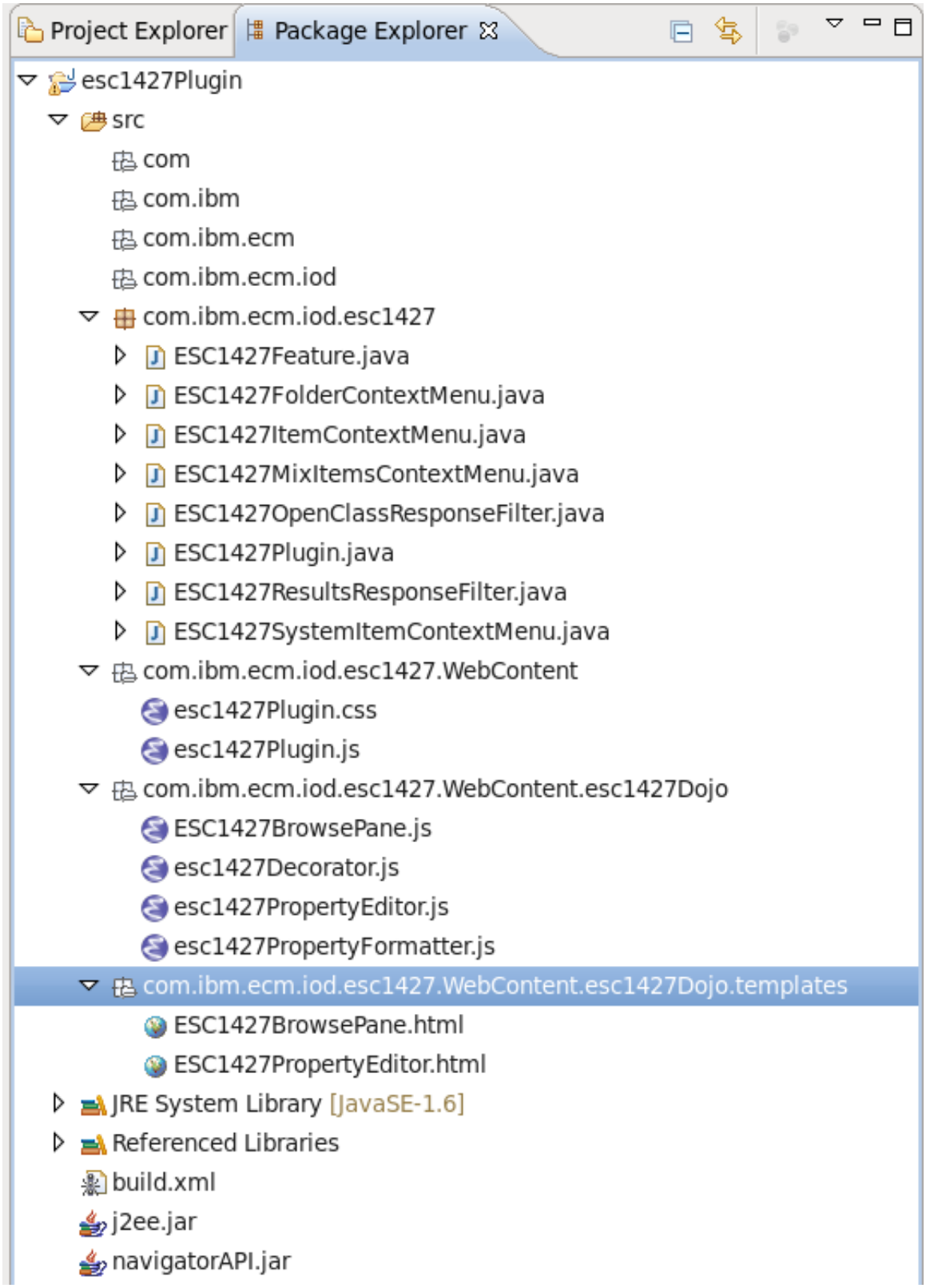
1. Right-click the **com.ibm.ecm.iod.esc1427.WebContent.esc1427Dojo.templates** folder in your project and then right click and select **Import** from the menu.

2. On the next screen, choose **File System**

3. Browse to the **"C:\Navigator\samplePlugins\esc1427Plugin_lab_6_4"** directory. Click OK and then check the box next to **ESC1427BrowsePane.html**.

4. Click **Finish.**

   After completing the first four steps, your eclipse project should look like this:


Project Explorer
Package Explorer
esc1427Plugin
src
com
com.ibm
com.ibm.ecm
com.ibm.ecm.iod
com.ibm.ecm.iod.esc1427
ESC1427Feature.java
ESC1427FolderContextMenu.java
ESC1427ItemContextMenu.java
ESC1427MixItemsContextMenu.java
ESC1427OpenClassResponseFilter.java
ESC1427Plugin.java
ESC1427ResultsResponseFilter.java
ESC1427SystemItemContextMenu.java
com.ibm.ecm.iod.esc1427.WebContent
esc1427Plugin.css
esc1427Plugin.js
com.ibm.ecm.iod.esc1427.WebContent.esc1427Dojo
ESC1427BrowsePane.js
esc1427Decorator.js
esc1427PropertyEditor.js
esc1427PropertyFormatter.js
com.ibm.ecm.iod.esc1427.WebContent.esc1427Dojo.templates
ESC1427BrowsePane.html
ESC1427PropertyEditor.html
JRE System Library [JavaSE-1.6]
Referenced Libraries
build.xml
j2ee.jar
navigatorAPI.jar

Step 4 – Setup the new desktop to enable the custom Browse feature

1. Open Firefox and to IBM Content Navigator again (http://localhost:9080/navigator?desktop=ECM) using user "**filenetadmin**" and password "**IBMFileNetP8**" and access the "**Administration**" feature.

2. Click on "**Plugins**"

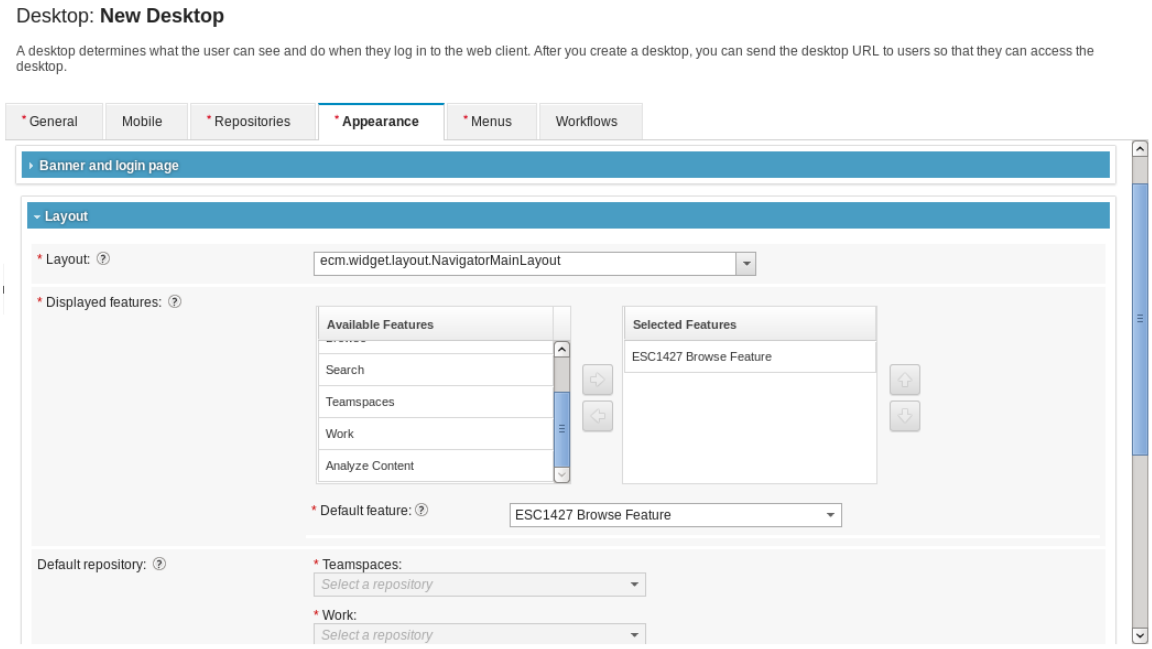
3. Select the "**IOD Session ESC-1427 Plug-in**" and select **Edit**

4. Click the **Load** button to make sure the plugin changes are reloaded and click **Save and Close**

5. Next, click on **Desktops** and click the **New Desktop** button. On the general tab, enter **esc1427** as the name.

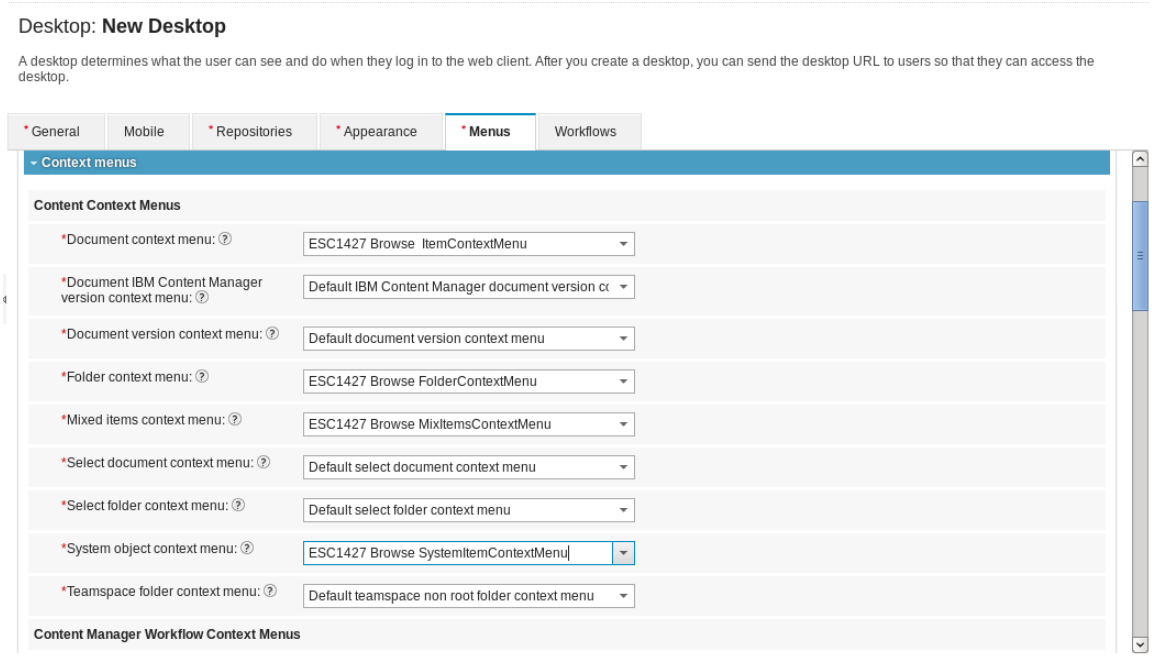
6. Under the "**Authentication**" section on the "**General**" tab, select the only repository available "**ECM**".

7. On the **Appearance** tab, enter **ESC1427** as the **Application Name** then collapse the **Banner and Login Page** section. In the **Selected Features** box, select all of the features and click the left arrow to remove them from the desktop. In the **Available Features** box, select the **ESC1427 Browse Feature** and click the right arrow to add it to the desktop.



8. Click the **Menus** tab. Collapse the **Toolbars** section.

9. Update the **Document context menu**, **Folder context menu**, **Mixed items context menu**, and the **System object context menu** so that each definition is using the **ESC1427** menu definition instead of the default entry. When you're done the desktop definition should look something like the following picture.



1. Click **Save and Close**. Log out and close the browser.

2. Reopen Mozilla Firefox. Enter http://localhost:9080/navigator/?desktop=**esc1427** to load your custom desktop.

End of Lab – Congratulations!