

# **Hands-on: Personalized Portlet Development Using WebSphere Portal Server V4.1 and WebSphere Personalization V4**

**SESSION NUMBER:** H1302

**INSTRUCTORS:** Bruce R. Olson

IBM

Software Engineer

[bolson@us.ibm.com](mailto:bolson@us.ibm.com)

Venkata (Vishy) Gadepalli

IBM

Software Engineer

[vgadepal@us.ibm.com](mailto:vgadepal@us.ibm.com)

## Lab Introduction

Start WebSphere Studio Application Developer (WSAD) and Inspect the MyPersAppWeb Project

Define User and Content Resources in the MyPersAppWeb Project

Define Content Spots in the MyPersAppWeb Project

Edit news.jsp to Add the Content Spots and Define the Layout

Export the Artifacts From the MyPersAppWeb Web Application to the File System

Export the MyPersApp.ear File

Install the MyPersApp.ear File

Verify that the Customer and Customernews Resources Have Been Defined

Start the Personalization Workspace and Add a Publishing Server

Create the MyPersApp Project

Edit the MyPersApp Project and Make it the Current Project

Create the UserLevel Classifier Rule

Create the GetCurrentUser Action Rule

Create the GetBasicNews Action Rule

Create 4 Additional News Action Rules

Create the GetCustomerNews Binding Rule

Assign the Rules to Content Spots in the Campaign Manager

Publish the Rules and Campaign

Preview the news.jsp page Using the Preview Launcher

Start WSAD and Create a New Project

[Update the Project Classpath](#)

[Edit MyPersPortletController.html.java](#)

[Import the Artifacts Previously Exported From the MyPersAppWeb Web Application](#)

[Create the MyPersPortlet WAR File](#)

[Install the MyPersPortlet WAR File Into the Portal](#)

[Log in as Several Different Users and Notice How the News is Different Depending on the User](#)

[Appendix A - Machine Setup](#)

## ***\_\_\_ Lab Introduction***

The purpose of this lab is to introduce you to the WebSphere Personalization, the support provided by WebSphere Studio Application Developer (WSAD) for Rules Based Personalization, and the integration of Personalization in WebSphere Portal Server.

We have created an exercise that starts by building a personalized web application. It demonstrates how to quickly create Resources and Content Spots using the Wizards in WSAD. The exercise guides you through adding these content spots to a JSP that you will use to add personalized content to the page. You then create rules in the Personalization Workspace and map these rules to the content spots. The exercise shows you how to preview this page in the Personalization Workspace. Once you have this working you can try logging into the web application as different users to see how the content is personalized depending on who the current user is.

The second part of this exercise guides you through converting the personalized web application into a portlet using the portlet development toolkit provided for WSAD.

This document contains instructions for running this on any machine having the pre-requisite software installed. See [Appendix A](#) before proceeding to ensure you have your machine configured correctly for this lab.

## ***\_\_\_ Start WebSphere Studio Application Developer (WSAD) and Inspect the MyPersAppWeb Project***

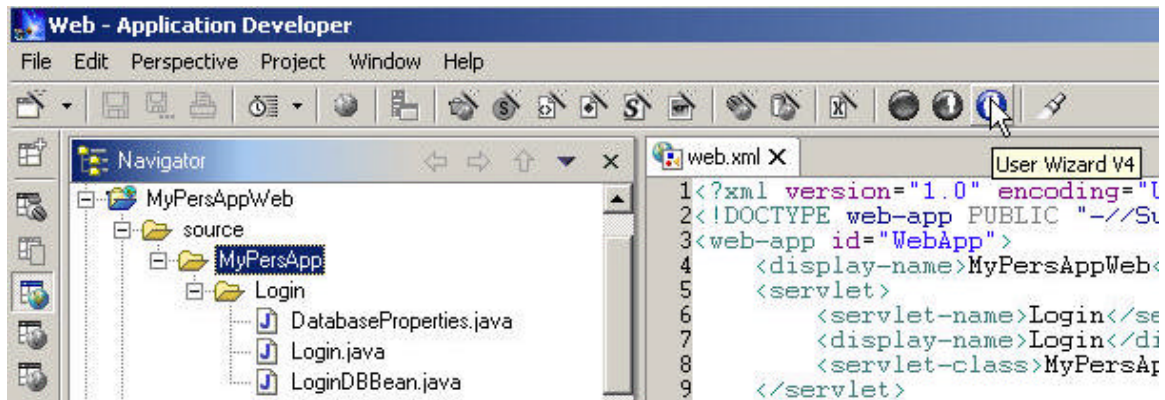
Double-click on the **WSAD** icon to start WebSphere Studio Application Developer (WSAD) V4. This is the tool we will start with in this hands-on lab. Take some time to browse the contents of the MyPersAppWeb project. We have provided you with some JSP and HTML pages, artwork, and a Login servlet so that you can spend your time working on aspects of the project directly related to personalization.

## ***\_\_\_ Define User and Content Resources in the MyPersAppWeb WSAD Project***

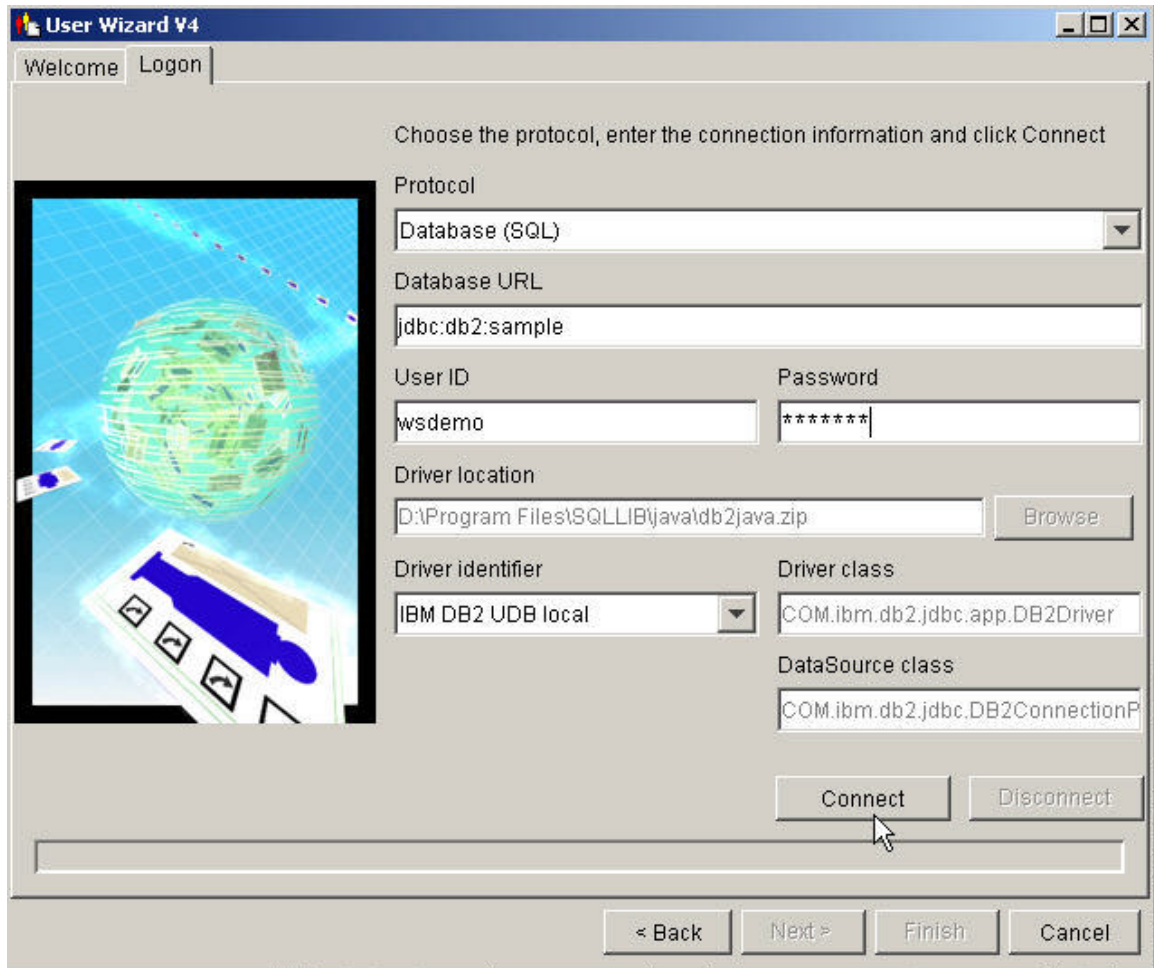
WebSphere resources are implemented as java interfaces that normalize the differences of underlying content stores. These resources make properties available to rules through methods for adding, deleting, finding, and modifying content. These resources can be created using the WSAD Wizards to wrapper existing database tables and LDAP repositories, or hand coded by a developer using WSAD or some other Java IDE.

There are 2 types of resources, User Resources and Content Resources. User resources contain attributes and characteristics of users who log on to an application that utilizes personalization. Content Resources are adapters that provide access to content that you want to conditionally show to end users, based on similarities between the content's attributes and the current user's attributes.

- Select the **MyPersApp** folder in the MyPersAppWeb project
- Click on the **User Wizard** button as shown in the following picture to invoke the User Wizard



- Fill out the Logon page in the User Wizard using the following picture as a guide. The password should be **wsdemo1**.
- Click the **Connect** button to connect to the database and populate the remaining pages in the dialog



User Wizard V4

Welcome Logon

Choose the protocol, enter the connection information and click Connect

Protocol: Database (SQL)

Database URL: jdbc:db2:sample

User ID: wsdemo Password: \*\*\*\*\*

Driver location: D:\Program Files\SQLLIB\java\db2java.zip Browse

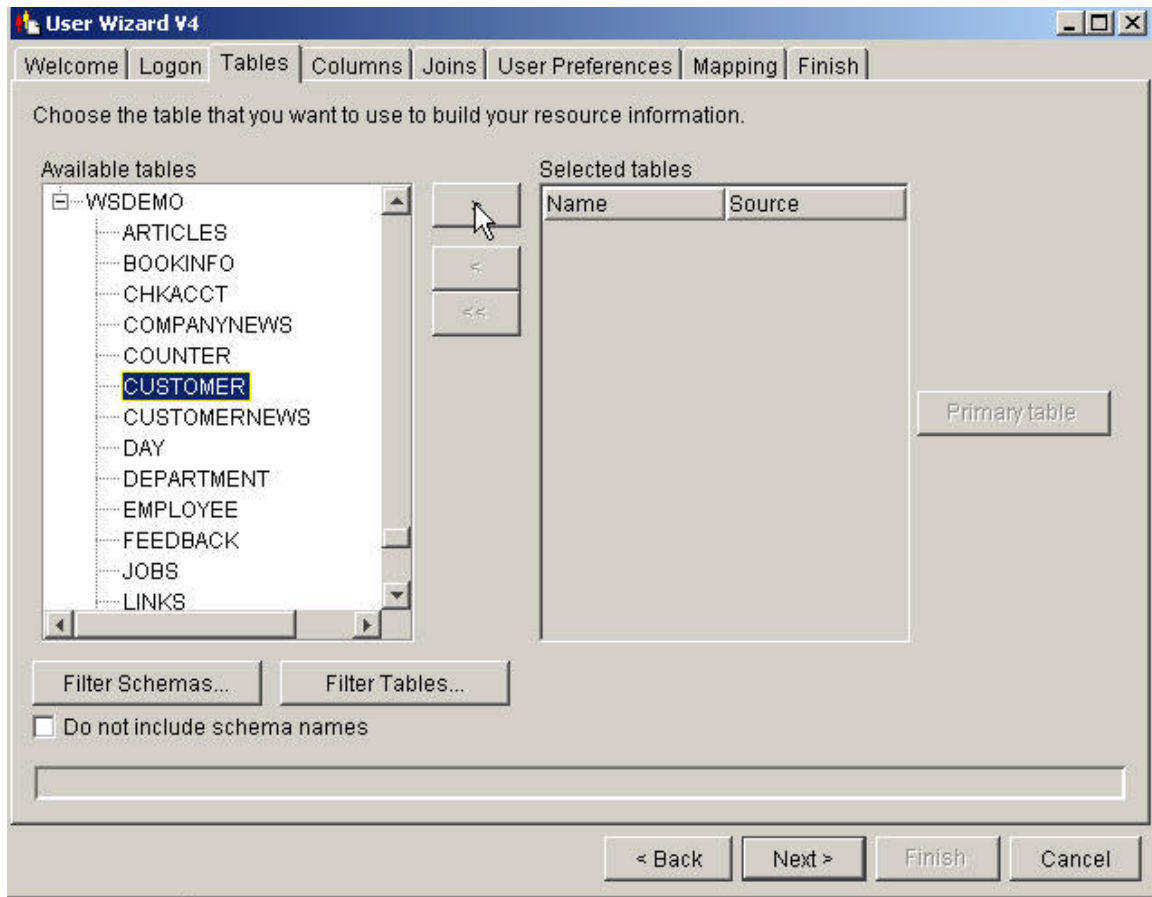
Driver identifier: IBM DB2 UDB local Driver class: COM.ibm.db2.jdbc.app.DB2Driver

DataSource class: COM.ibm.db2.jdbc.DB2ConnectionP

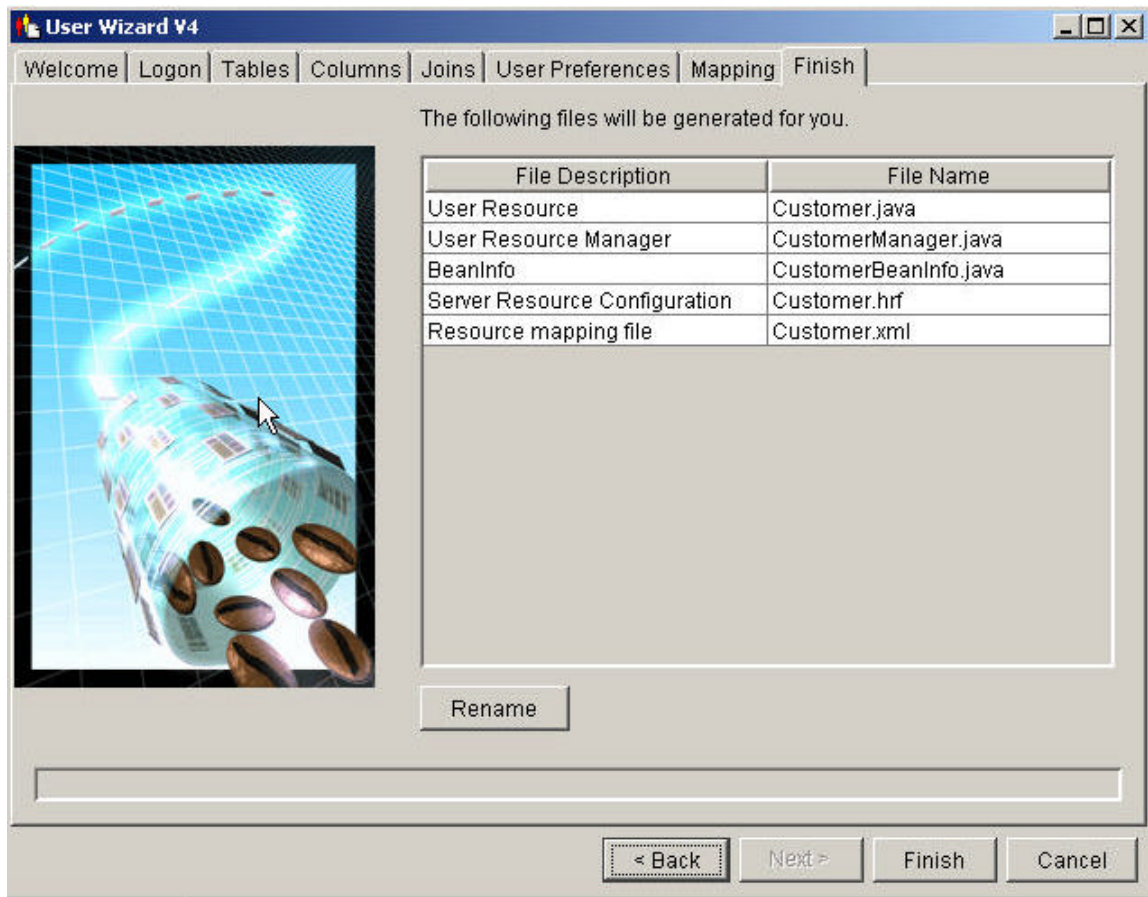
Connect Disconnect

< Back Next > Finish Cancel

- Select the **Tables** page
- Select the **Customer** table and then click on the > button to choose the table to build the resource from
- Press the **Next** button
- Click the >> button to select all of the fields in the table
- Keep clicking the **Next** button to look at the other pages in the dialog. You don't need to change anything on any of the pages.

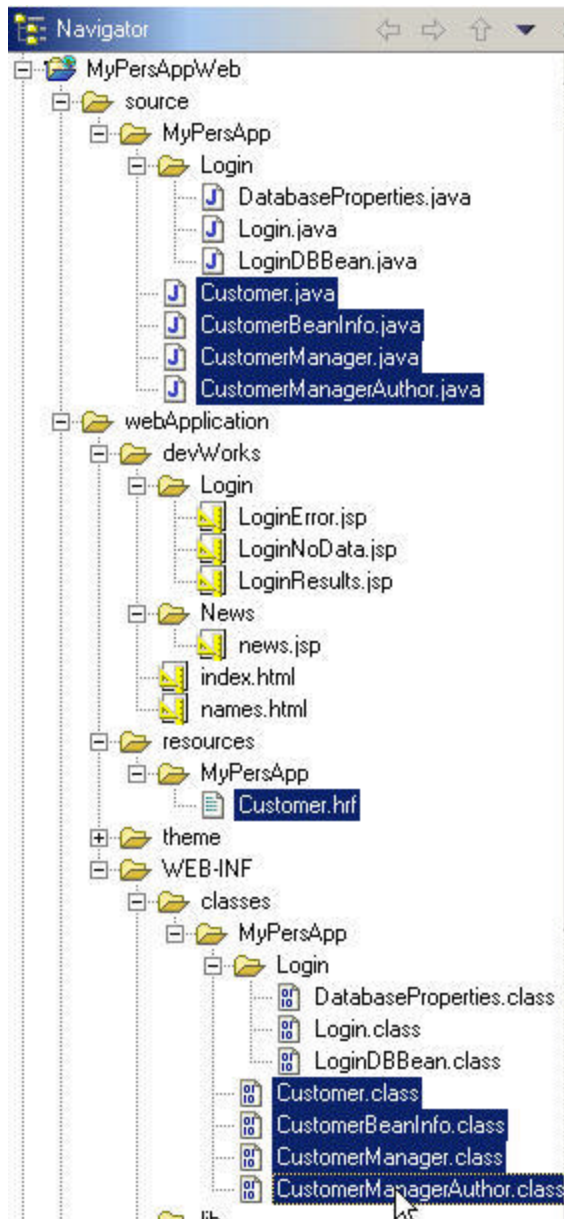


- Notice the files that will be created by the wizard
- Click the Finish button to complete the wizard and generate the files

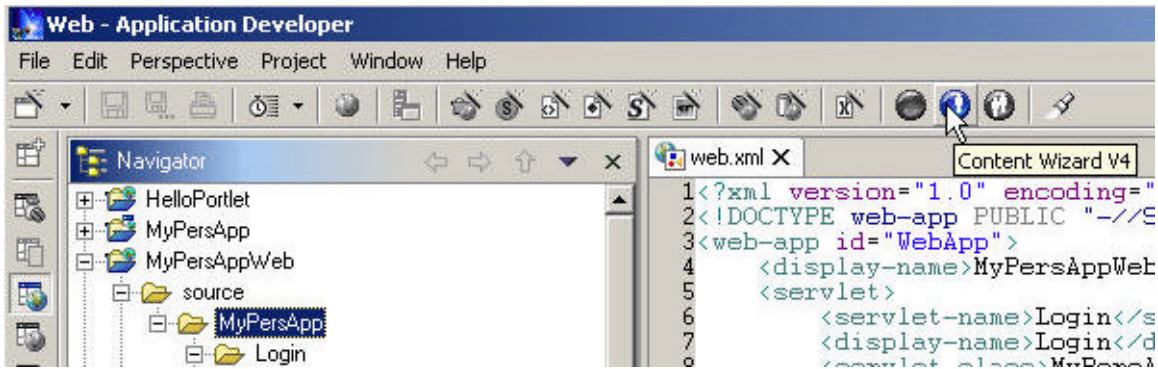


- The following picture highlights the files that the wizard generates for you. Notice that the java files are compiled and placed in the appropriate place in the project for you.



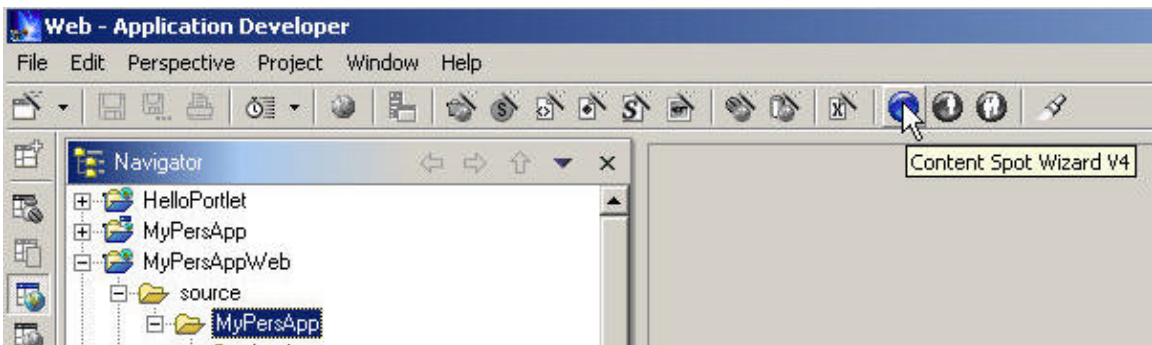


- Follow the same steps to create a content resource. The only differences are that you should:
  - Click on the **Content Wizard** button to invoke the wizard this time
  - Select the **Customernews** table

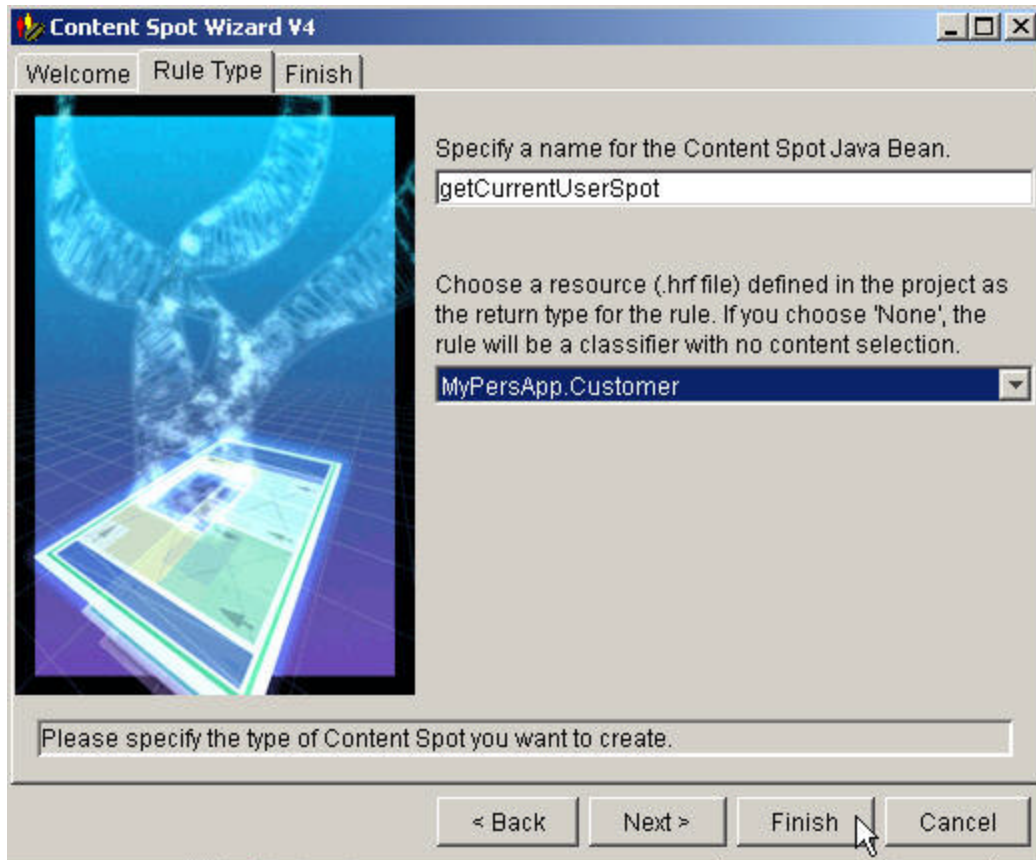


## ***Define Content Spots in the MyPersAppWeb WSAD Project***

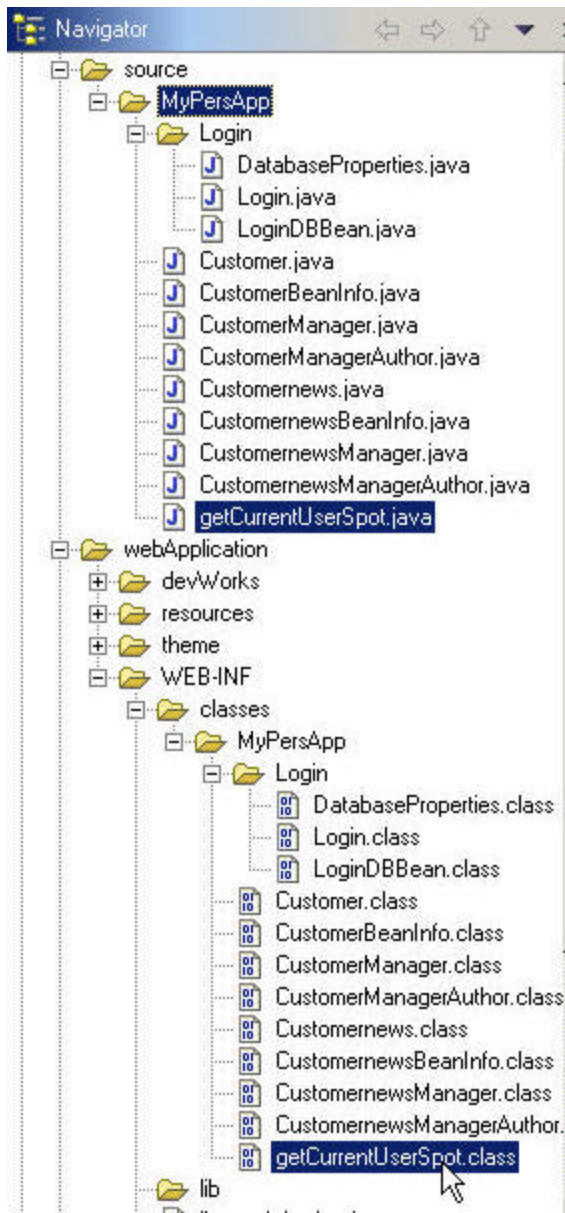
- Select the **MyPersApp** folder in the MyPersAppWeb project
- Click on the **Content Spot Wizard V4** button as shown in the following picture to invoke the Content Spot Wizard



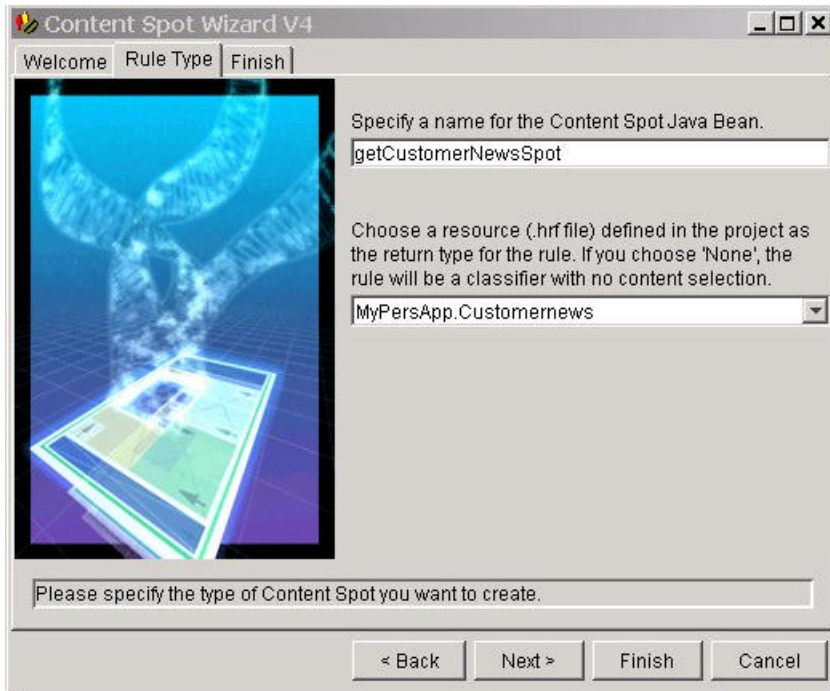
- On the Rule Type page name the Content Spot **getCurrentUserSpot**
- Select **MyPersApp.Customer** from the list-box. This identifies the return type required for the rule you will associate with this content spot.
- Click the **Finish** button to generate the content spot files



- The following picture shows the 2 files that are generated for that content spot



- Follow the same steps to create a second content spot named **getCustomerNewsSpot**. Select **MyPersApp.Customernews** for the resource.



## ***Edit news.jsp to Add the Content Spots and Define the Layout***

- Double click on news.jsp to edit it
- Select the **Design** tab
- Erase the existing line of text
- Drag and drop getCurrentUserSpot.class at the top of file
- Click **OK** to accept the default values for the bean
- Drag and drop getCustomerNewsSpot.class at the top of the file
- Click **OK** to accept the default values for the bean
- Select the **Source** tab
- Use the following picture to update the contents of the file
  - Change the title to **Personalized News**
  - Change the useBean declaration as follows: (Notice the / is removed before the > in the id lines.)

The original declaration in your jsp looks like:

```
<jsp:useBean class="MyPersApp.getCurrentUserSpot"
id="getCurrentUserSpot"/>
<jsp:useBean class="MyPersApp.getCustomerNewsSpot"
id="getCustomerNewsSpot"/>
```

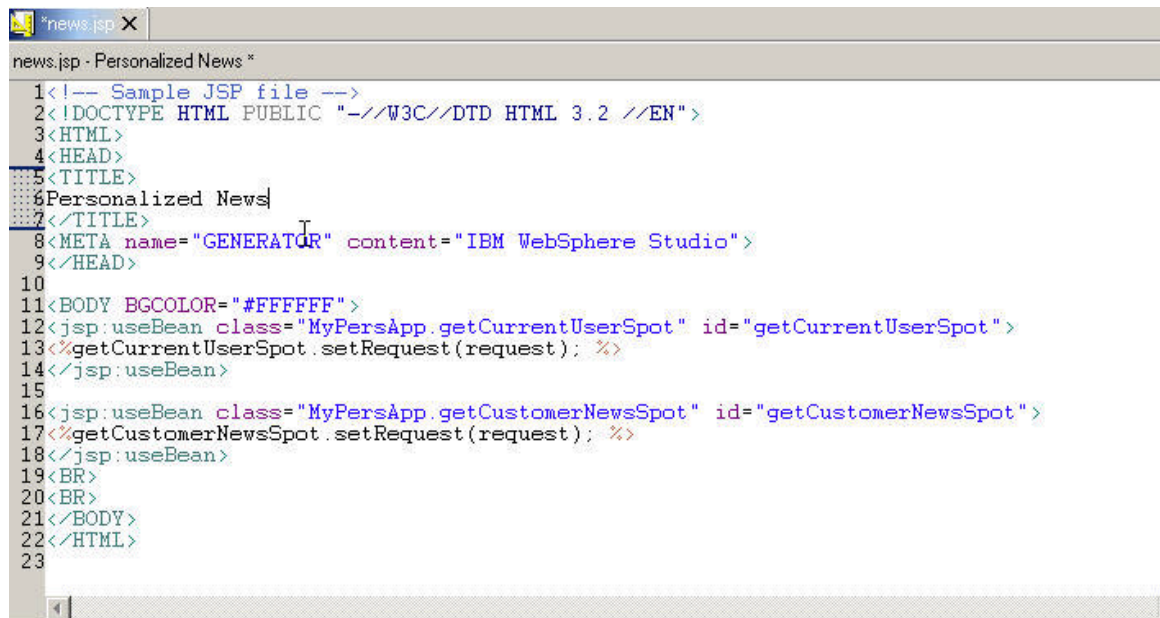
Change the above lines to look like this:

```

<jsp:useBean class="MyPersApp.getCurrentUserSpot"
id="getCurrentUserSpot">
<% getCurrentUserSpot.setRequest(request); %>
</jsp:useBean>
<jsp:useBean class="MyPersApp.getCustomerNewsSpot"
id="getCustomerNewsSpot">
<% getCustomerNewsSpot.setRequest(request); %>
</jsp:useBean>

```

These changes are necessary for the Personalization runtime, and must be added for every content spot you use in a JSP.



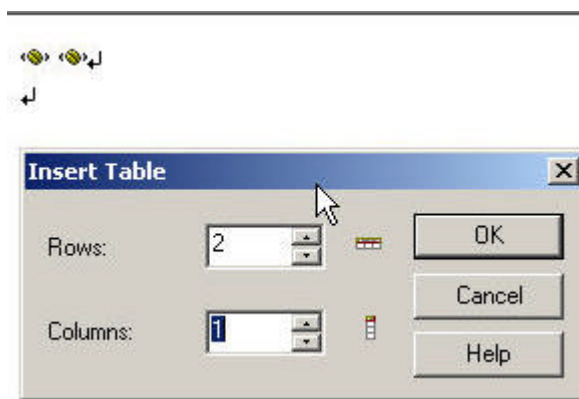
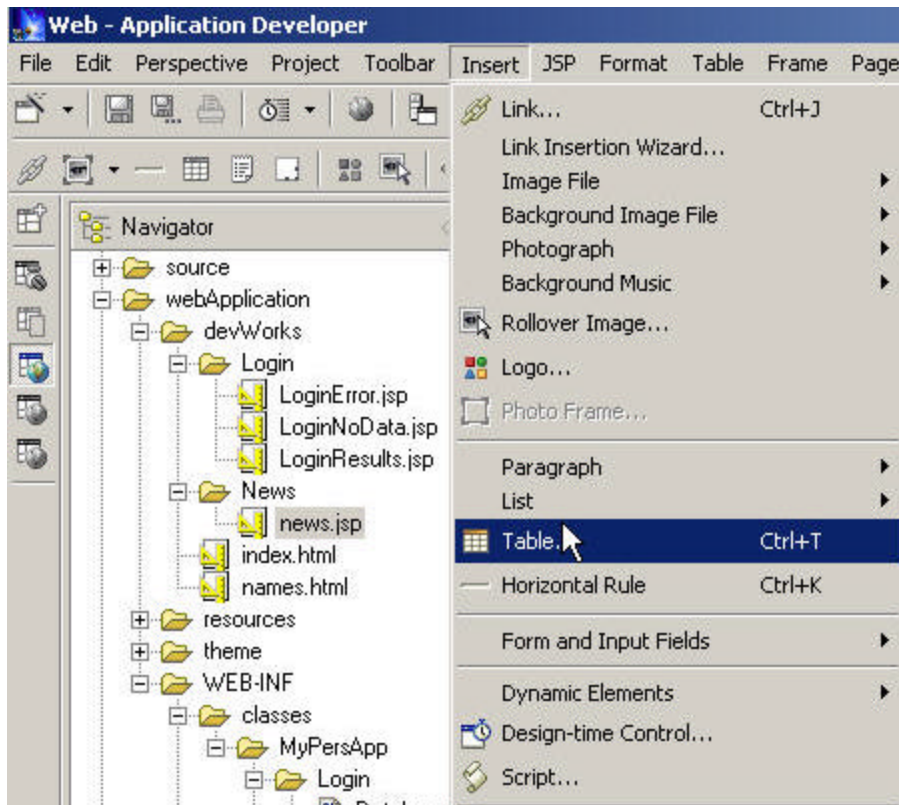
```

news.jsp - Personalized News *
1<!-- Sample JSP file -->
2<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 //EN">
3<HTML>
4<HEAD>
5<TITLE>
6Personalized News
7</TITLE>
8<META name="GENERATOR" content="IBM WebSphere Studio">
9</HEAD>
10
11<BODY BGCOLOR="#FFFFFF">
12<jsp:useBean class="MyPersApp.getCurrentUserSpot" id="getCurrentUserSpot">
13<%getCurrentUserSpot.setRequest(request); %>
14</jsp:useBean>
15
16<jsp:useBean class="MyPersApp.getCustomerNewsSpot" id="getCustomerNewsSpot">
17<%getCustomerNewsSpot.setRequest(request); %>
18</jsp:useBean>
19<BR>
20<BR>
21</BODY>
22</HTML>
23

```

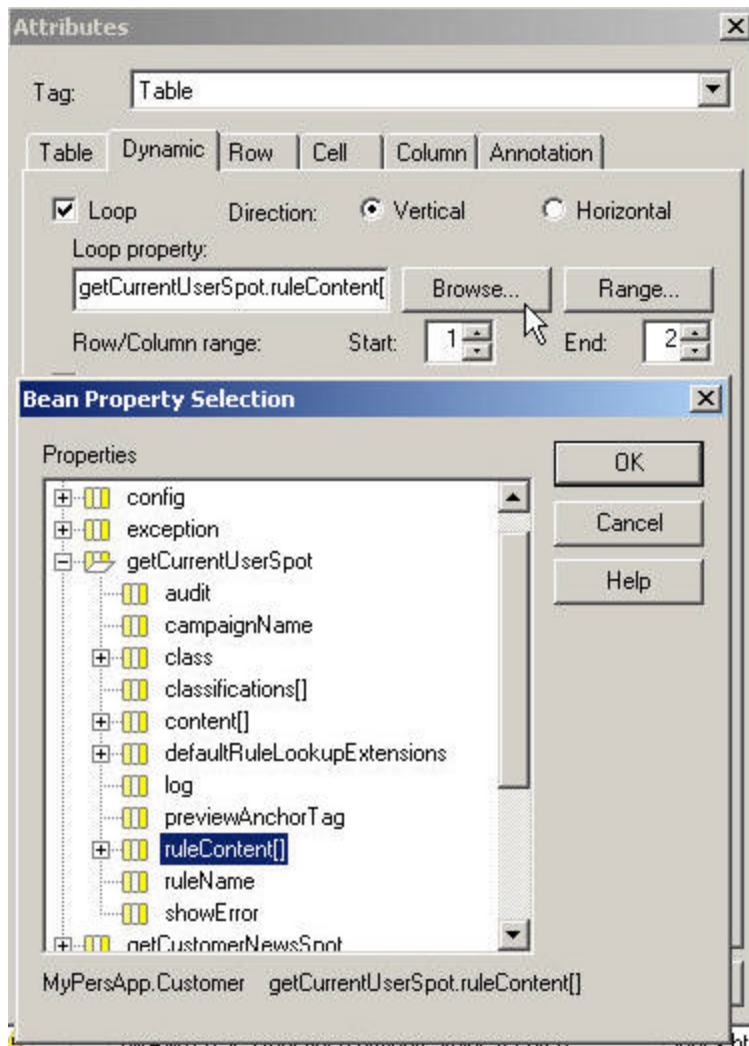
- Select the **Design** tab
- Select **Insert->Table** to create a new table
- Specify 2 rows and 1 column
- Press the **OK** button





- Double click on the table to bring up its Attributes dialog
- Select the **Dynamic** Tab
- Check the **Loop** check-box
- Specify 1 for the **Start** row and 2 for the **End** row
- Click on the **Browse...** button to bring up the Bean Property Selection dialog
- Select **getCurrentUserSpot.ruleContent[]** and press the **OK** button
- Press the **OK** button on the Attributes dialog to dismiss it

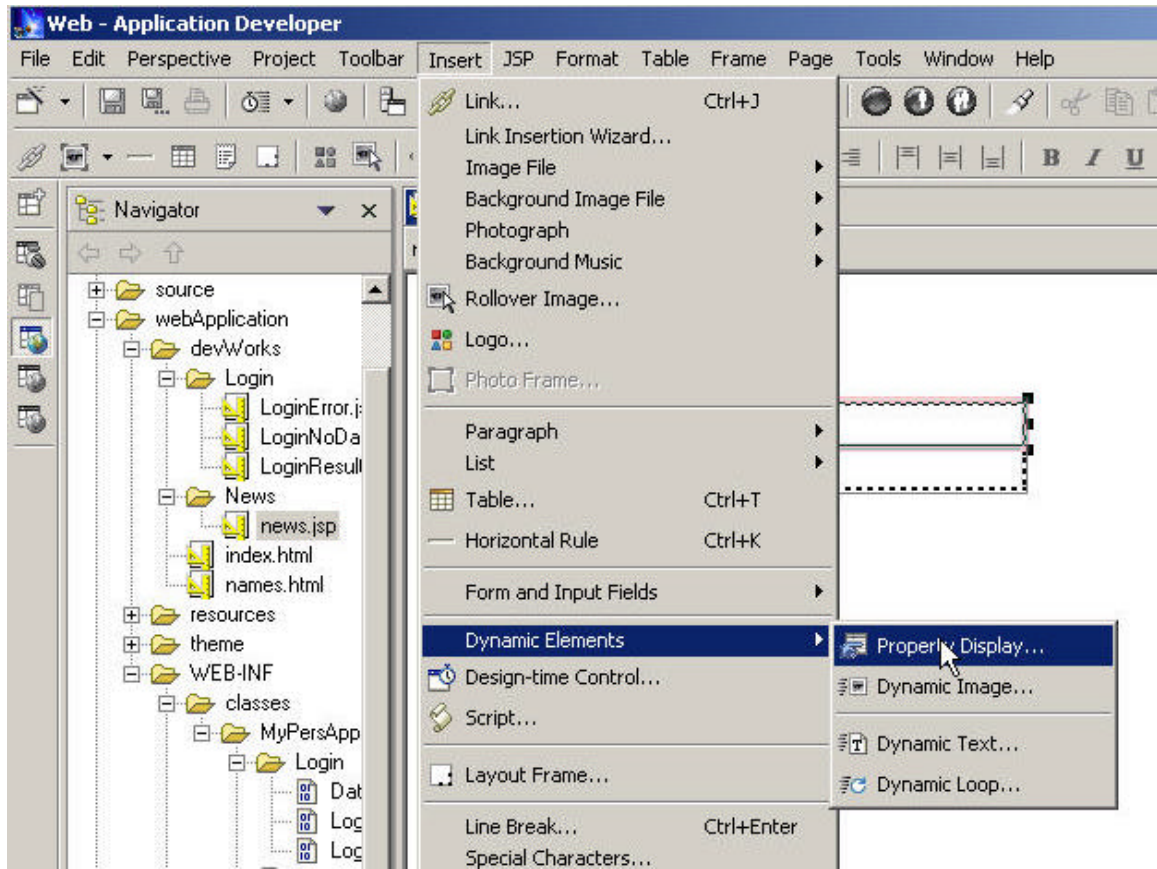
Essentially, this step specifies which resource we are going to use to populate the rows of the table with data.



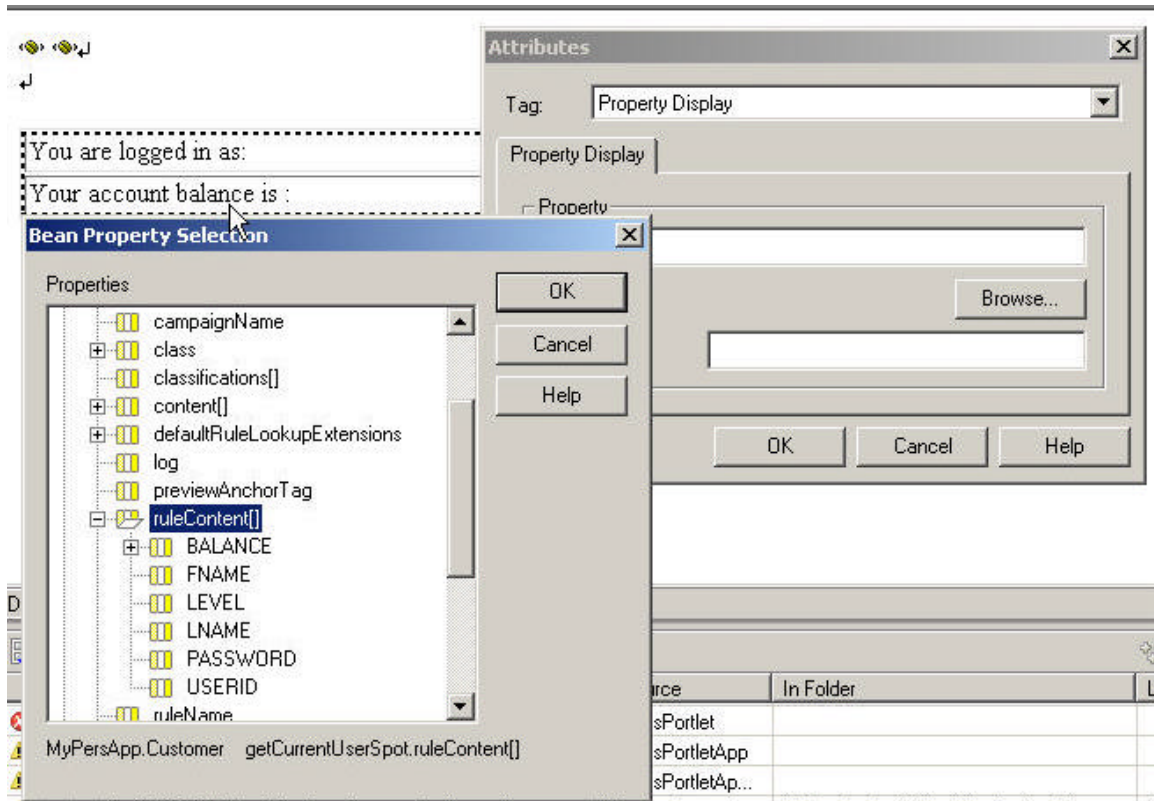
Now we will specify which attributes of that resource we will use to populate the rows of the table.

- Type **You are logged in as:** in the first row of the table
- Click inside the first row to the right of the text and then select **Insert->Dynamic Elements->Property Display...** to bring up the Attributes dialog





- In the Attributes dialog click on the **Browse** Button
- Notice that in the Bean Property Selection dialog that **getCurrentUserSpot.ruleContent[]** is already highlighted
- Select the **USERID** attribute and press the **OK** button
- Press the **OK** button to dismiss the Attributes dialog

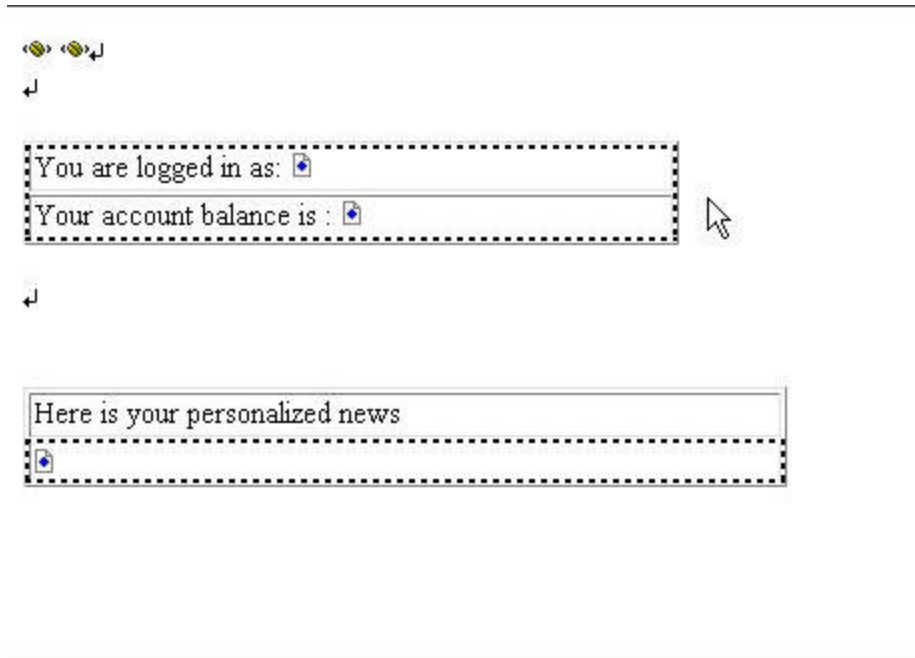


- In the second row of the table type **Your account balance is:**
- Follow the same steps as above to specify the contents of the second row. The only change is you should select the **BALANCE** attribute this time.

Complete the news.jsp page by creating a second table.

- Click outside the first table and press Enter a few times to create some white space in the page
- Insert a second table in the page. Use the following picture to complete this second table
  - In this table the **Start** and **End** row for the loop should both be 2
  - Select **getCustomerNewsSpot.ruleContent[]** for the **Loop property:** of the table
  - In the second row you will select the **Content** attribute

Make sure that you save news.jsp after verifying that it looks like the following picture. You can do this by pressing **<Ctrl><S>** or by selecting **File->Save news.jsp**



### \_\_\_\_ ***Export the Artifacts from the MyPersAppWeb Web Application to the File System***

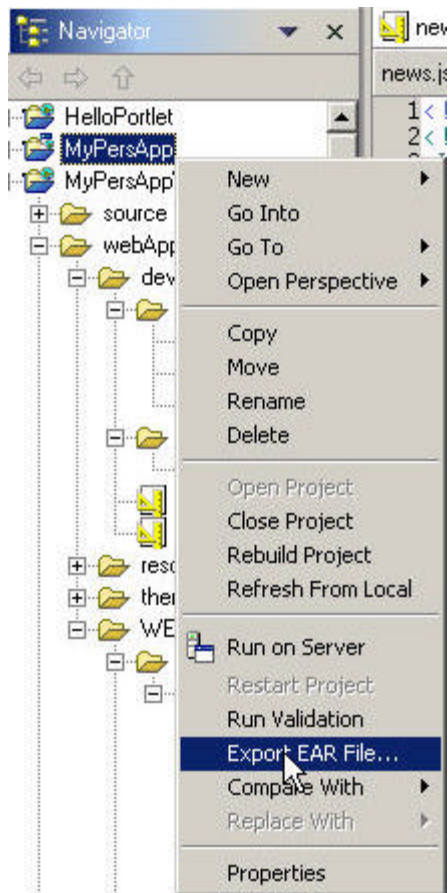
We need to export the files contained in the MyPersAppWeb project to the file system. We will need these files later when we create a project for the portlet.

- Highlight the **MyPersAppWeb** web application
- Select **File->Export->File System**
- Specify **C:\PersPortlet** for the location to export the files to and press the **FINISH** button

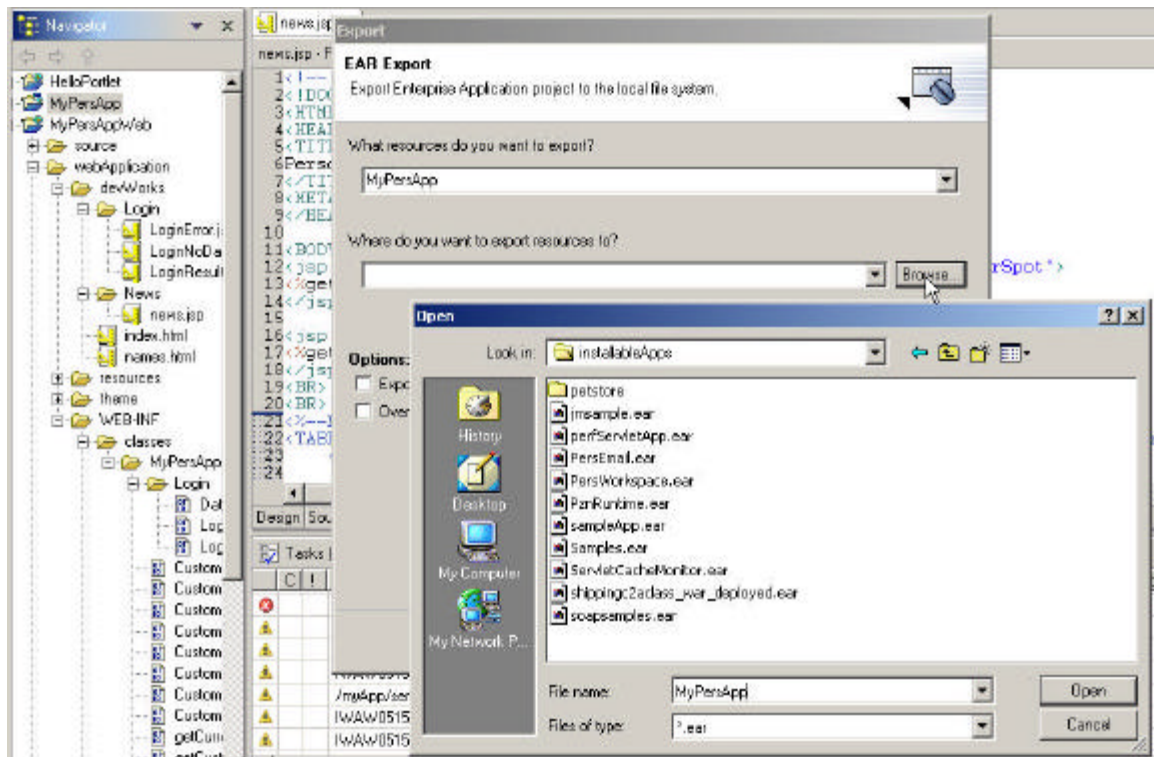
### \_\_\_\_ ***Export the MyPersApp.ear File***

In this step we will package up our application as an EAR so we can install it in WAS.

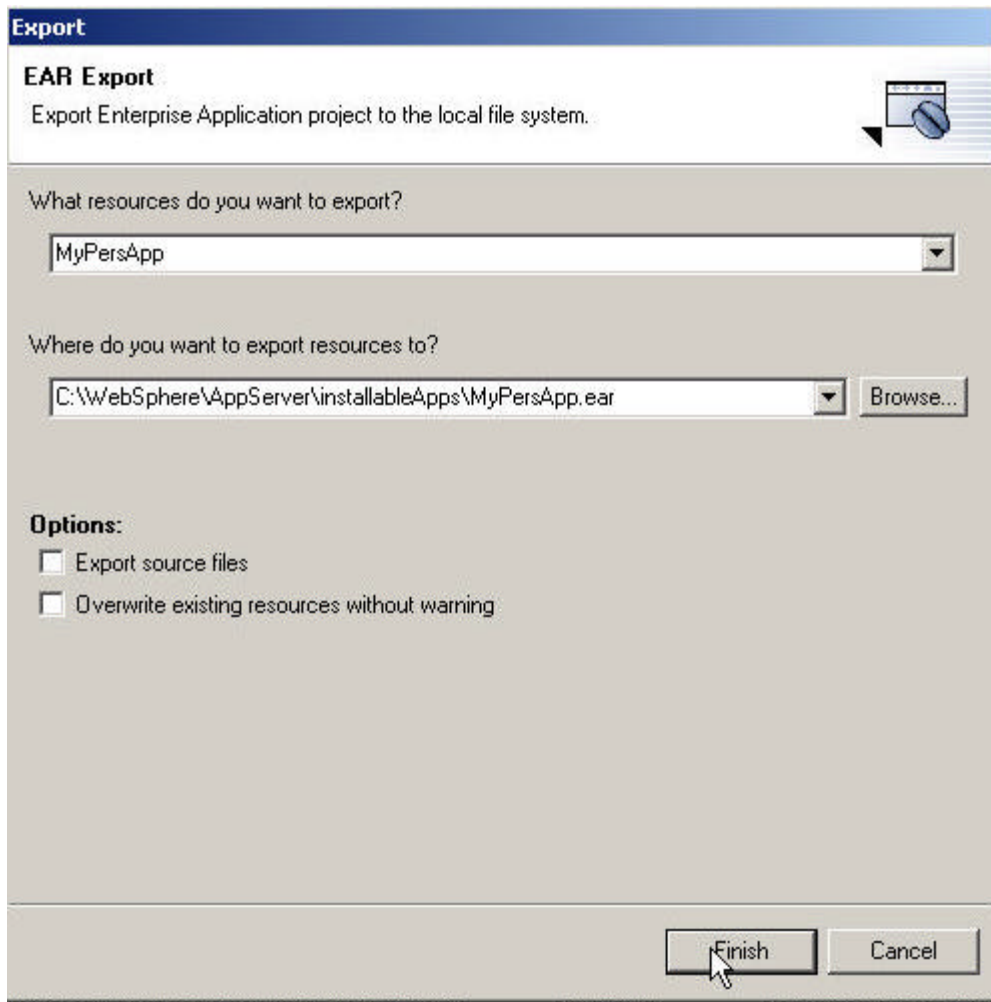
- Select **MyPersApp->Export EAR File...**



- Press the **Browse...** button
- Navigate to the **C:\WebSphere\AppServer\installableApps** directory
- Specify **MyPersApp** for the file name and press the **Open** button



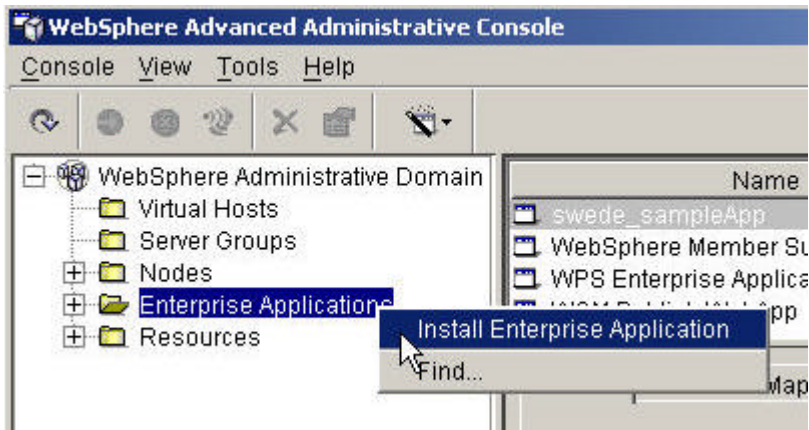
- Press the **Finish** button



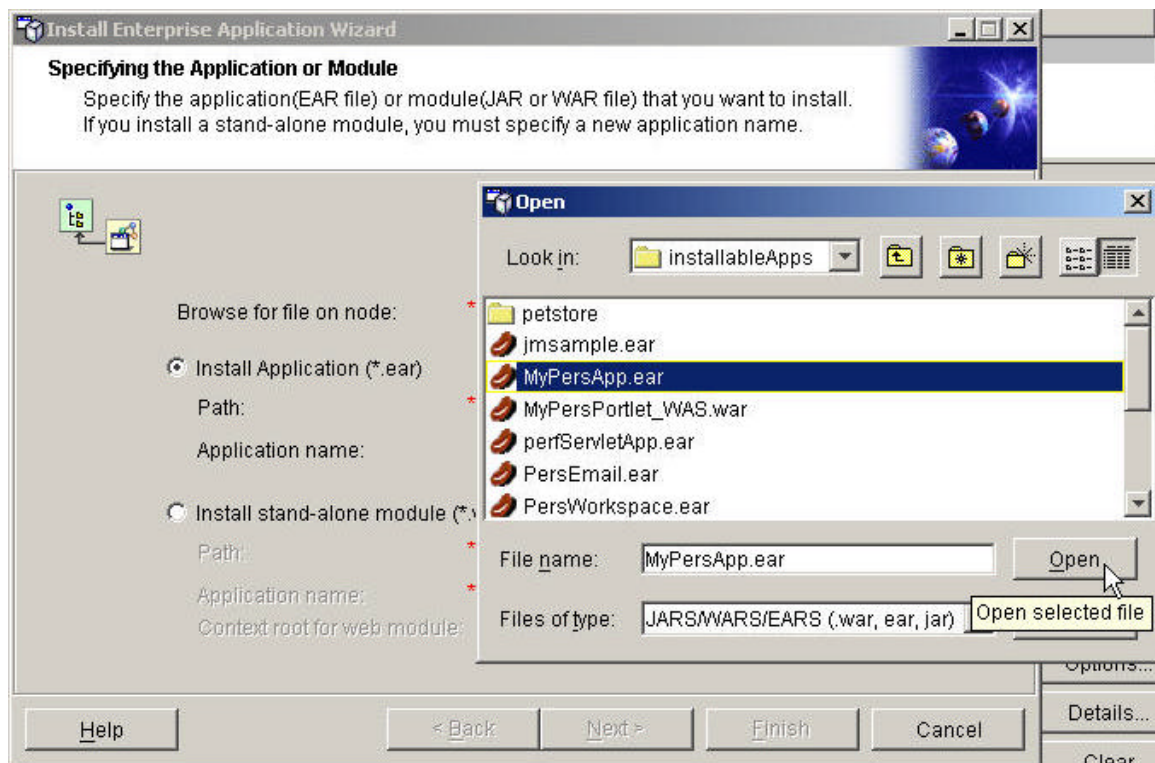
### ***Install the MyPersApp.ear File***

- Double-click the **Services** icon on the desktop
- Select **IBM WS AdminServer 4.0**, right-click, and then click on **Start**
- Double-click the **Websphere Console** icon on the desktop
- When the Console has finished initializing select **Enterprise Applications**, right-click, and select **Install Enterprise Application**

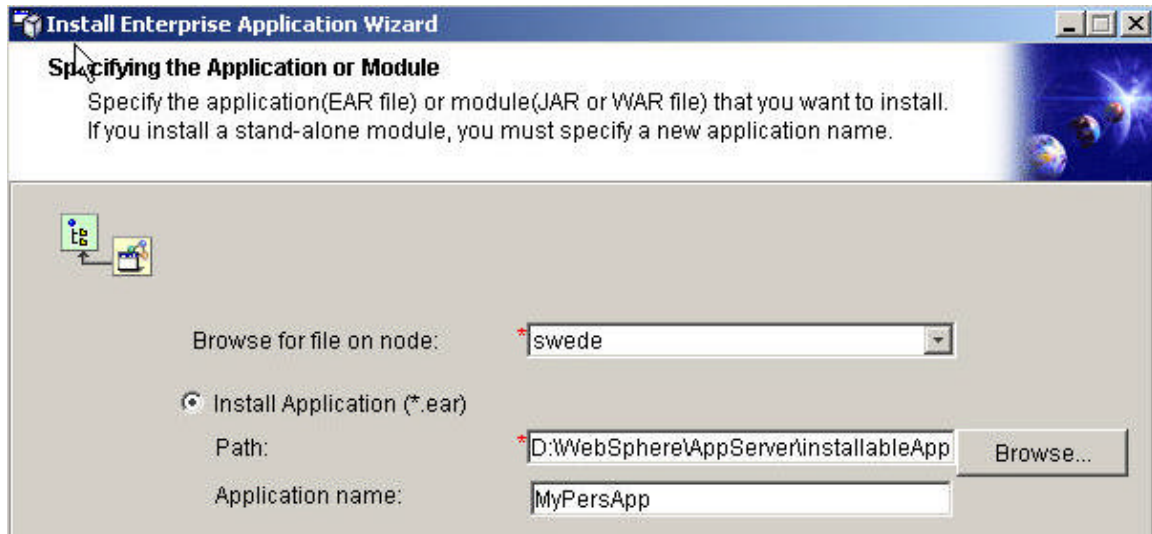




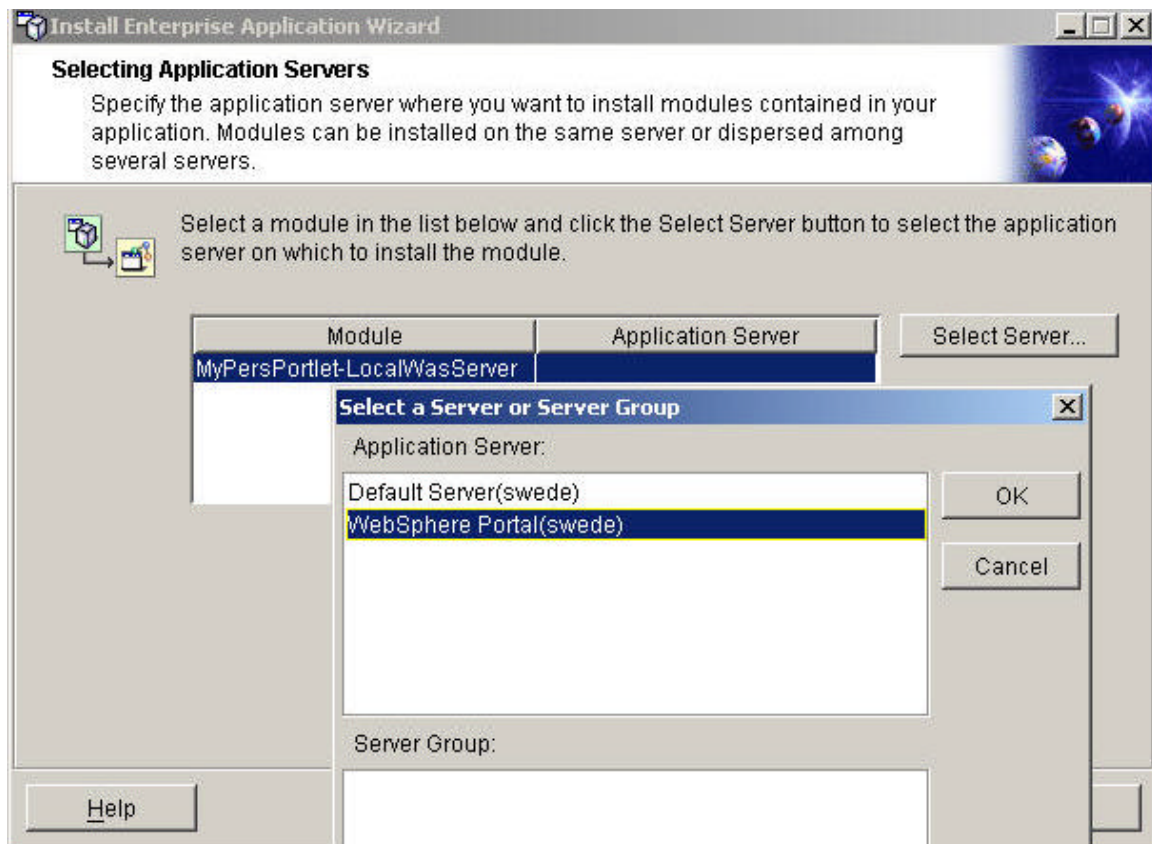
- Click on the **Install Application (\*.ear)** radio button
- Press the **Browse...** button, and navigate to the installableApps directory
- Select **MyPersApp.ear** and press the **Open** button.



- Specify **MyPersApp** for the Application name



- Keep pressing the **Next** button, accepting the defaults until you come to the following screen
- Press the **Select Server...** button
- Select the **WebSphere Portal** application server and press the OK button
- Continue pressing the **Next** button until you are able to press the **Finish** button



- Stop and Start the Websphere Portal application server

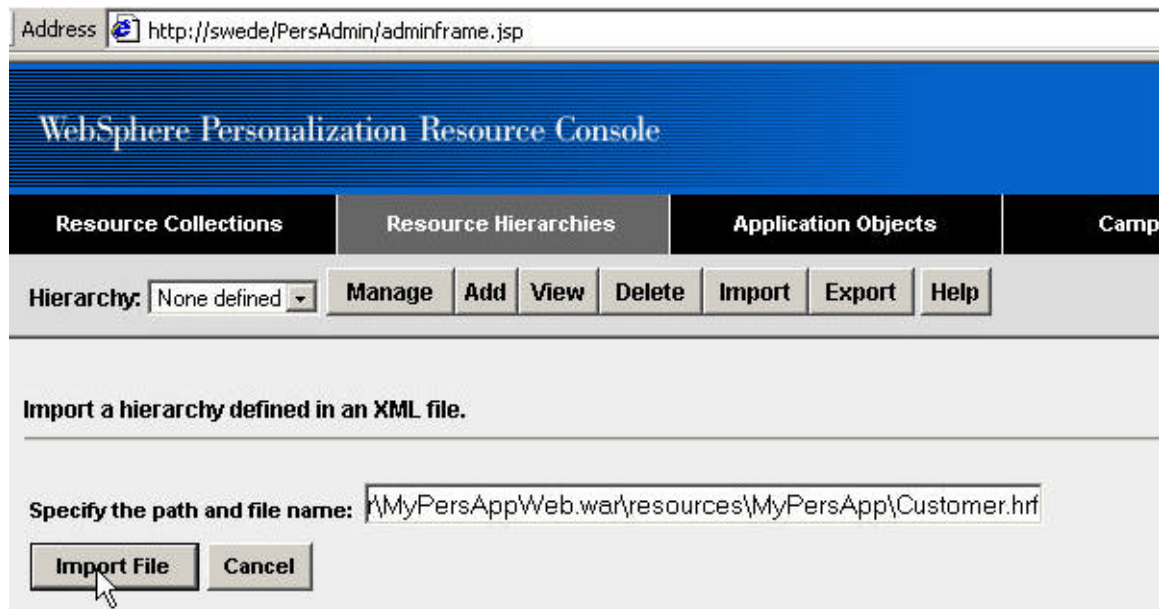


- Highlight the **WebSphere Portal** Application Server in the WebSphere Advanced Administrative Console
- Right-click and select **Stop**
- After you receive a message indicating the application server has successfully stopped, right-click and select **Start**
- Highlight the node in the WebSphere Administration Console, right-click and select **Regen Webserver Plugin**
- Use the Services dialog to stop and start the HTTP Server

## ***\_\_\_ Import the Customer and Customernews Resources into the Personalization Runtime***

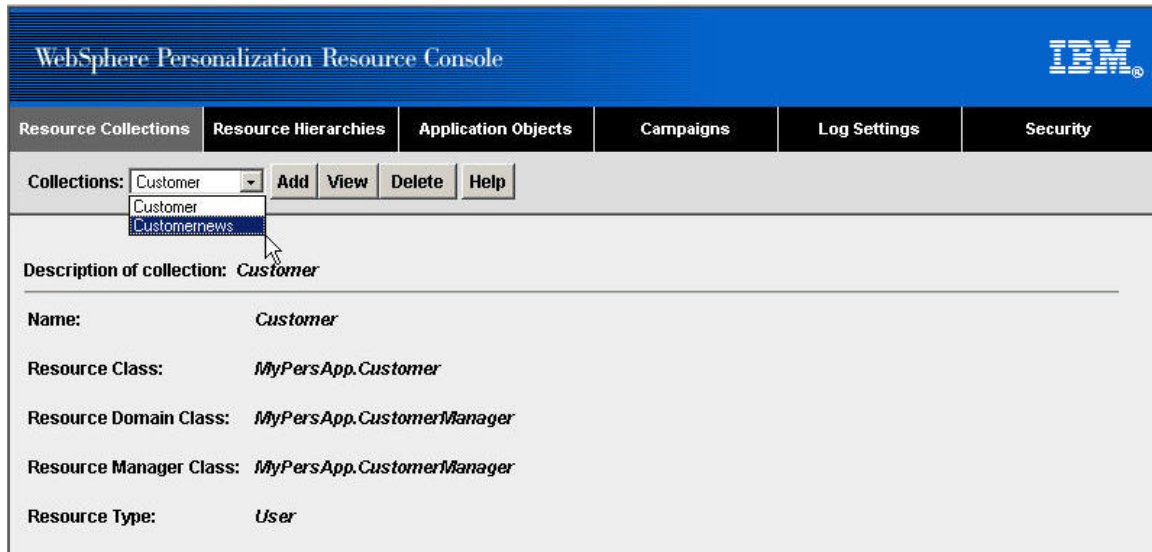
This step is needed so that the Personalization runtime recognizes Customer and CustomerNews as Personalization Resources. The WebSphere Personalization Resource Console is implemented as a web application.

- **Double-click on the Internet Explorer icon** to start Internet Explorer
- We have added a folder of links for this lab under Internet Explorer's Favorites. Use the appropriate link to start the Resource console
- Select the Resources Hierarchies tab
- Press the **Import** button. The Customer.hrf and Customernews.hrf files are in **C:\WebSphere\AppServer\installedApps\MyPersApp.ear\MyPersAppWeb.war\resources\MyPersApp**.
- Notice in the following picture that you need to specify the name of the file in addition to the path
- Press the **Import File** button for each of the resources



## \_\_\_\_ **Verify that the Customer and Customernews Resources Have Been Defined**

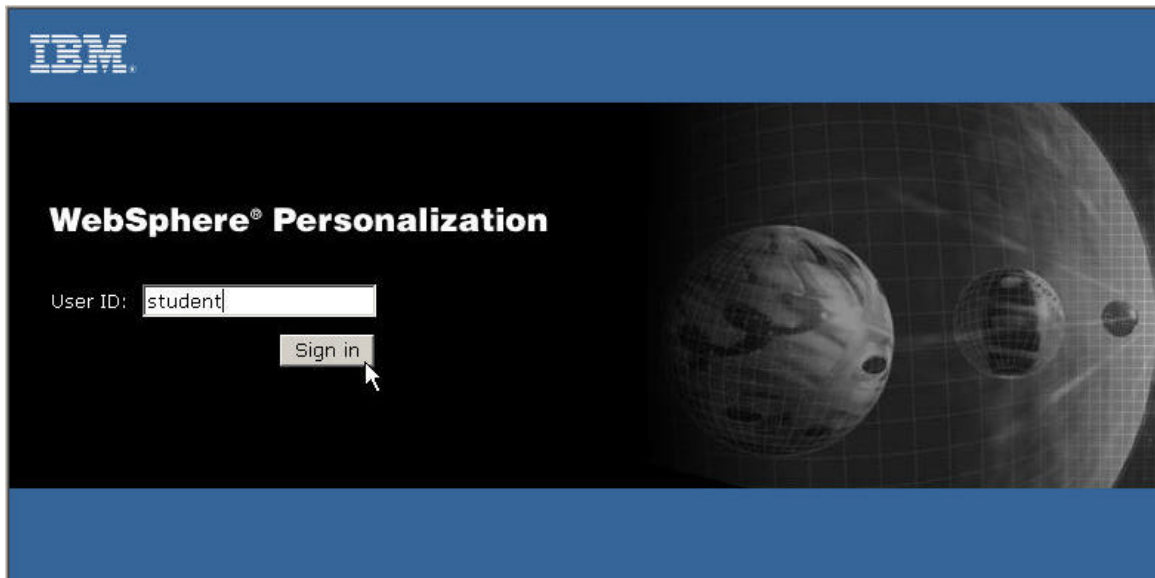
- Press the **Resource Collections** tab
- You should see the Customer and CustomerNews resources as shown below



## \_\_\_\_ **Start the Personalization Workspace and Add a Publishing Server**

The WebSphere Personalization Workspace is implemented as a web application.

- **Double-click on the Internet Explorer icon** to start Internet Explorer
- Use the appropriate link to start the Personalization workspace
- Log in as **student** as shown below

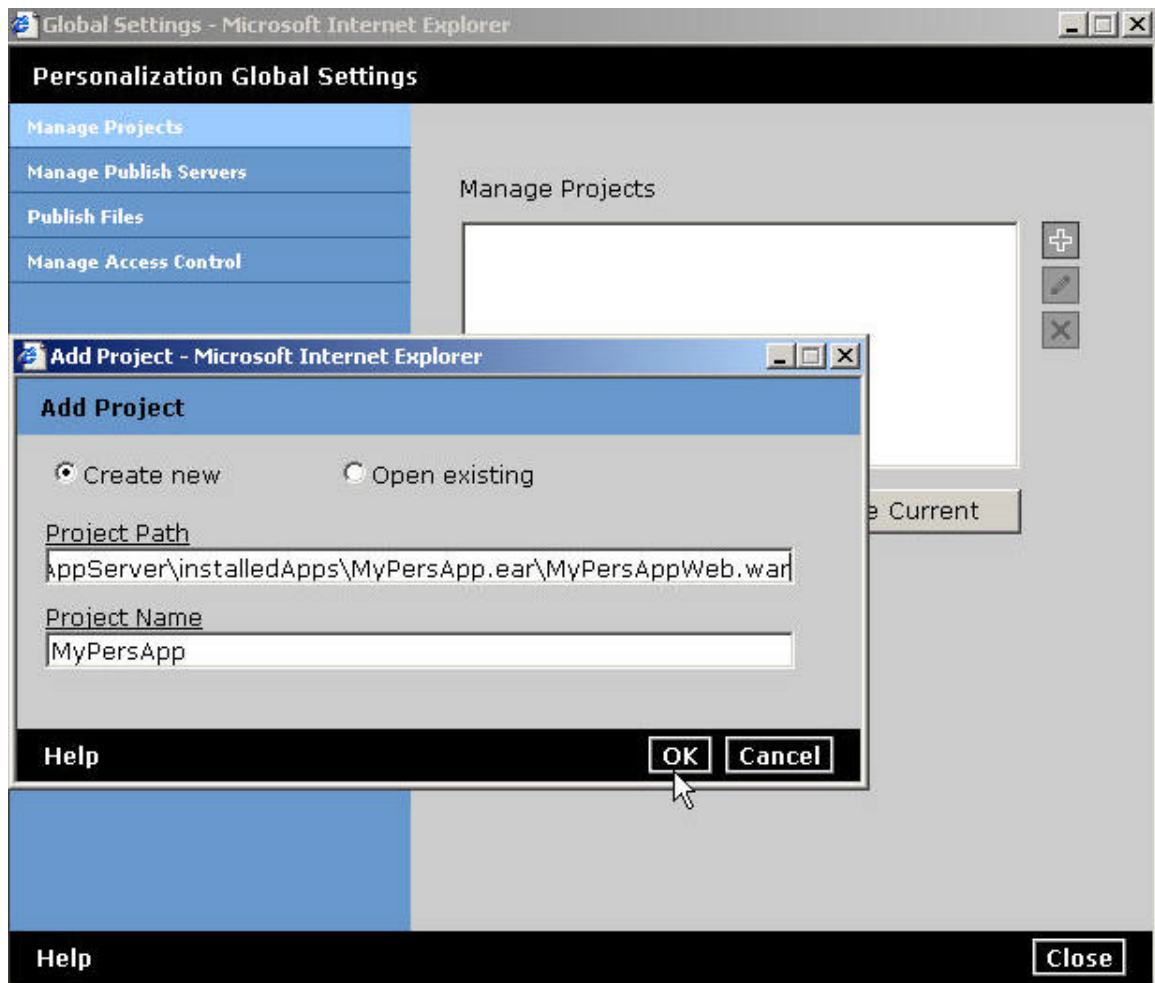


- Click **Global Settings** on the right side of the workspace
- In the Global Settings window click **Manage Publish Servers**
- Click the **plus sign** at the right to add a new publish server
- Use the following picture to fill out the fields
- Click **Save**
- Close the Global Settings Window.

The image shows a "Add Publish Server" dialog box within a Microsoft Internet Explorer window. The dialog box has a blue header bar with the title "Add Publish Server". Below the header, there are several input fields: "Name" with the value "devWorksServer", "URL" with the value "http://localhost", a checkbox labeled "Requires authentication" which is unchecked, "User Name" with an empty field, and "Password" with an empty field. At the bottom of the dialog box, there are three buttons: "Help", "Save", and "Cancel". A mouse cursor is pointing at the "Save" button.

## ***Create the MyPersApp Project***

- Click **Global Settings** on the right side of the workspace
- Select **Manage Projects**
- Click the **plus sign** on the right side of the Global Settings window to add a new project to the workspace as shown below
- Set the Project Name to: **MyPersApp**
- Set the Project Path to:  
**C:\WebSphere\AppServer\installedApps\MyPersApp.ear\MyPersAppWeb.war**



## ***Edit the MyPersApp Project and Make it the Current Project***

- Use the following picture to fill in the fields

**Important Notes:** This picture was taken on a system where WebSphere was installed on the D drive. Make sure that your paths reflect the fact that WebSphere is installed on the C drive on the lab machines.

The 2 paths that aren't completely shown in the picture are as follows:

Resource Path:

**C:\WebSphere\AppServer\installedApps\MyPersApp.ear\MyPersAppWeb.war\resources\MyPersApp**

Resource and Content Spot Class Path:

**C:\WebSphere\AppServer\installedApps\MyPersApp.ear\MyPersAppWeb.war\WEB-INF/classes**

Edit Project: MyPersApp - Microsoft Internet Explorer

**Edit Project**

Project Path  
D:\WebSphere\AppServer\installedApps\MyPersApp.ear\MyPersAppWeb.war

Project Name  
MyPersApp

Campaign Path  
D:\WebSphere\AppServer\installedApps\MyPersApp.ear\MyPersAppWeb.war

Rule Paths  
D:\WebSphere\AppServer\installedApps\MyPersApp.ear\MyPersAppWeb.war

Resource Paths  
D:\WebSphere\AppServer\installedApps\MyPersApp.ear\MyPersAppWeb.war\resource:

Resource and Content Spot Class Path  
D:/WebSphere/AppServer/installedApps/MyPersApp.ear/MyPersAppWeb.war/WEB-INF/

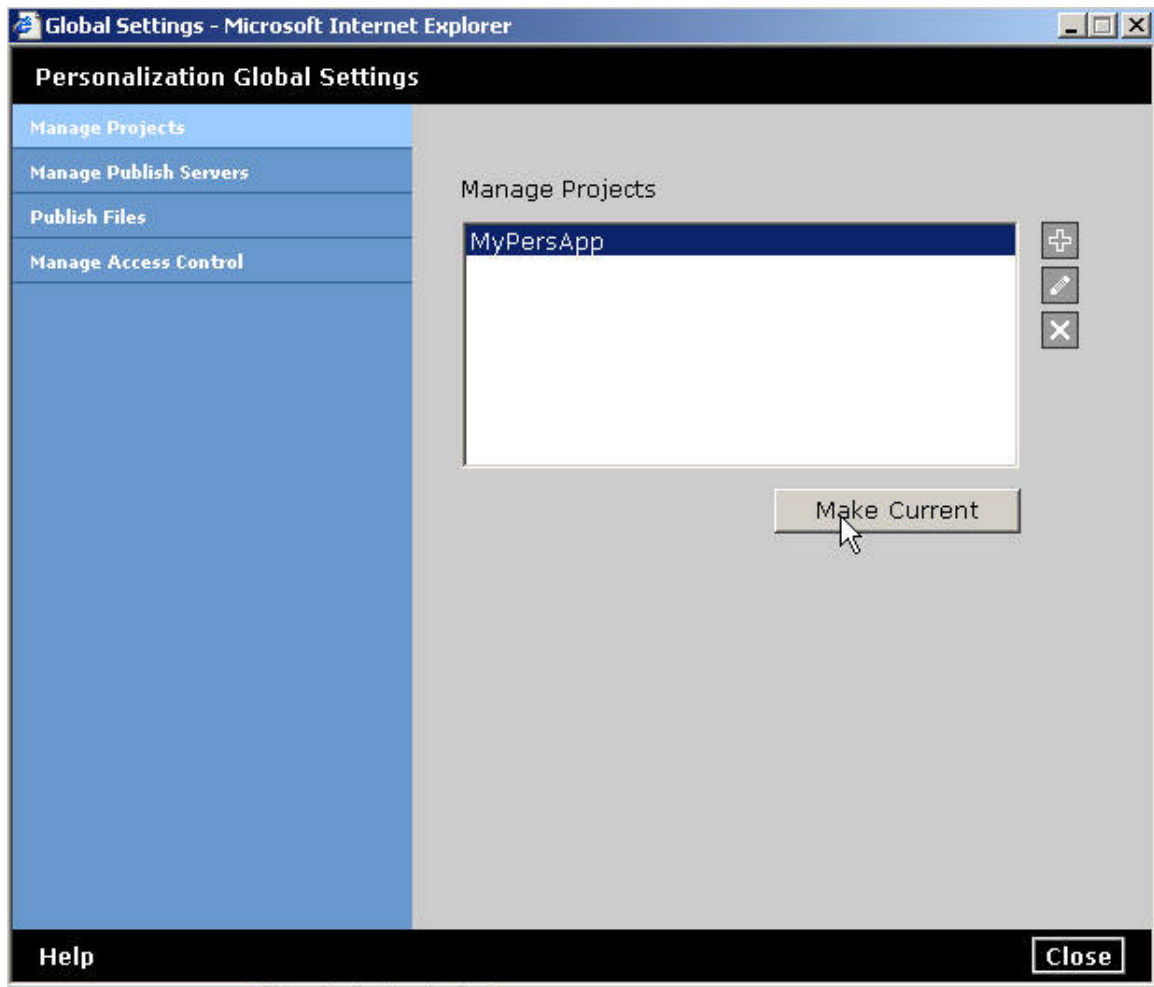
Preview User Profile Class (ex. PersSample.Personnel)  
MyPersApp.Customer

Preview Start Page URI (ex. news/news.jsp)  
myApp/devWorks/News/news.jsp

Preview Server  
devWorksServer

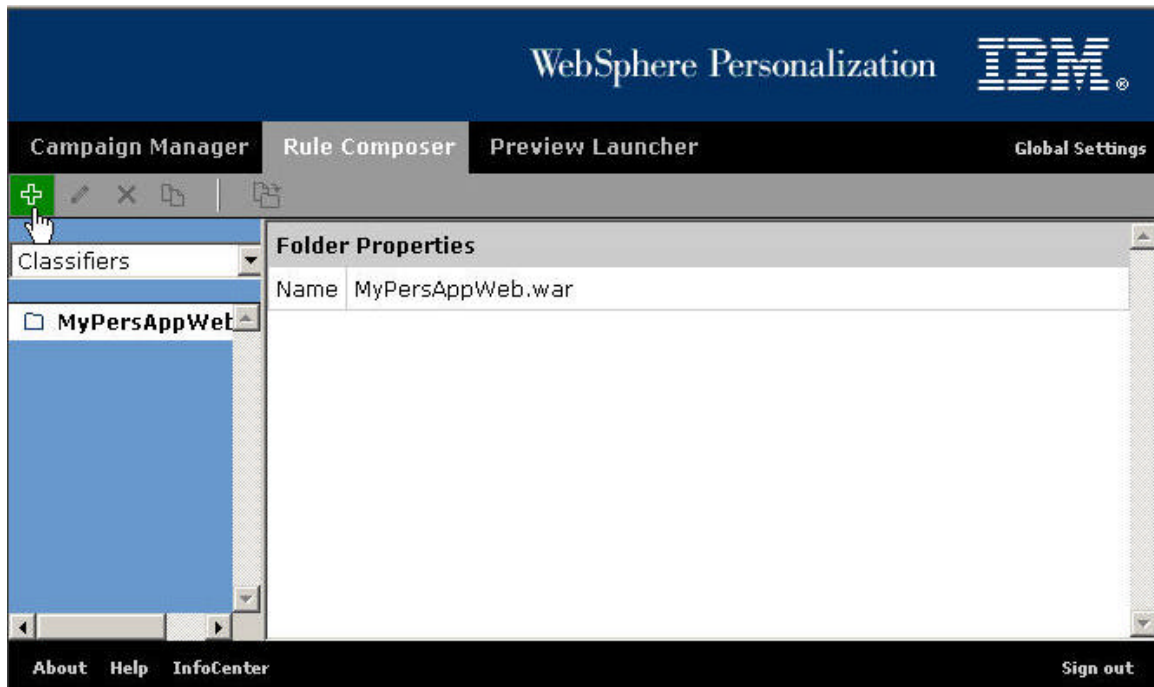
**Help** **Save** **Cancel**

- Click **Save** to save the project
- Click the **Make Current** button to make MyPersApp the current project



### --- ***Create the UserLevel Classifier Rule***

- Select the **Rule Composer** tab in the workspace
- Select the **Classifiers** rule type in the list-box
- Click the **plus sign** at the left to create a new classifier



- Name the classifier **UserLevel**
- Click **Classification** in the rule body and name this classification **Basic**
- Click on **Resource.Attribute** and select the radio buttons as shown in the following picture



Add Classifier - Microsoft Internet Explorer

Name of the Classifier:

Comment about the Classifier:

Classifier Description (click on an underlined value to edit it):

UserLevel is  
☒ Basic when  
Resource.Attribute is equal to value

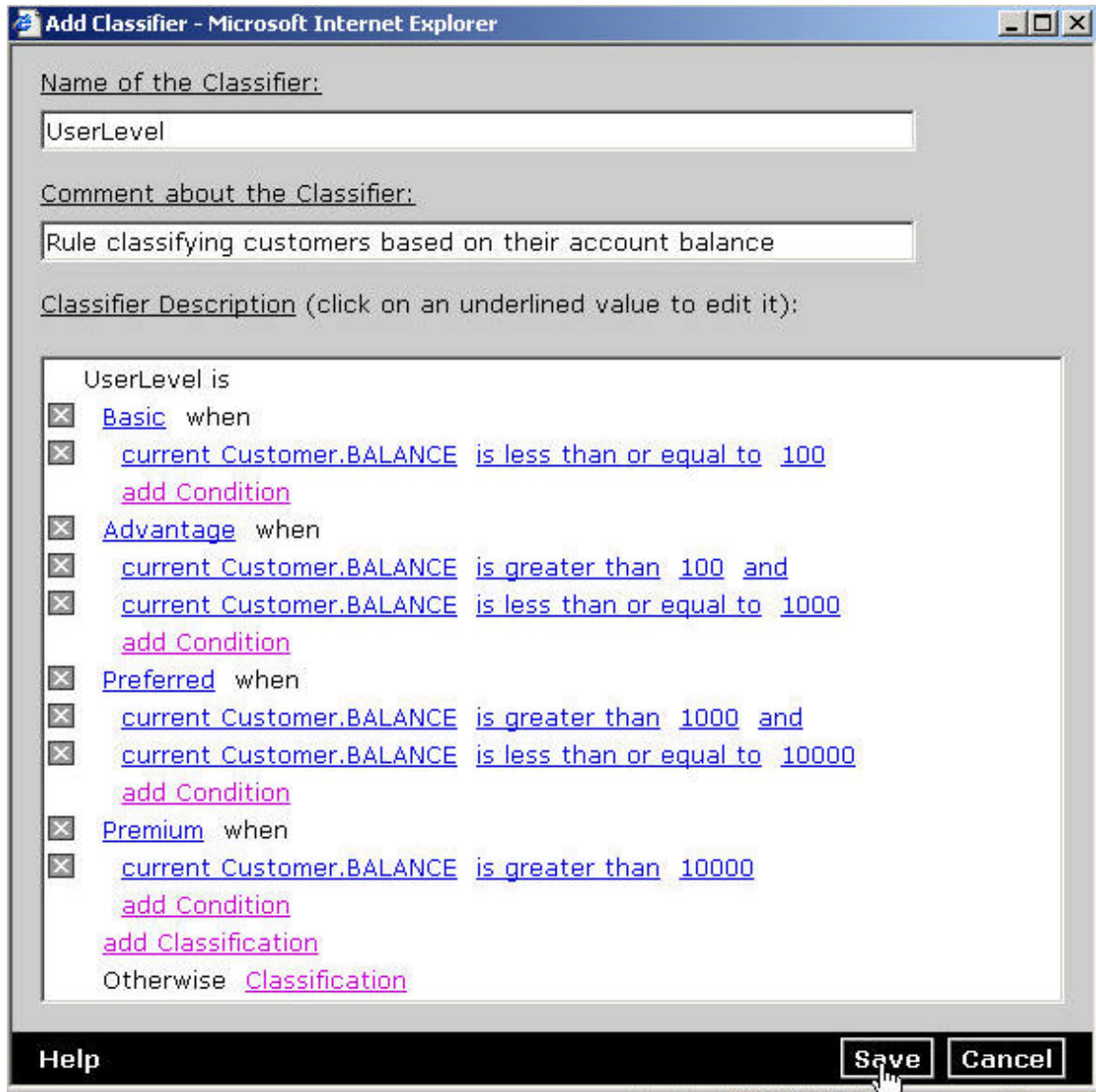
Select a Resource and its Attribute - Microsoft Internet Explorer

Resource	Collection
<input type="radio"/> current Browser Capability	
<input type="radio"/> current Category Count	
<input checked="" type="radio"/> current Customer	Customer
<input type="radio"/> current Date	
<input type="radio"/> current Page Request	
<input type="radio"/> current User Session	

Attribute	Type
<input checked="" type="radio"/> BALANCE	Number
<input type="radio"/> FNAME	Text
<input type="radio"/> LEVEL	Text
<input type="radio"/> LNAME	Text
<input type="radio"/> PASSWORD	Text
<input type="radio"/> USERID	Text

Help Save

- Click **is equal to** and change it to **is less than or equal to**
- Click **value** and enter 100
- Continue adding Classifications until the rule looks like the following picture.  
 Use **add Condition** when you need to add more than one condition for a classification



## \_\_\_\_ **Create the *GetCurrentUser* Action Rule**

There are 2 types of Action rules, “*Select content*” rules and “*Update*” rules. “*Select content*” rules are used to specify what content you are going to show to your end users based on the current user, as well as information contained in the Request object or in the user’s Session. “*Update*” rules set information into the user’s Session or User resource based on information from the Request object, the user’s Session, the User resource, or from the current date or time. We are creating rules of the first type.

- Select **Actions** in the **Rule Composer** list-box
- Click on the **plus sign** to add a new action rule named **GetCurrentUser**

Notice in the following picture that you need to click the **Specify another**

**attribute** radio button and then click the **Browse...** button when setting the value field. When you save the value it will change from Customer.USERID to current Customer.USERID.

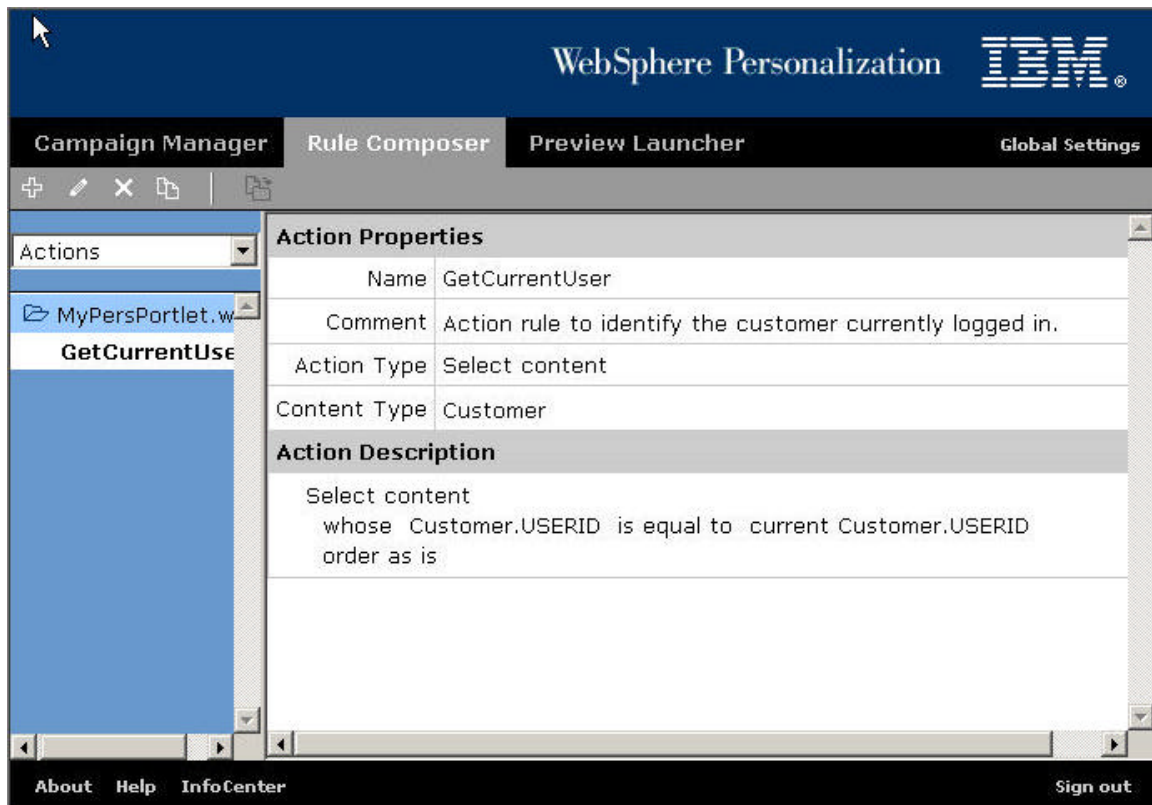
The screenshot shows three overlapping web browser windows from Microsoft Internet Explorer.

- Add Action - Microsoft Internet Explorer:** This window contains fields for "Name of the Action:" (GetCurrentUser), "Comment about the Action:" (Action rule to identify the customer currently logg), "Type of Action:" (Select content), and "Action Description" (whose Customer.USERID is equal to value). A dropdown menu shows "Select content".
- Specify Value - Microsoft Internet Explorer:** This window has two radio buttons: "Specify a value of type Text" and "Specify another attribute". The "Specify another attribute" option is selected. There is a text input field and a "Browse..." button.
- Select a Resource and its Attribute - Microsoft Internet Explorer:** This window contains two tables.

Resource	Collection
<input type="radio"/> Browser Capability	
<input type="radio"/> Category Count	
<input checked="" type="radio"/> Customer	Customer
<input type="radio"/> Date	
<input type="radio"/> Page Request	
<input type="radio"/> User Session	

Attribute	Type
<input type="radio"/> FNAME	Text
<input type="radio"/> LEVEL	Text
<input type="radio"/> LNAME	Text
<input type="radio"/> PASSWORD	Text
<input checked="" type="radio"/> USERID	Text
<input type="radio"/> [Empty]	Text

- Following is a picture of the rule after you have finished.



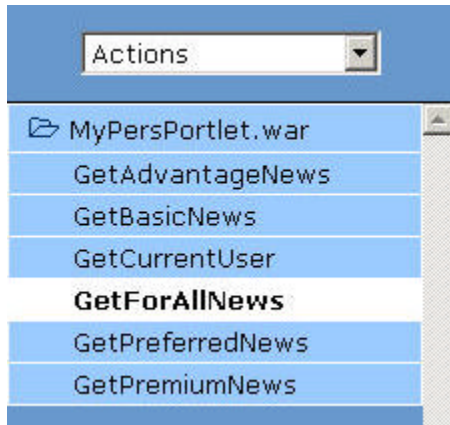
## \_\_\_ Create the GetBasicNews Action Rule

- Create the **GetBasicNews** action rule so that it looks like the following picture. For this rule and the remainder of the action rules you need to click the **Specify a value of type Text** radio button when setting the value field.

Action Properties	
Name	GetBasicNews
Comment	
Action Type	Select content
Content Type	Customernews
Action Description	
Select content whose Customernews.LEVEL is equal to basic order as is	

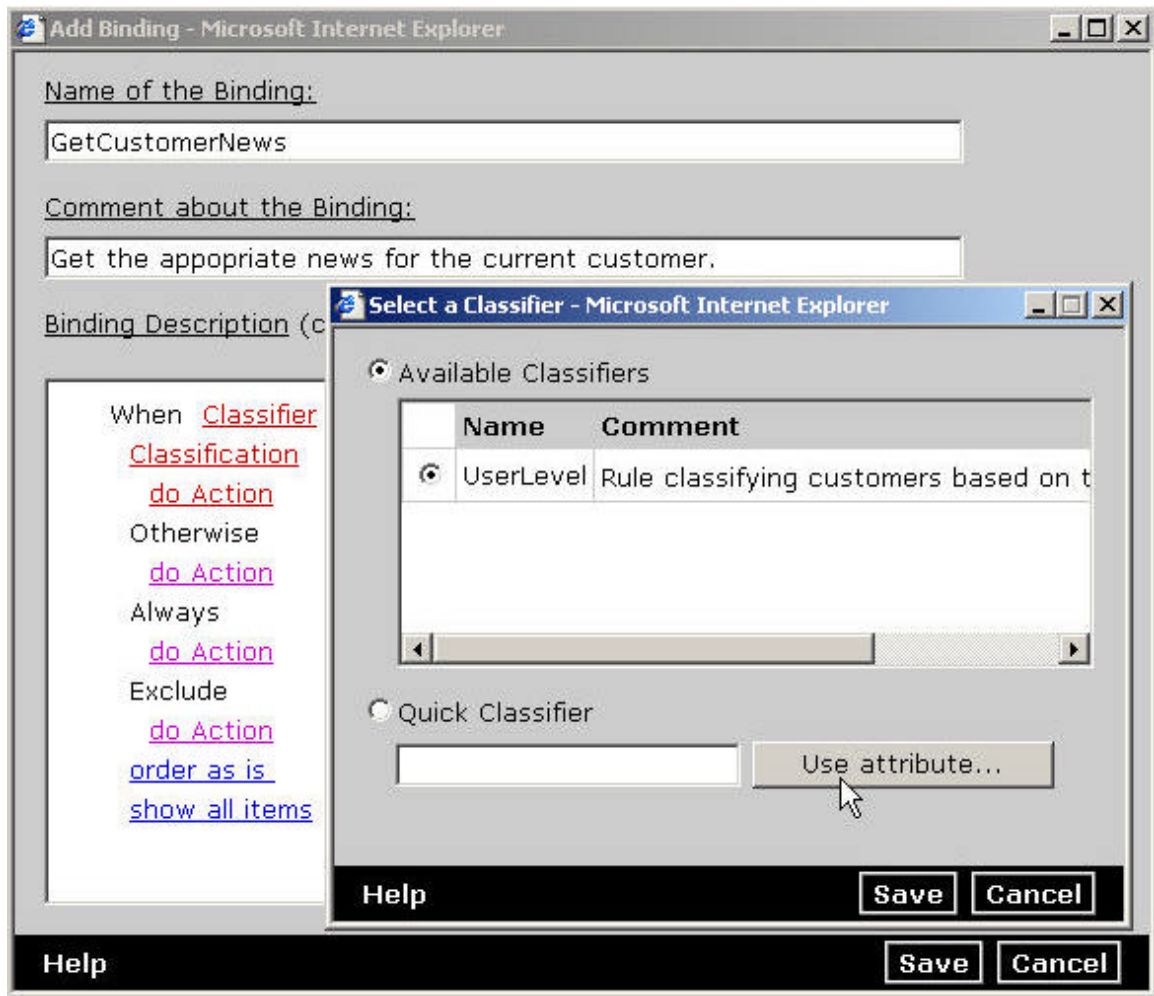
## \_\_\_ Create 4 Additional News Action Rules

- Create the 4 additional news action rules shown in the following picture. They are **patterned** after the GetBasicNews rule. For example the **GetPreferredNews** rule should select content whose level is equal to **preferred**. The only one that doesn't follow this pattern is the **GetForAllNews** rule whose value should be set to **all**, not forall!.

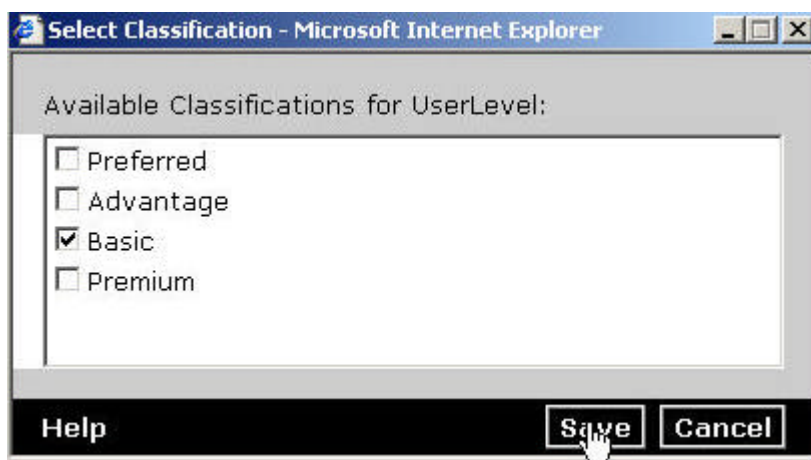


### \_\_\_\_ **Create the GetCustomerNews Binding Rule**

- Create the **GetCustomerNews** Binding rule using the following pictures as a guide. Notice that when you click on Classifier you get a dialog with a list of available classifiers. In this case the only one is UserLevel.
- Select **Bindings** from the **Rule Composer** list-box
- Click the + icon at the left

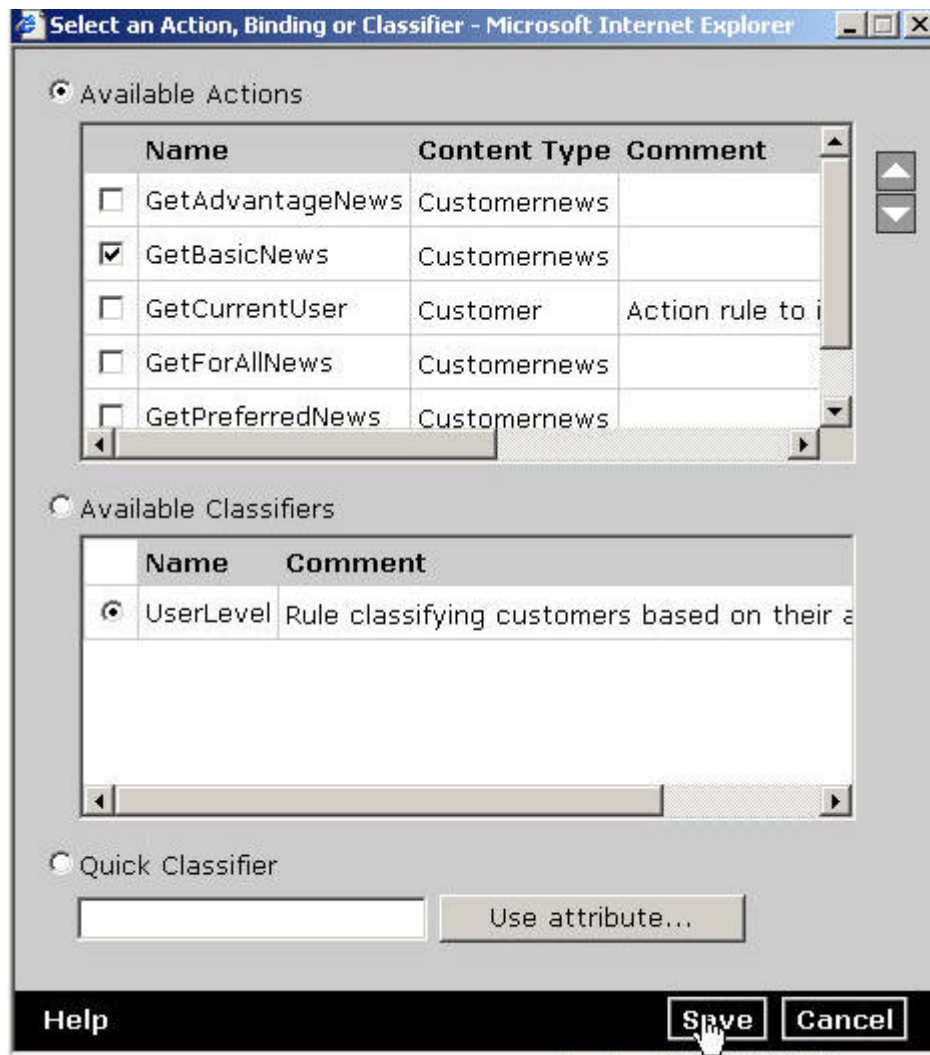


- When you click on **Classification** you are presented with a list of Classifications defined by the classifier you selected.



- Similarly when you click on **do Action**, you are presented with a list of available actions.





- Complete the **GetCustomerNews** binding rule using the following picture as a guide.



Binding Properties	
Name	GetCustomerNews
Comment	Get the appopriate news for the current customer.
Content Type	Customernews
Classifiers used by this binding	
<a href="#">UserLevel</a>	
Actions used by this binding	
<a href="#">GetBasicNews</a> , <a href="#">GetAdvantageNews</a> , <a href="#">GetPreferredNews</a> , <a href="#">GetPremiumNews</a> , <a href="#">GetForAllNews</a>	
Binding Description	
<pre> When UserLevel is   Basic     do GetBasicNews   Advantage     do GetAdvantageNews   Preferred     do GetPreferredNews   Premium     do GetPremiumNews   Otherwise     do Action   Always     do GetForAllNews </pre>	

## \_\_\_ **Assign the Rules to Content Spots in the Campaign Manager**

- Select the **Campaign Manager** tab in the workspace
- Select **Normal View** on the left side. This displays the list of content spots in this project. Notice that the rule column is empty.

Campaign Manager   Rule Composer   Preview Launcher		
<div> <div>Web site</div> <div>Priorities</div> <div>Normal View</div> </div>		
Rule usage when there are no active campaigns		
Content Spot	Content Type	Rule
getCurrentUserSpot	Customer	<a href="#">empty</a>
getCustomerNewsSpot	Customernews	<a href="#">empty</a>

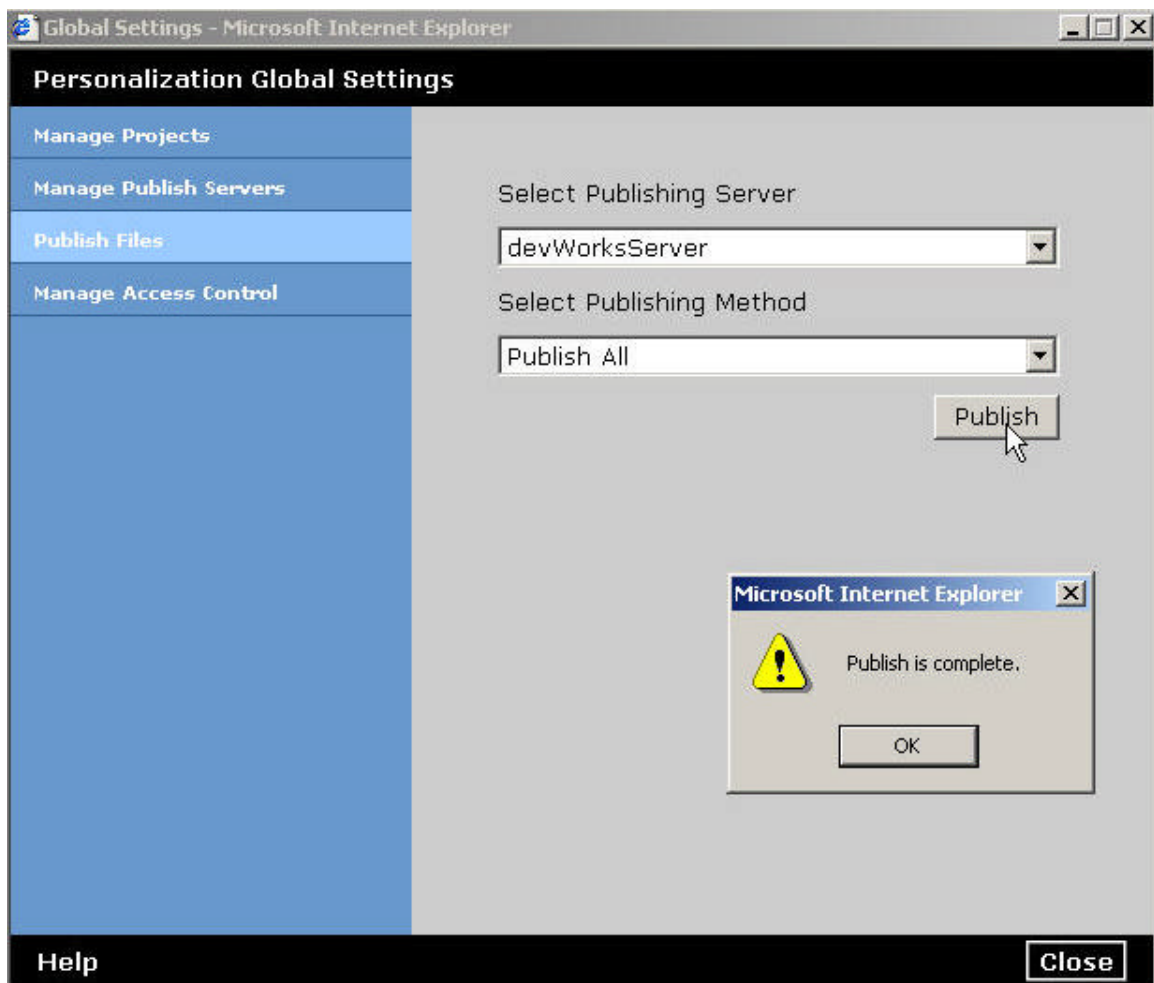
- Assign a rule to each of the content spots by clicking on the empty link in the rule column and then selecting a rule from the list of applicable rules. Use the following picture as a guide.

Campaign Manager   Rule Composer   Preview Launcher		
<div> <div>Web site</div> <div> <div>Priorities</div> <div>Normal View</div> </div> </div>		
Rule usage when there are no active campaigns		
Content Spot	Content Type	Rule
getCurrentUserSpot	Customer	<a href="#">GetCurrentUser</a>
getCustomerNewsSpot	Customernews	<a href="#">GetCustomerNews</a>

## \_\_\_\_ ***Publish the Rules and Campaign***

Now that we have all of our rules defined and assigned to content spots in the project we need to publish them.

- Click **Global Settings** at the right of the workspace
- Select **Publish Files**
- Make sure **devWorksServer** is selected and then click the **Publish** button



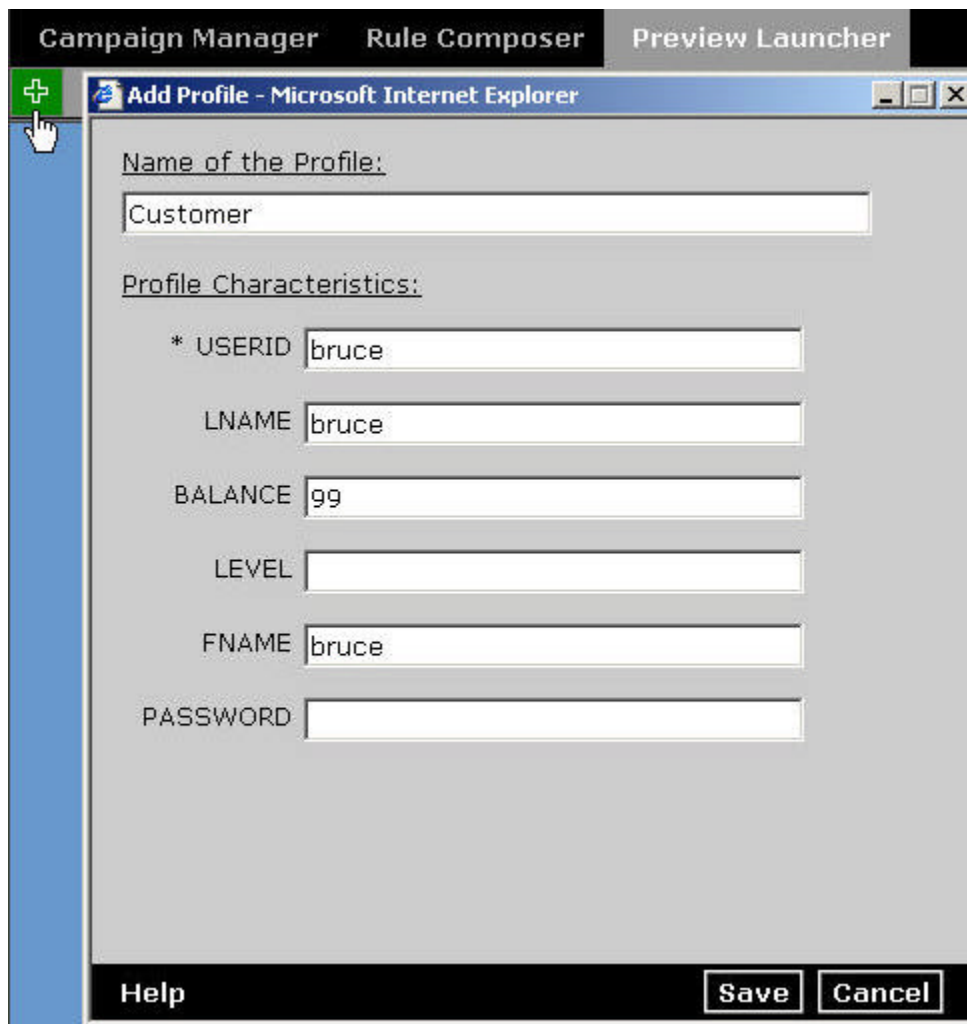
## ***Preview the news.jsp Page Using the Preview Launcher***

Next we will preview the news.jsp page to see what it looks like.

- In the workspace window select the **Preview Launcher** tab
- Click on the **plus sign** to add a profile for previewing.

You can add as many different profiles as you wish with different attributes and use them to verify that your rules are working as intended to personalize your content. In this lab we will create a profile named **Customer**.

- Use the following picture to fill out the profile

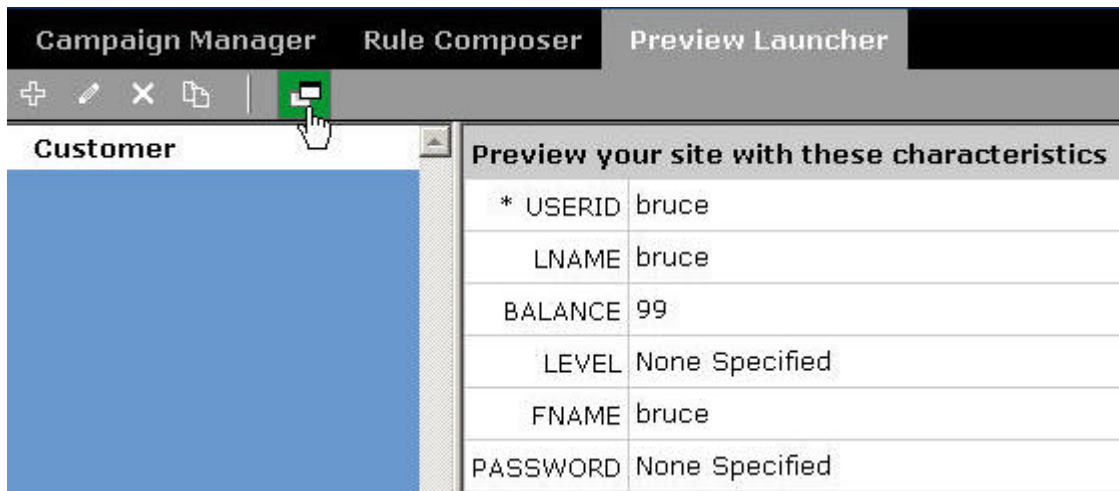


The screenshot shows the 'Add Profile' dialog box within the Campaign Manager application. The dialog has a title bar that reads 'Add Profile - Microsoft Internet Explorer'. It features a sidebar on the left with a green plus icon and a blue vertical bar. The main area contains the following fields:

- Name of the Profile:** A text box containing the word 'Customer'.
- Profile Characteristics:** A section containing several labeled text boxes:
  - \* USERID: bruce
  - LNAME: bruce
  - BALANCE: 99
  - LEVEL: (empty)
  - FNAME: bruce
  - PASSWORD: (empty)

At the bottom of the dialog, there are three buttons: 'Help', 'Save', and 'Cancel'.

- Once you have at least one profile defined you can preview your site by selecting the **preview icon** as shown in the following picture

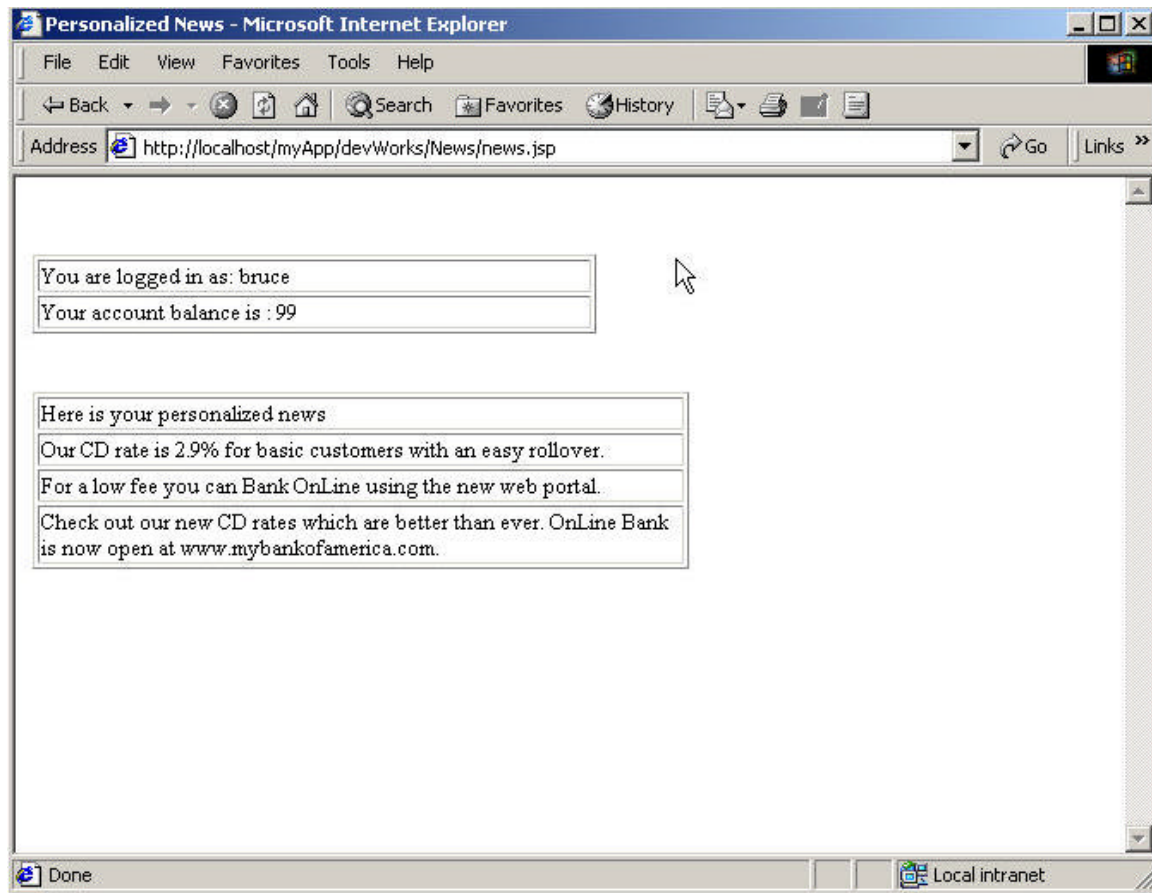


You will be presented with a dialog that allows you to choose a profile, set the time, and select which page you want to preview as shown in the following picture.

- Click OK and the page you are previewing will be displayed

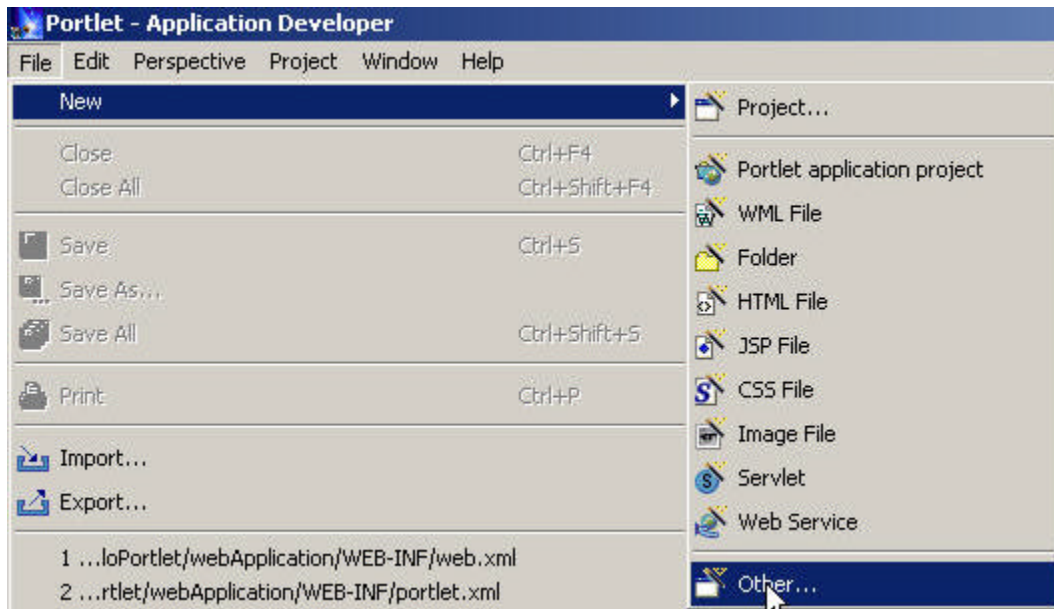


- You should see the following page

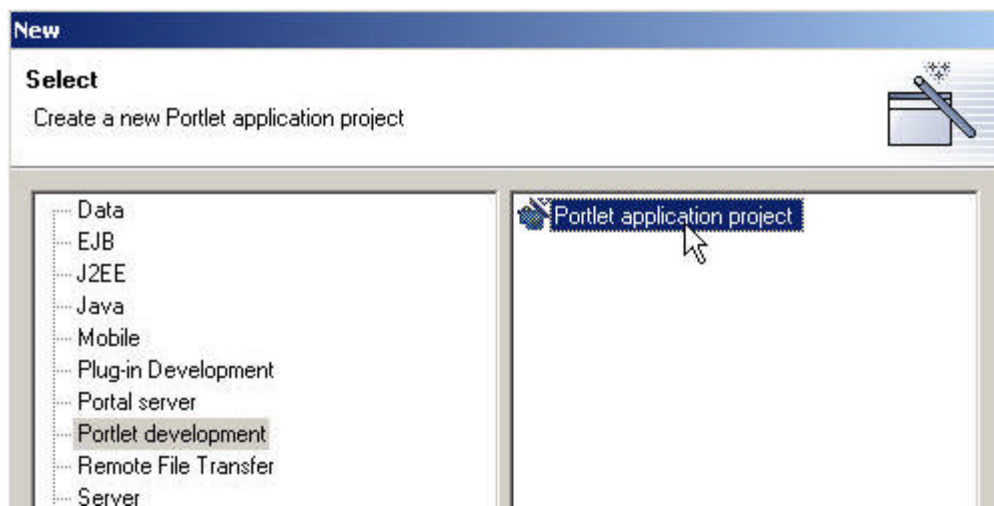


## ***Start WSAD and Create a New Project***

- Start WSAD using the WSAD icon on the desktop
- We have added the Portal Development plugin to WSAD. To get into the Portlet perspective, select **Perspective->Open->Other...** and select **Portlet perspective**.
- Create a Portlet project by selecting **File->New->Other...** as shown below



- In the subsequent window select **Portlet development** in the left pane, **Portlet application project** in the right pane, and then press the **Next** button.



- Fill out the Project Window using the following picture as a guide. Again this picture was taken on a system where WSAD was installed on the D drive—make sure your location points to the C drive.
- Press the **Next** button

**Create a Portlet project**

**Define the Portlet project.**  
Specify a name and location for the Portlet project. Also, specify a new or existing EAR project that will refer to this Portlet project as a web module.

Project name:

☒ Use default location  
Location:

Enterprise Application project name:

Context root:

☐ Create CSS file

< Back   Next >   Finish   Cancel

- Select the **MVCPortlet** radio button and press the **Next** button

**Create a Portlet project**

**Portlet selection**  
Create a Portlet project with the sample MVCPortlet.

Portlet type:

☐ None  
☐ AbstractPortlet  
☐ JSPPortlet  
☒ **MVCPortlet**  
☐ ServletInvokerPortlet  
☐ XSLPortlet

- Use the following picture as a guide to change the **Controller class name base**. You shouldn't need to change any of the other fields.
- Press the **Finish** button to finish creating the project



**Create a Portlet project**

**MVCPortlet parameters**  
Enter the properties of the MVCPortlet.

Portlet application name: MyPersPortlet application

Portlet name: MyPersPortlet portlet

Concrete portlet application name: Concrete MyPersPortlet application

Concrete portlet name: Concrete MyPersPortlet portlet

Default locale: en English

Concrete portlet title: Concrete MyPersPortlet portlet

Controller class name base: MyPersPortletController

Markups: ☒ html ☐ chtml ☐ wml ☐ vxml

< Back Next > Finish Cancel

## \_\_\_\_ ***Edit MyPersPortletController\_html.java***

- Expand the **MyPersPortlet** project you just created and double-click the **MyPersPortletController\_html.java** file to edit it
- Use the following picture to complete the **doView()** method and add the missing import statements
- Notice that the method sets the current user into the session using **pzn.UserName**. This is necessary so that the personalization runtime can find the current logged in user.

---

```

import java.io.*;

import org.apache.jetspeed.portlet.*;
import org.apache.jetspeed.portlets.*;
import com.ibm.wps.portlets.*;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpSession;

public class MyPersPortletController_html extends AbstractPortletController {

    public void init( PortletConfig portletConfig ) throws UnavailableException {
        super.init( portletConfig );
    }

    public void doView( PortletRequest request, PortletResponse response ) throws :
        PortletContext portletcontext = getPortletConfig().getContext();
        String currentUserName = request.getUser().getUserID();

        System.out.println("##### USERID: " + currentUserName);
        //put the user name in the session for personalization use
        HttpSession httpsession = request.getSession();
        httpsession.setAttribute("pzn.userName", currentUserName);

        portletcontext.include("/devWorks/News/news.jsp", request, response);
    }
}

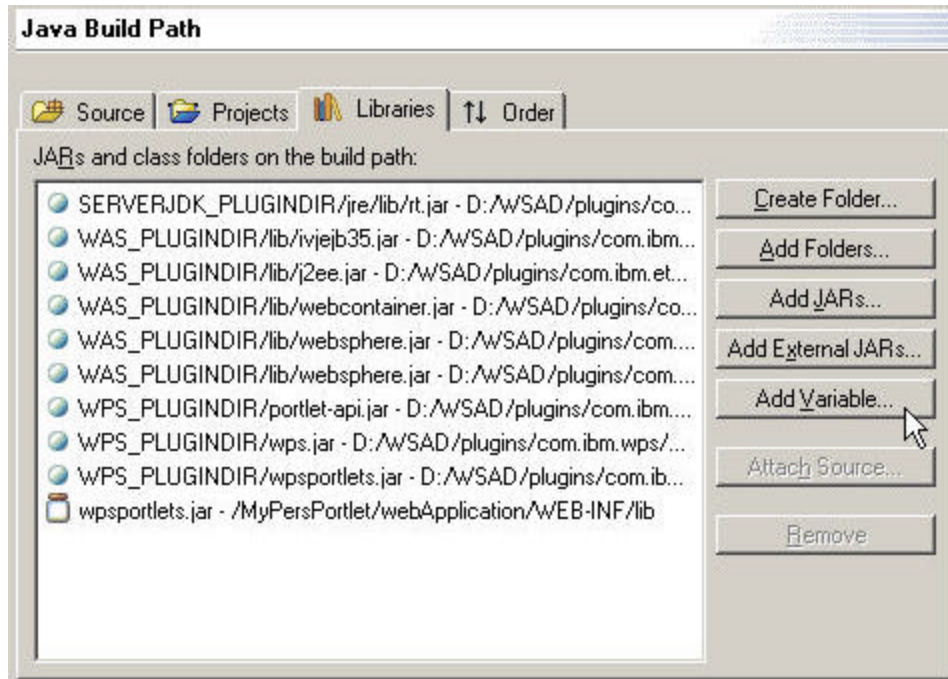
```

## \_\_\_\_ **Update the Project Classpath**

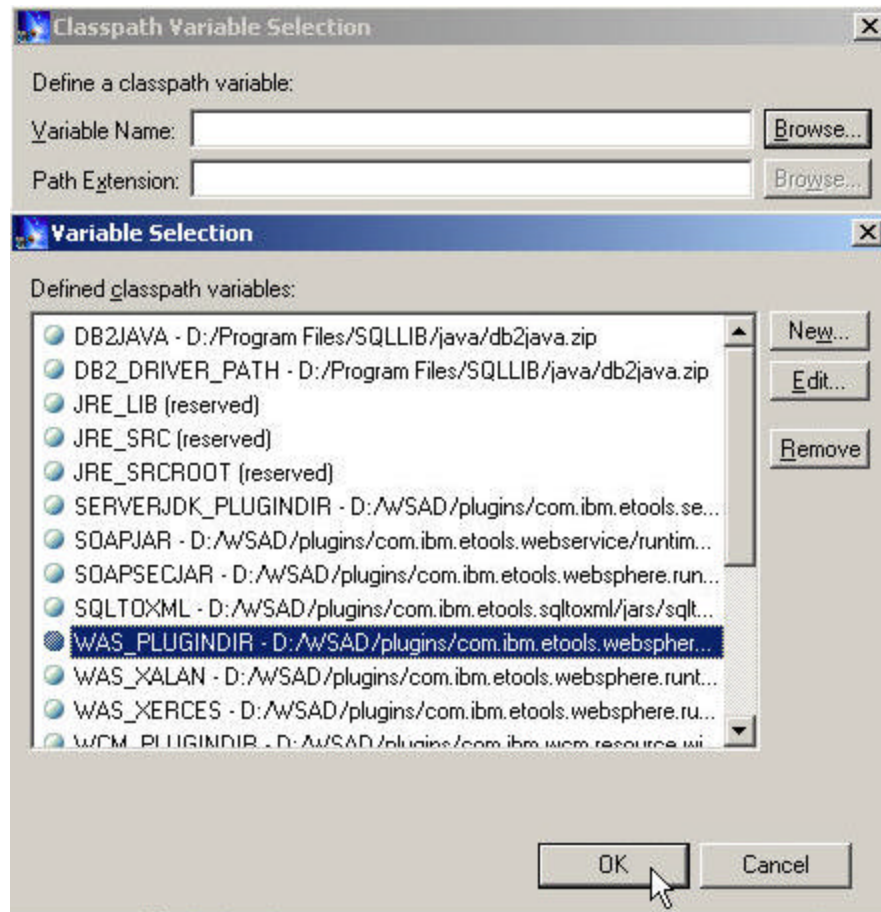
- Right-click on the **MyPersPortlet** project, and click **properties**
- Select **Java Build Path** from the list
- Select the **Libraries** tab and you will see the following picture
- Add each of the variables in the table below to the project. Notice that the variable name is different for the last 2 entries.

Variable Name	Path Extension
WAS_PLUGINDIR	lib/cm.jar
WAS_PLUGINDIR	lib/webtlsrn.jar
WAS_PLUGINDIR	lib/ws-base-extensions.jar
WCM_PLUGINDIR	lib/CTACompile.jar
WCM_PLUGINDIR	lib/databeans.jar

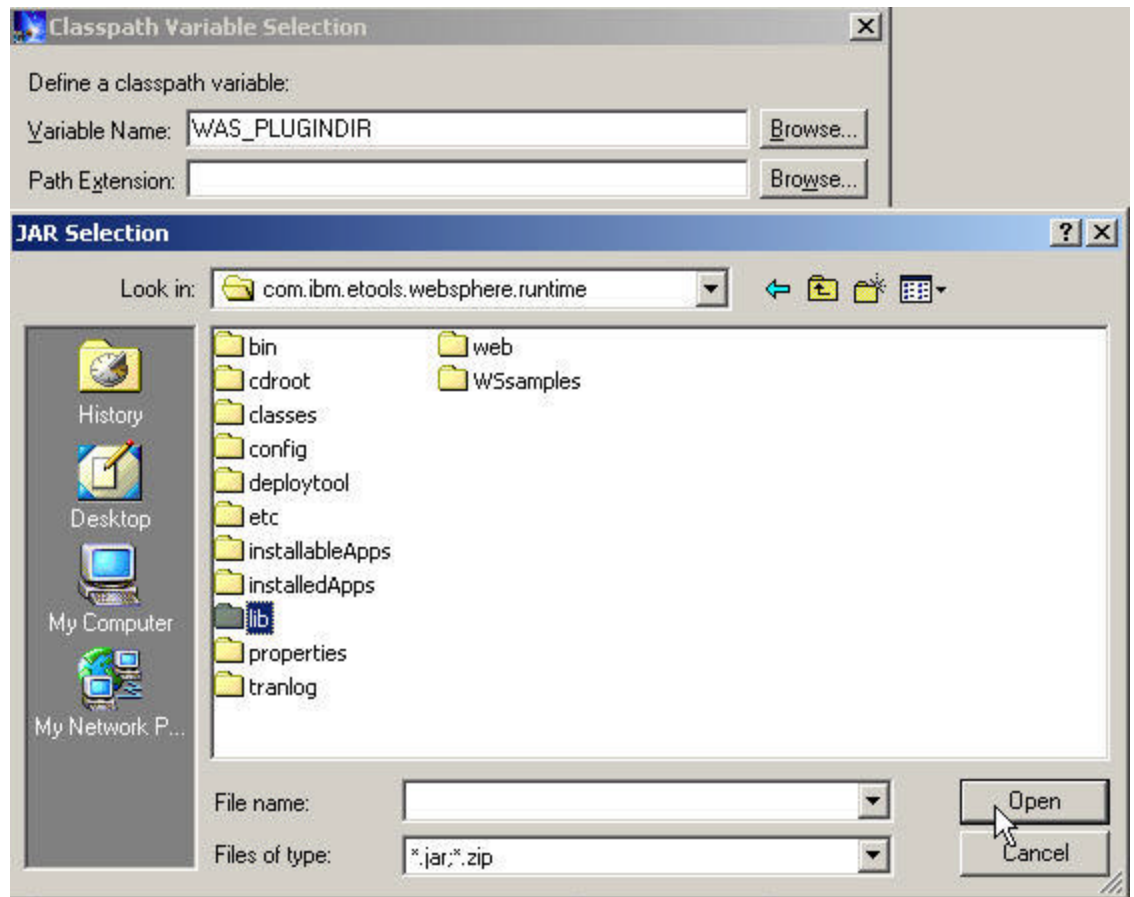
- Use the following steps as a pattern to add each of the variables in the table
  - Press the **Add Variable...** button



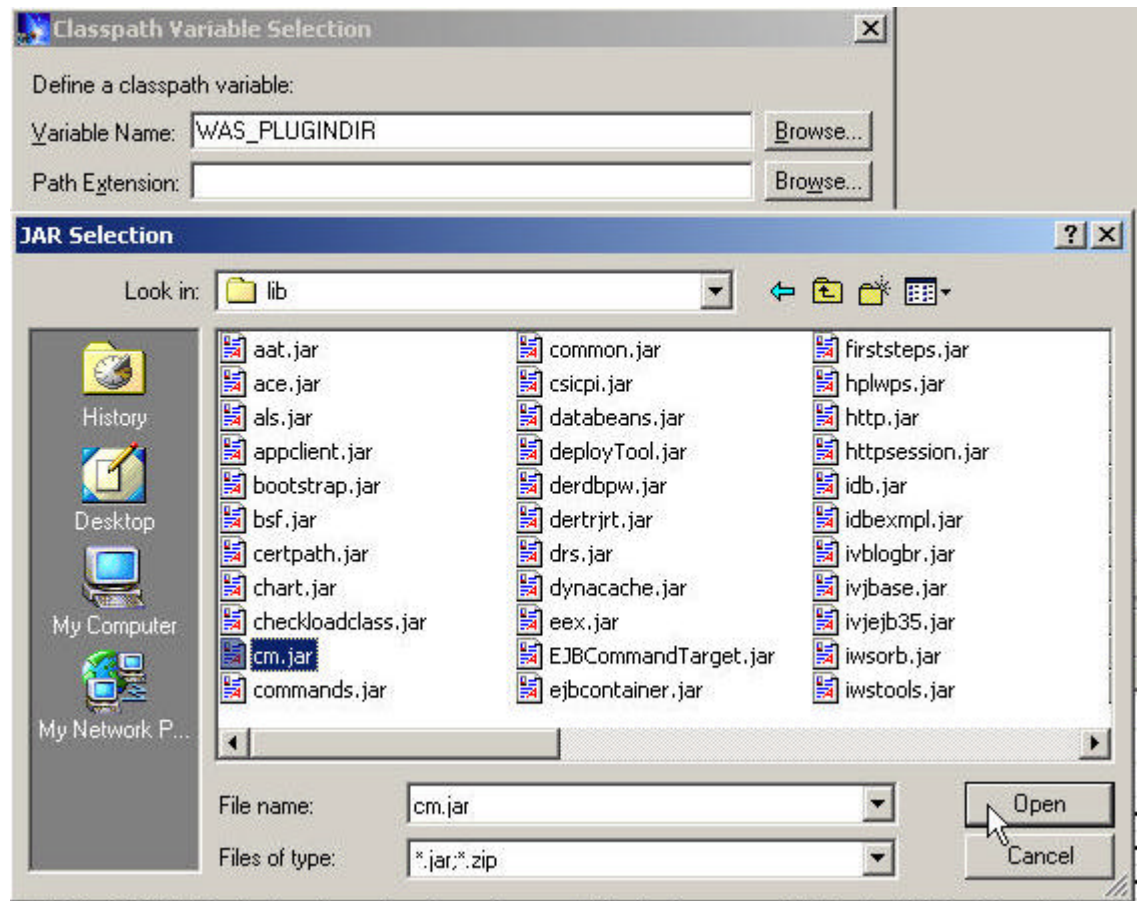
- Select the **WAS\_PLUGINDIR** variable as show below and press the **OK** button



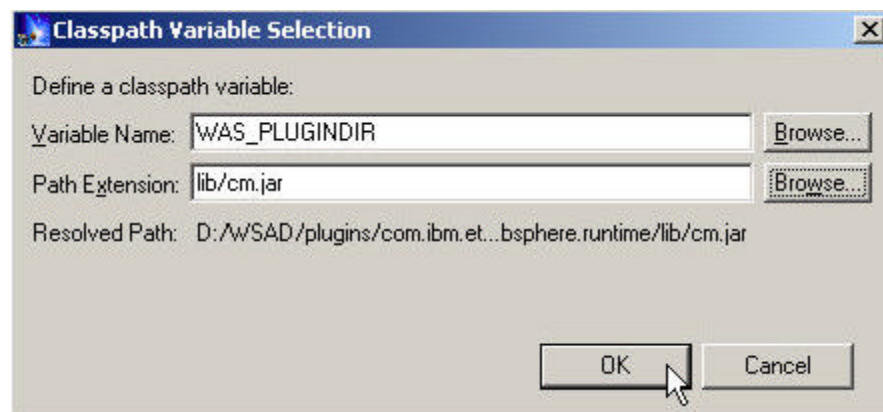
- Press the **Browse...** button next to the **Path Extension:** entry field
- Select the **lib** directory and press the **Open** button



- Select **cm.jar** and press the **Open** button

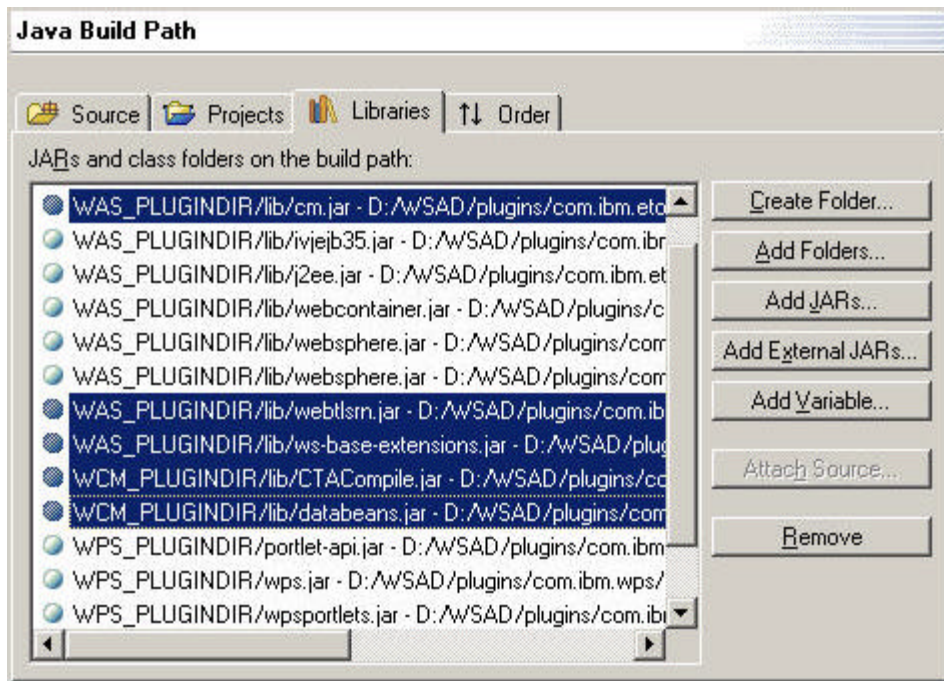


- Finally press the OK button to complete this addition to the classpath

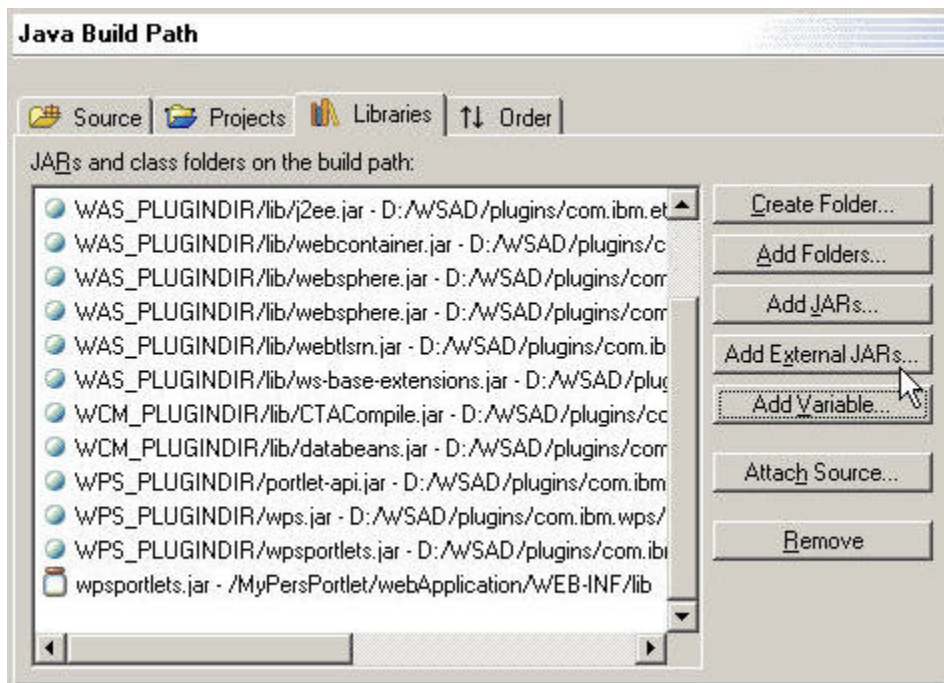


- After adding all 5 variables from the preceding table the **Java Build Path** window should look like the picture below





- Next press the **Add External JARs...** button

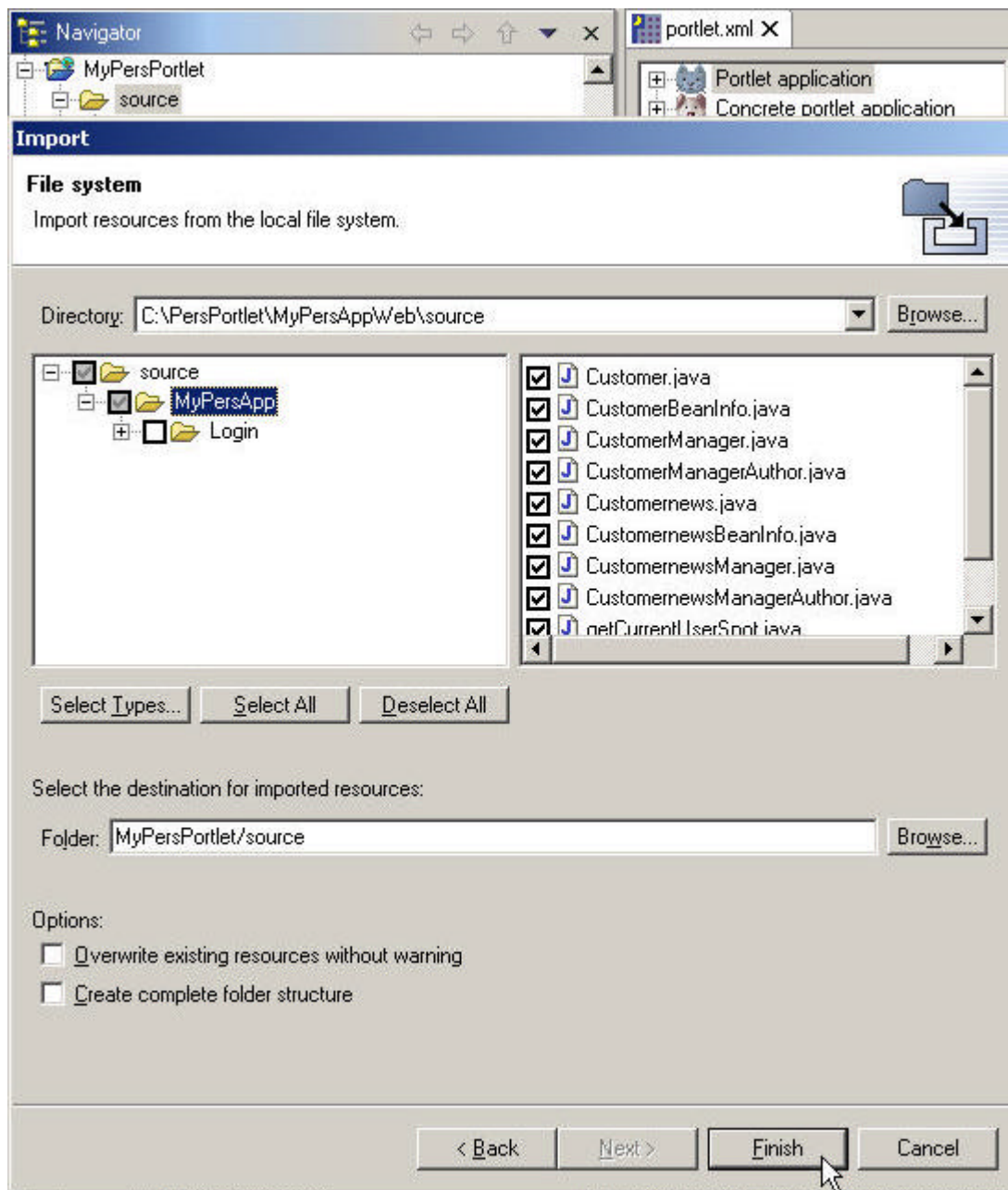


- Navigate to the **C:\WebSphere\AppServer\personalization\lib** directory
- Select **prCommon.jar** and press the **Open** button
- Press the **OK** button on the **Properties for MyPersPortlet** window to finish updating the Java Build Path

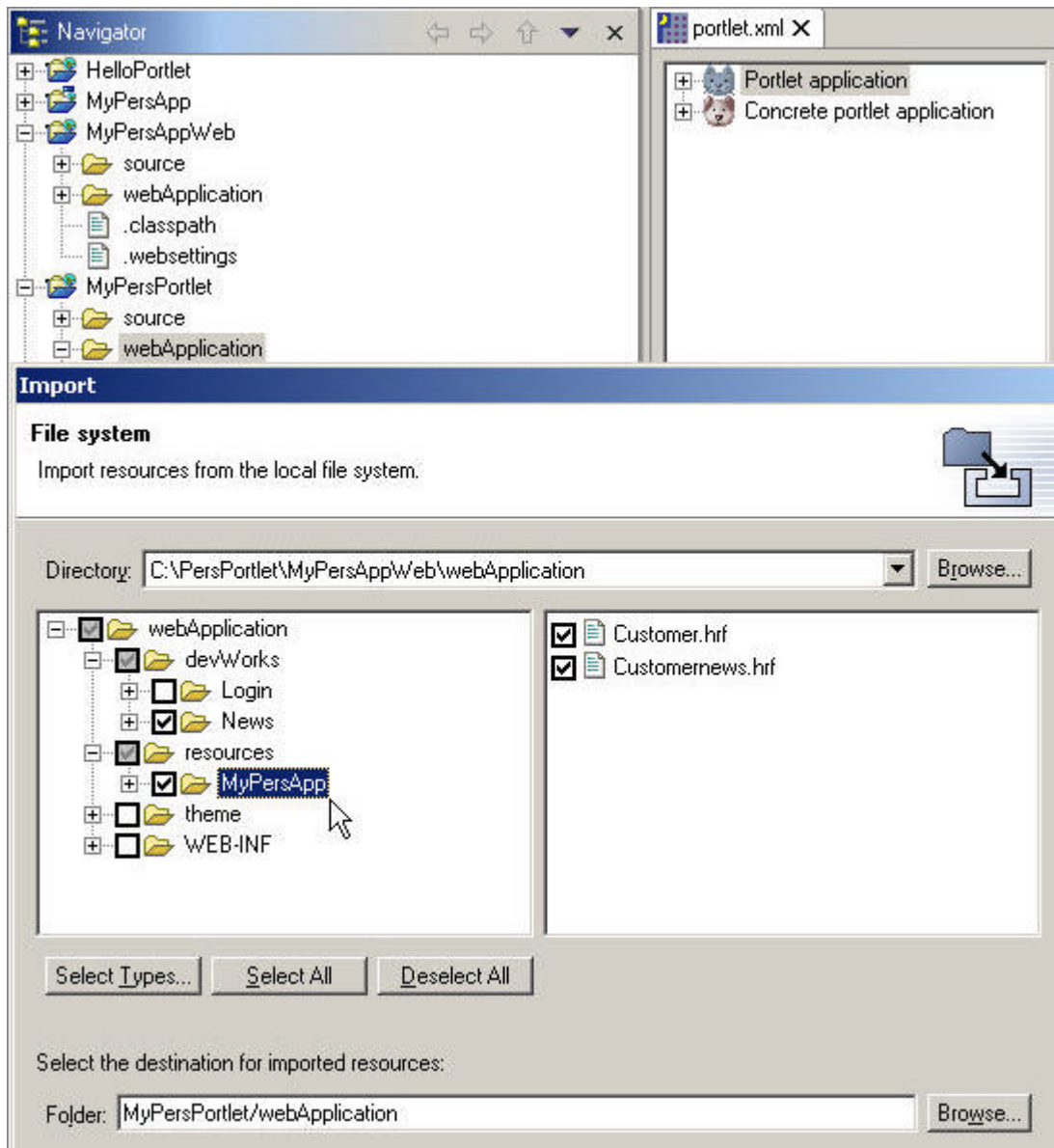


### ***\_\_\_\_ Import the Artifacts Previously Exported From the MyPersAppWeb Web Application***

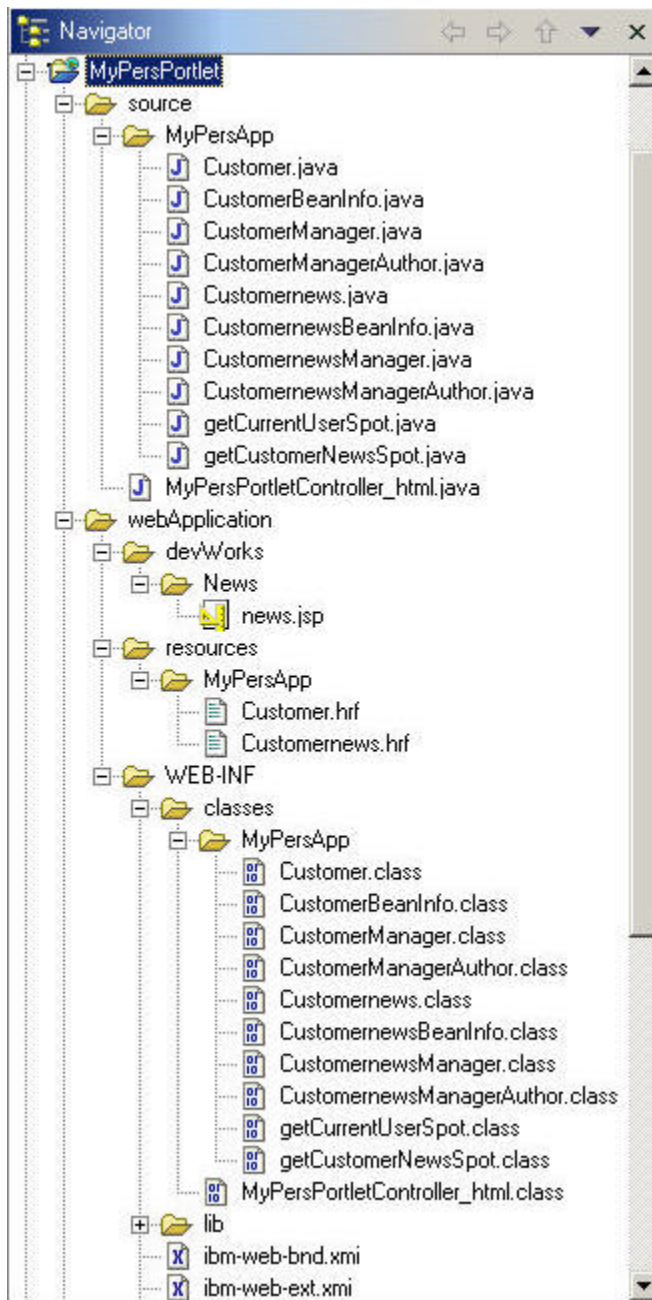
- Highlight the **source** directory and select **File->Import**
- On the Import window select **File system** and press the **Next** button
- Browse to the directory shown in the picture below. This is the temporary directory where we previously exported files from the **MyPersAppWeb** web application.
- Select everything in the **source** directory except for the **Login** folder as shown in the following picture
- Press the **Finish** button to import all of these files



- Highlight the **webApplication** directory and select **File->Import**
- On the Import window select **File system** and press the **Next** button
- Browse to the directory shown in the picture below
- Select the **webApplication\resources\MyPersApp** directory, and the **webApplication\devWorks\News** directory as shown below
- Press the **Finish** button to import all of these files

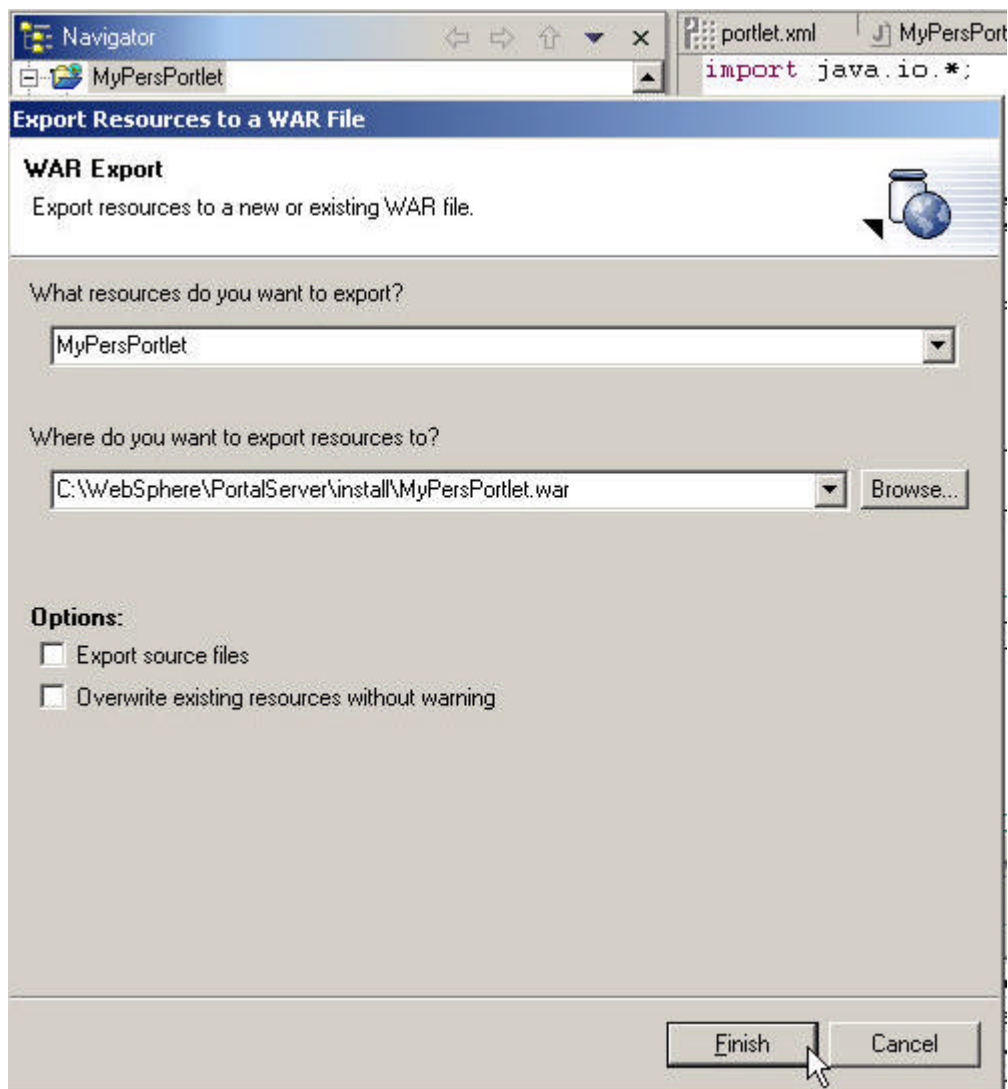


- The following picture is how your project should look after finishing all of the steps.



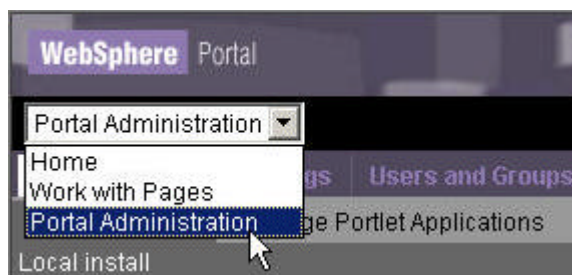
### **Create the MyPersPortlet War File**

- Select the **MyPersPortlet** web application, right-click and select **Export WAR...**
- Use the following picture to fill out the next window
- Press the **Finish** button

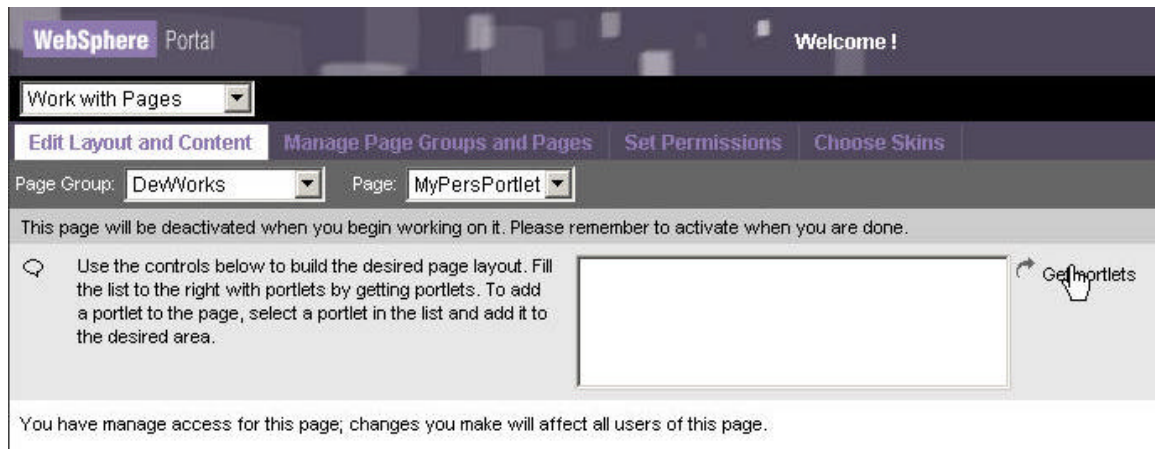


### ***Install the MyPersPortlet WAR File Into the Portal***

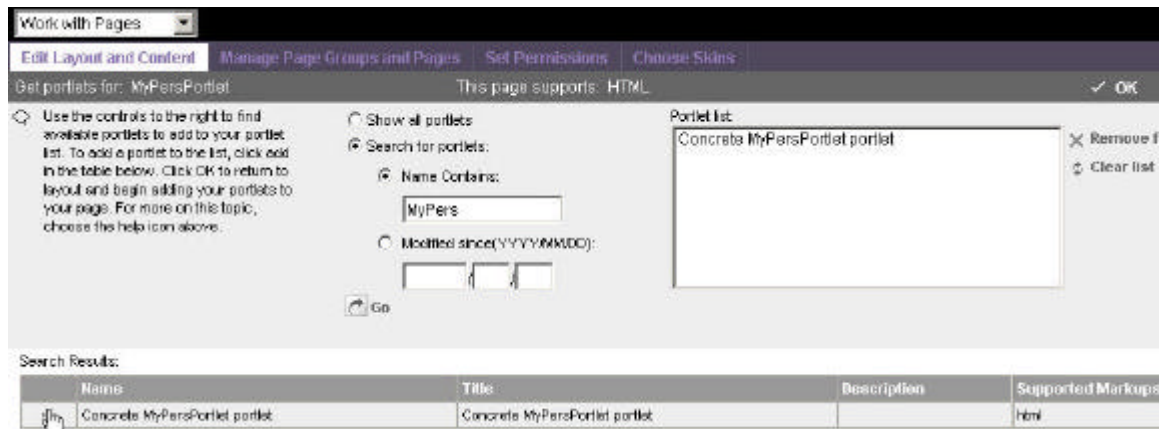
- Start Internet Explorer and use the provided link to start the Portal home page
- Login in as **wpsadmin** with password of **wpsadmin**
- Select the **Portal Administration** page group using the combo-box on the left



- Press the **Browse** button and browse to the directory you exported the WAR file to in the previous step
- Press **Next** on the right side and then **Install** to install the portlet
- Select the **Work with Pages** page group in the combo-box
- Select the **Edit Layout and Content** tab
- Select the **DevWorks** page group and the **MyPersPortlet** page in the 2 combo-boxes
- Click the **GetPortlets** icon to bring up the following page

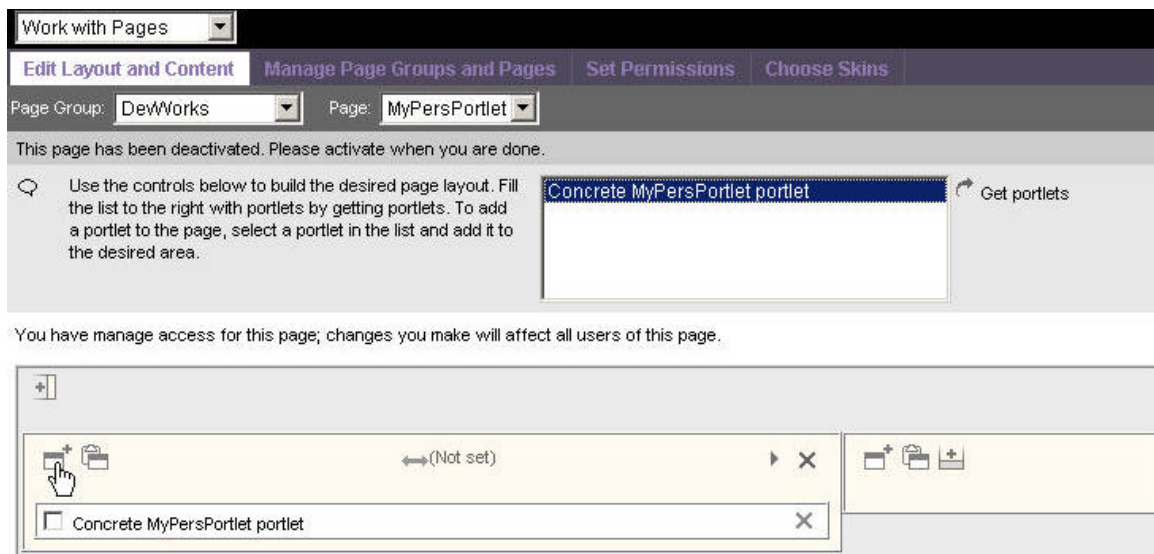


- Click the **Search for portlets:** and **Name Contains:** radio buttons
- Fill in the entry field with **MyPers**
- Click the **Go** icon to get the list of portlets containing the string **MyPers**
- Select **Concrete MyPersPortlet portlet** by clicking the plus sign, to add it to **Portlet list:** as shown below
- Click the **OK** icon to the far right



- On the next screen highlight the **MyPersPortlet** in the list as shown, and click on the icon with the hand pointing to it in the picture below. This adds the portlet to the page.





- Press the **Activate** button at the far right to activate the page
- Select the **DevWorks** page group in the combo-box at the top of the page and you will see the portlet you just installed
- Next, go to the **Portal Administration** page group
- Click on the **Security** tab
- Select the **Special Groups** radio button and select **All authenticated users** from the list-box
- Press the **Go** icon at the bottom left
- A table of all portlets and their permissions is returned. For the **Concrete MyPersPortlet portlet** check the **View** radio button and then click on **Save** at the top of the page.
- Log out of the portal and proceed to the next step

### Log in as Several Different Users and Notice How the News is Different Depending on the User

Make sure you log in as different users to see that the news displayed is related to your level. We enrolled the valid users for you in the portal. The passwords are the same as the userids. The users that are in the customer table are as follows:

- bruce
- cathy
- johnny
- mario
- michael
- sally



## \_\_\_\_ **Machine Setup**

This hands-on lab was conducted in May 2002 at the DeveloperWorks Live 2002 conference. This appendix has been added to describe how to run this lab outside of the conference lab environment. Following is a list of software and setup requirements:

- IBM WebSphere Portal V4.1 Enable Beta and all of its prerequisites. (WebSphere Portal V4.1.1 electronic GA is now available. We highly recommend you use this instead of the Beta. The Portal Toolkit is included with this.)
- IBM WebSphere Studio Application Developer V4.03
- IBM WebSphere Personalization wizards for Studio Application Developer (These are available from the WebSphere Personalization web site in the Support & downloads section)
- IBM Portal Toolkit for WebSphere Studio Application Developer Beta (This is not widely available at this time. If you can't obtain this plugin, you can use the MyPersPortletInitial.ear provided with this lab to get you started on the 2<sup>nd</sup> part of the lab.)
- You must create the DB2 Sample database and then run the DDL we provide in **MyPersPortal.sql** to build and populate the tables used in this lab.
- You must create a JDBC Driver named **PersPortletDriver**, and a Data Source named **PersPortletDS** with a JNDI name of **jdbc/myperspportlet\_ds**.
- You must create a Page Group Named **DevWorks**, a page in this page group named **MyPersPortlet**, and give access rights to both for all users.
- You must create the valid users documented in the last step of this lab by clicking on **Sign up** on the main Portal page and filling out the registration form.