

Implementing WA Dynamic Dropdowns in the BlueSky Footwear Sample Store

Introduction:

The functionality in the search form "shoeSearchDD" in the BlueSky Footwear Sample Store that comes with WA eCart, WA eCommerce Suite and WA SuperSuite is not included in the sample store because it requires the WA Dynamic Dropdowns to implement, and not all of these products include WA Dynamic Dropdowns.



Find Your Shoe:

Select Your Gende ▼

Select Shoe Type ▼

Select a Shoe ▼

Search:

We've included some content for the menus, but to use the Dynamic Dropdowns you'll need to first open the Template files that contain this form and then remove the Call To JavaScript behavior that is attached to the List Menu elements, select, select2 and select3.



Click on the first List/Menu element and open the Behaviors panel. An onChange event will appear in the list. Select that event and click the Delete (-) button on the Behaviors panel to remove the event.

Select the event:



Click the Delete button:

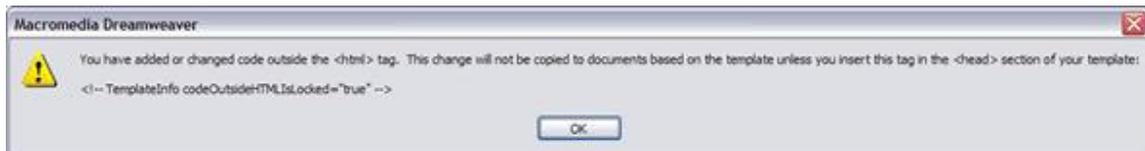


Prerequisite:

Before performing this operation check the opening HTML tag for this comment:

```
<!-- InstanceBegin template="/Templates/Main.dwt.asp"
codeOutsideHTMLIsLocked="false" -->
```

Make sure that the codeOutsideHTMLIsLocked="false" is set... if it's set to "true" you will not be allowed to insert server side code that is found before the <html> and after the </html> tags



Though this allows you to add code to your individual pages, one offshoot of this setting is that it causes alerts to appear when you add any code outside the HTML tags in the template file. The alerts inform you that the code you are inserting will not be used in the files that the templates affect. To work around this we'll create snippets of each code block that we create so that we can use them throughout the site.

Step 1: Add Recordsets:

Then you need to add three Recordsets: First to return data from the ProdCategory table which supplies the Mens and Womens dynamic labels and CatID values to populate the first List/Menu; Second to return the ProdSubCat table names and values along with the SubCatID to filter and populate the second List/Menu; Third to return the Products table along with the ProdID, ProdName, ProdCatID, ProdSubCatID which will be used to filter and populate the third List/Menu. A fourth optional recordset can be added to populate the

second List/Menu with values so that shoppers can select from that menu to filter and populate the third menu without selecting an item from the first menu.

First Recordset –rsGender:



The screenshot shows a 'Recordset' dialog box with the following configuration:

- Name: rsGender
- Connection: connBlueSky
- Table: ProdCategory
- Columns: All (selected)
- Filter: None
- Sort: None

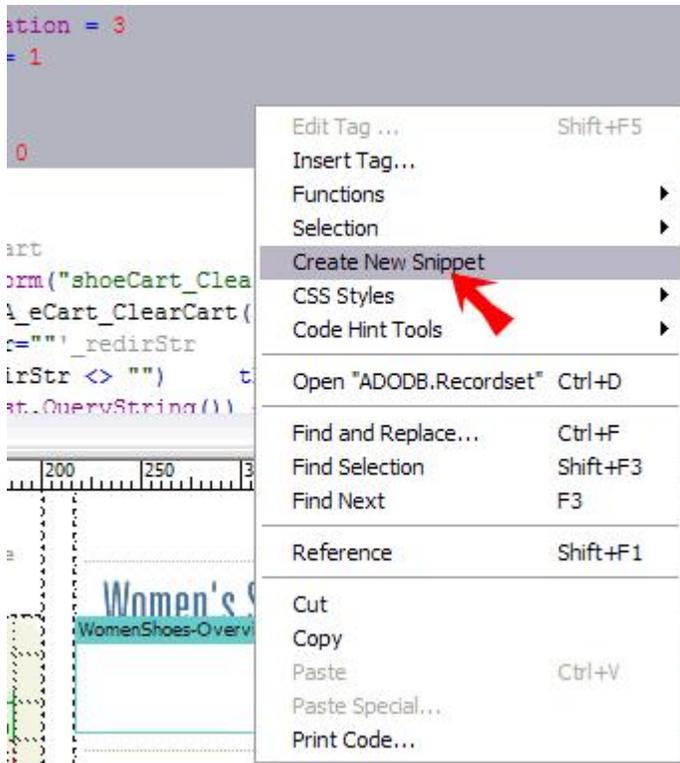
Buttons on the right include OK, Cancel, Test, Advanced..., and Help.

Code generated from rsGender recordset:

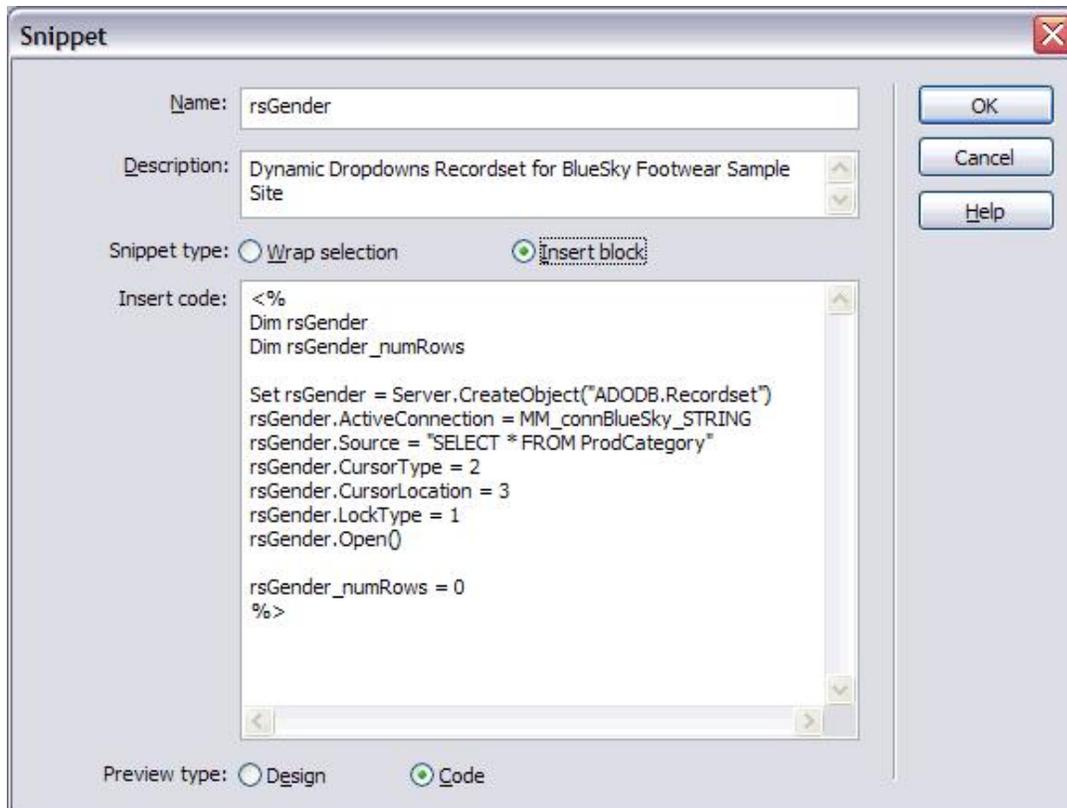
```
5 <<
6 Dim rsGender
7 Dim rsGender_numRows
8
9 Set rsGender = Server.CreateObject("ADODB.Recordset")
10 rsGender.ActiveConnection = MM_connBlueSky_STRING
11 rsGender.Source = "SELECT * FROM ProdCategory"
12 rsGender.CursorType = 2
13 rsGender.CursorLocation = 3
14 rsGender.LockType = 1
15 rsGender.Open()
16
17 rsGender_numRows = 0
```

NOTE: All references to table names in the MySQL database should be lowercase.

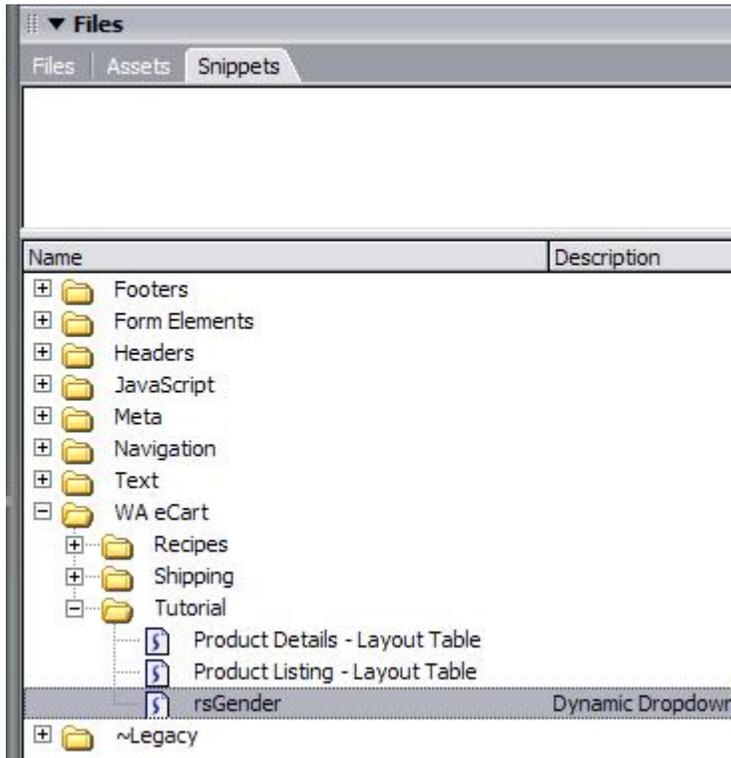
Next to prevent having to repeat this process throughout the site, create a new snippet from the code:



Snippet dialog settings:

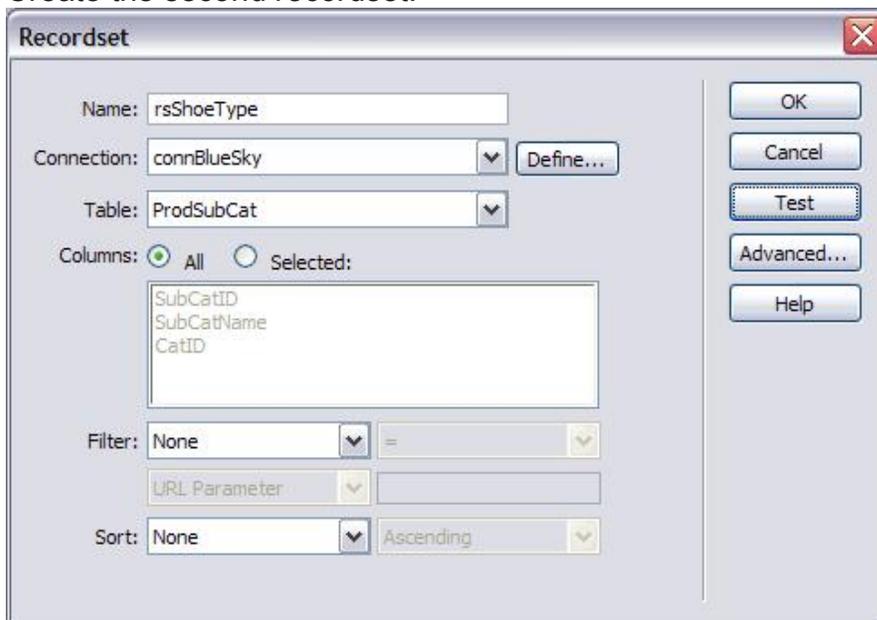


Saved snippet will appear in the Snippets panel under the currently-selected folder:

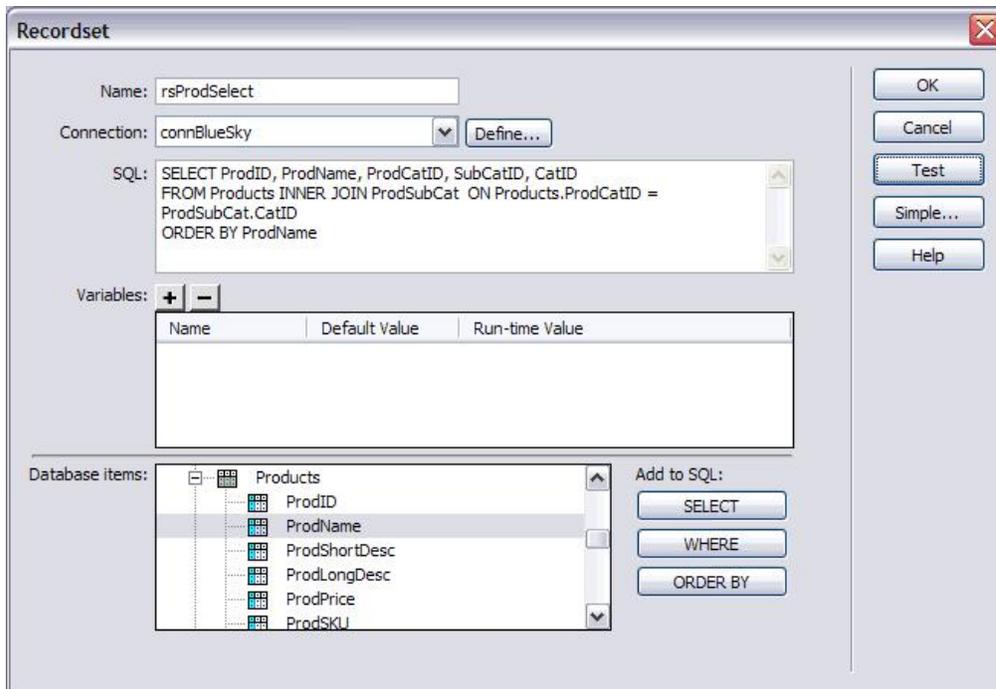


By creating snippets for the recordsets and their close tags (if needed) you can shorten your development time as the Dynamic Dropdowns are found in several template files.

Create the second recordset:



Create the third recordset:

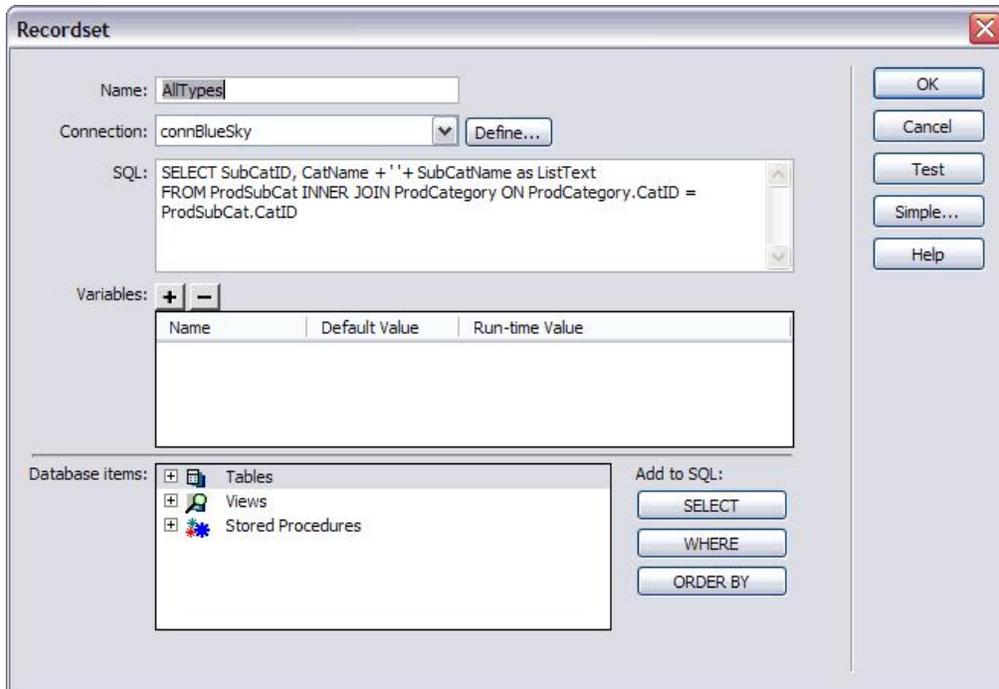


For PHP, the rsProdSelect SQL Statement should be:

PHP:

```
SELECT products.ProdID, products.ProdName, products.ProdCatID, products.ProdSubCatID,
probsubcat.SubCatID, prodsubcat.CatID, prodsubcat.SubCatName
FROM products INNER JOIN prodsubcat ON Products.ProdCatID = ProdSubCat.CatID
```

Optional: Create the fourth recordset:



For PHP, the SQL for this recordset should be:

PHP:

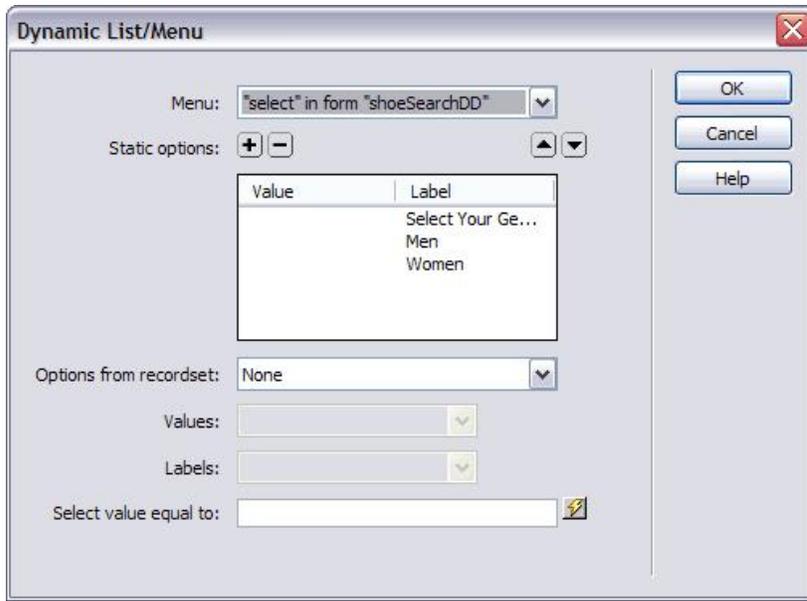
```
SELECT prodsubcat.SubCatID, CONCAT_WS(' ', CatName, SubCatName) AS ListText
FROM prodsubcat INNER JOIN prodcategory ON prodcategory.CatID = prodsubcat.CatID
```

also remember to create snippets for the code that closes the recordset at the end of the page if applicable for your server language. These will need to be inserted in the pages that will contain the Dynamic Dropdowns.

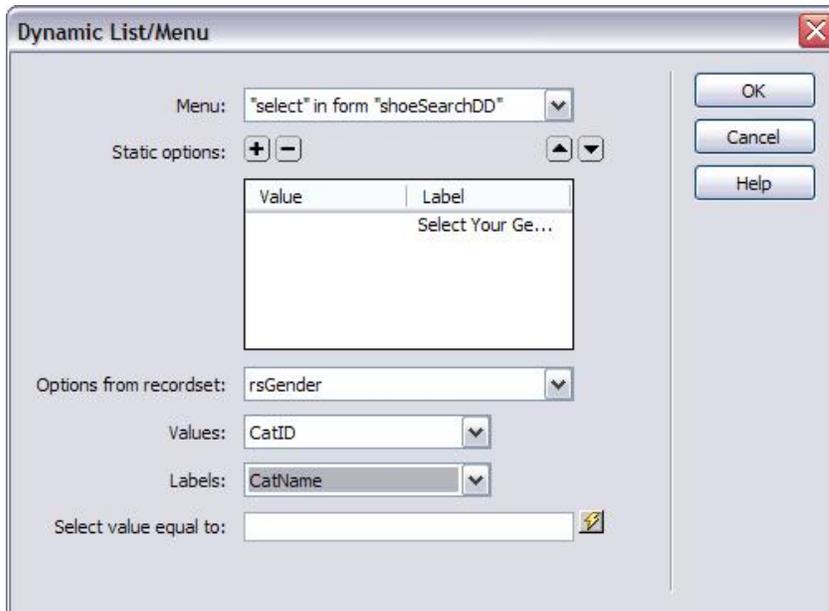
Step 2: Populate the first List/Menu:

With the Recordsets created, click on the first List/Menu named “select” and then in the Property inspector, click the Dynamic button to set a Dynamic List/Menu Server Behavior. Make sure the “select” form element is selected in the Menu dropdown and then remove the static labels “Men” and “Women”. In the Options from recordset dropdown, select your first recordset and then select CatID as the Values and CatName for the Labels from the corresponding dropdowns. Do not set the Select value equal to field. Click OK to close and save the Dynamic List/Menu dialog.

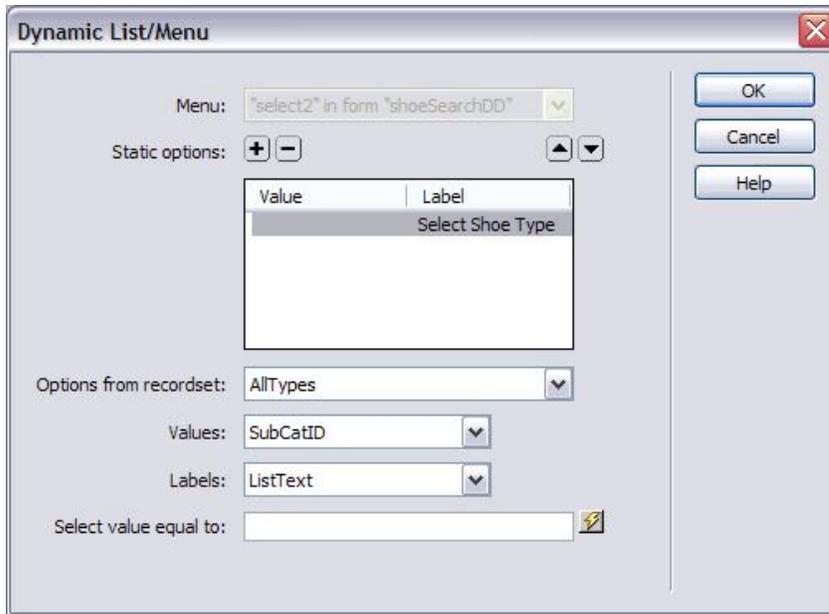
Before:



After:

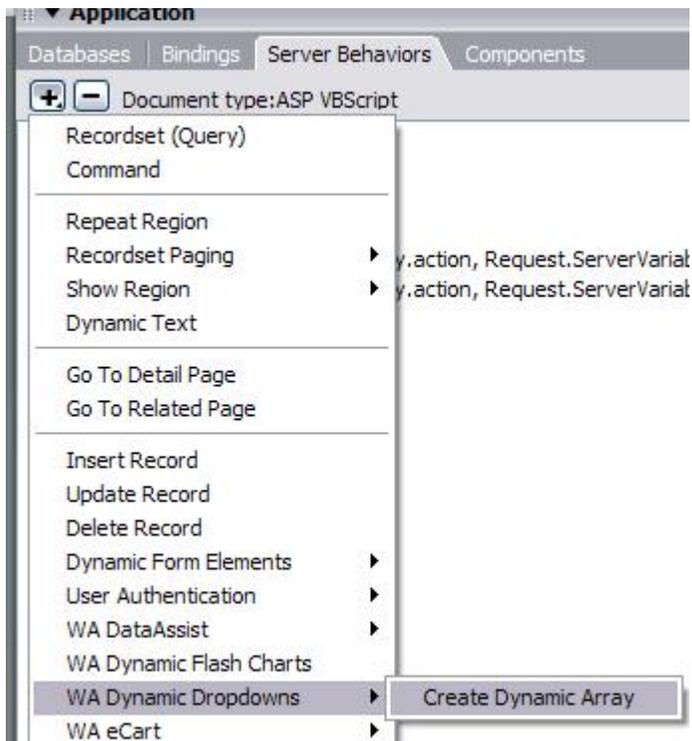


OPTIONAL: Create second Dynamic List/Menu to set default values for the second List/Menu before it is populated from the first List/Menu dynamically. To do this, select the second List/Menu in the search form named "select2" and then click the Dynamic... button on the Property inspector. Enter the following settings in the Dynamic List/Menu dialog:



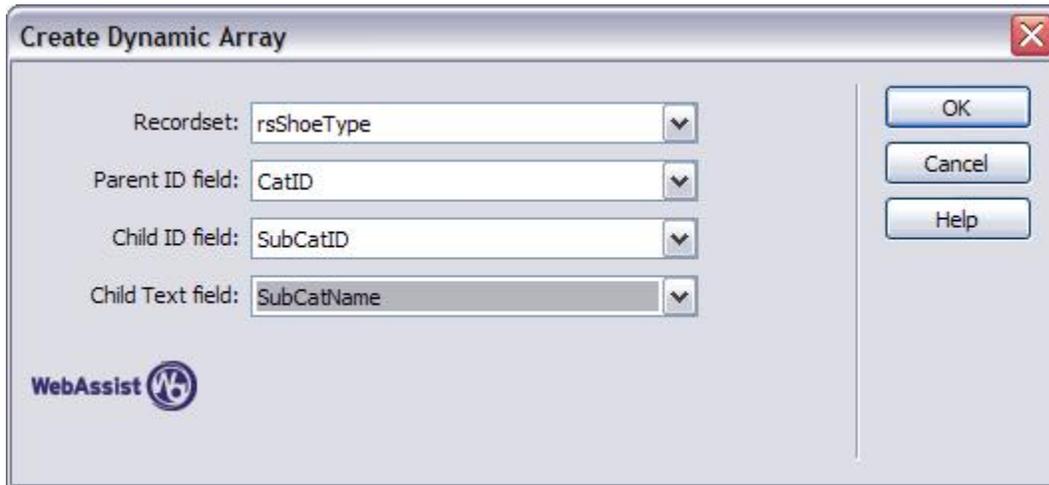
Step 3: Create Dynamic Arrays:

Once you've got your recordsets created, you'll then use the WA Dynamic Dropdowns Create Dynamic Array Server Behavior to create dynamic arrays for the second and third List/Menus based on the selection from the previous one,



the first menu selecting from between Mens and Womens and passing the CatID value which is in turn used to in the Create Dynamic Array to create an array

from the second recordset using the CatID as the parent ID field and the SubCatID as the child ID field.



The screenshot shows a dialog box titled "Create Dynamic Array" with a close button (X) in the top right corner. It contains four dropdown menus for configuration: "Recordset" is set to "rsShoeType", "Parent ID field" is set to "CatID", "Child ID field" is set to "SubCatID", and "Child Text field" is set to "SubCatName". On the right side, there are three buttons: "OK", "Cancel", and "Help". The "WebAssist" logo is located in the bottom left corner of the dialog.

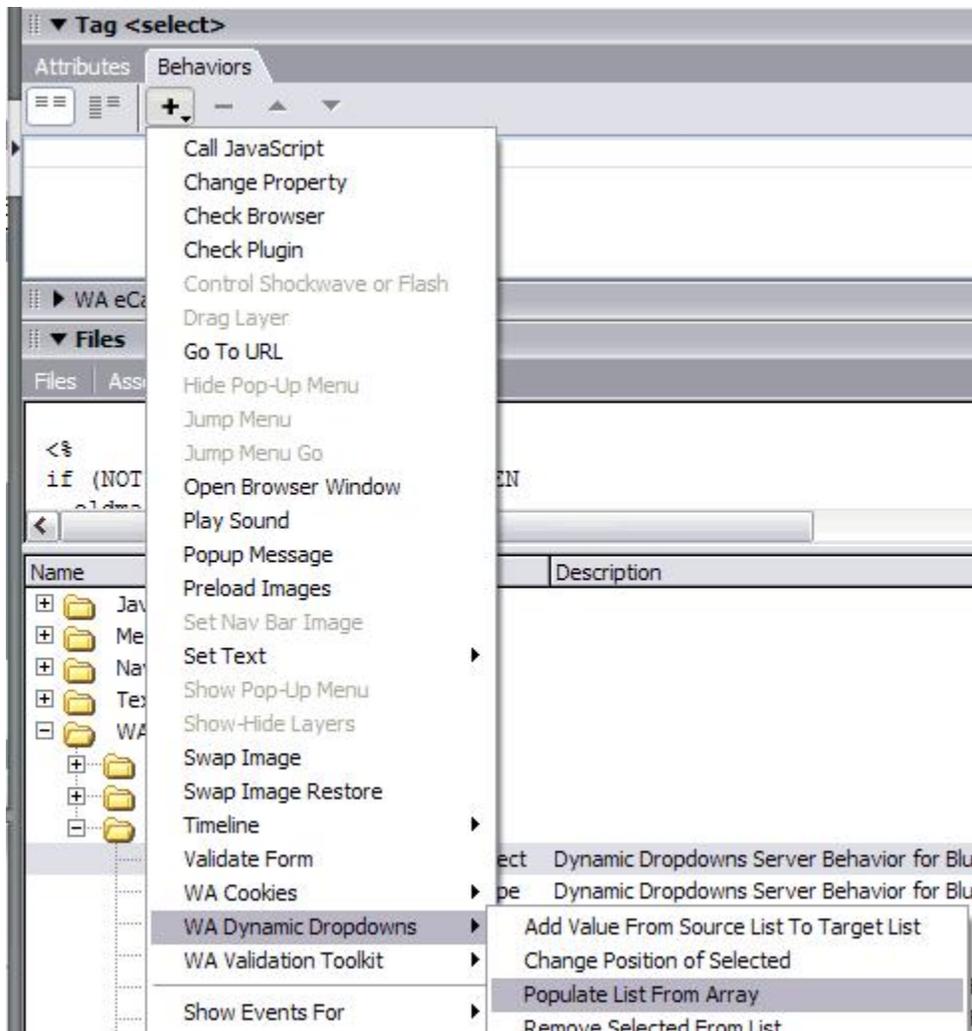
The SubCatName is used as the Child Text field value. Another Dynamic Array will be needed for the third List/Menu which uses the third recordset with the Parent ID field set to ProdSubCatID, the Child ID field set to the ProdCatID and the Child Text Field set to the ProdName.



The screenshot shows a dialog box titled "Create Dynamic Array" with a close button (X) in the top right corner. It contains four dropdown menus for configuration: "Recordset" is set to "rsProdSelect", "Parent ID field" is set to "SubCatID", "Child ID field" is set to "ProdID", and "Child Text field" is set to "ProdName". On the right side, there are three buttons: "OK", "Cancel", and "Help". The "WebAssist" logo is located in the bottom left corner of the dialog.

Step 4: Populate Lists From Array

Next you'll need to select the first List/Menu and apply a client Behavior from the Behaviors panel (NOT the Server Behaviors!) Select the first List/Menu and click the Add (+) button on the Behaviors panel and choose WA Dynamic Dropdowns > Populate List From Array.



The first command will use the first Create Dynamic Array as its array source and the first List/Menu as its source list. The target list of course is the second List/Menu. You can leave the Target list options blank for now. Their functionality is described in the WA Dynamic Dropdowns documentation. Click OK and notice that the Behavior is set to run on the onChange event of the first List/Menu.

Populate List From Array

Participants

Array source: rsShoeType

Source list: select "select" in form "shoeSearchDD"

Target list: select "select2" in form "shoeSearchDD"

Target list options

Leave top: 1 Leave bottom: 0

Prepend source text: Delimiter: :

WebAssist

OK
Cancel
Help

Next you'll create another Populate List From Array Behavior by selecting the second List/Menu and setting the Array Source as the second Create Dynamic Array. The source list is the second List/Menu and the target list is the third List/Menu. Again, you can leave the Target list options alone and click OK.

Populate List From Array

Participants

Array source: rsProdSelect

Source list: select "select2" in form "shoeSearchDD"

Target list: select "select3" in form "shoeSearchDD"

Target list options

Leave top: 1 Leave bottom: 0

Prepend source text: Delimiter: :

WebAssist

OK
Cancel
Help

After you create the Populate List From Array behaviors, you'll need to copy the JavaScript in the onChange event for the List/Menus and set them on the onLoad event of the base template, Main.dwt.* (where * stands for the extension that your server model uses for its pages)

```

455 <td align="center"><select name="select" class="findList" onChange=
456 <option selected value="">Select Your Gender</option>

```

Paste this code from both of the onChange events into the onLoad event of the template file Main.dwt.* (where * stands for the extension that your server model uses for its pages)

```

86 </head>
87 <body onLoad="WA_FilterAndPopulateSubList(rsShoeType_WA,MM_findObj('select'),MM_findObj('select2'),1,0,false,': ')
  ':WA_FilterAndPopulateSubList(rsProdSelect_WA,MM_findObj('select2'),MM_findObj('select3'),1,0,false,': ');
  'MM_preloadImages('../images/navigation/nav_shoeshop_roll.gif','../images/navigation/nav_orderhistory_roll.gif','

```

You may find that when you test your application, an error appears, indicating that a function already exists. The Populate List From Array JavaScript in PHP will be applied 2 times, so you may need to remove the following function from the second instance of that code block:

PHP:

```

function WA_DD_Replace($startStr) {
    $startStr = str_replace("'", "|WA|", $startStr);
    $startStr = str_replace("\\", "\\\\", $startStr);
    $startStr = preg_replace("/[\r\n]{1,}/", " ", $startStr);
    return $startStr;
}

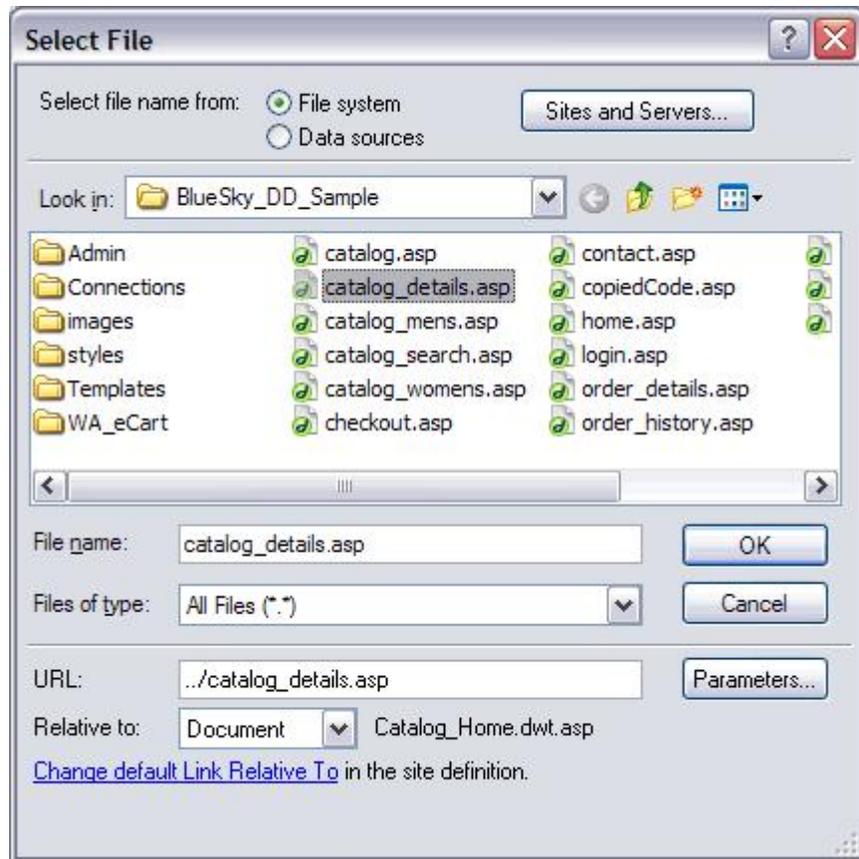
```

Step 5: Implementing the submit on the onChange event:

The last thing you need to do on the Catalog_Home.dwt file is to add a Call to JavaScript behavior to the third List/Menu using the following JavaScript code:

```
document.shoeSearchDD.submit();
```

to submit your form when an item in the third select list is selected. The form Action should be set to submit to the catalog_details page.



Select the third List/Menu from the Dynamic Dropdowns search form and then click on the Add (+) button in the Behaviors panel (not the Server Behaviors panel). Choose Call JavaScript from the list of available behaviors and in the dialog, paste the code shown above:



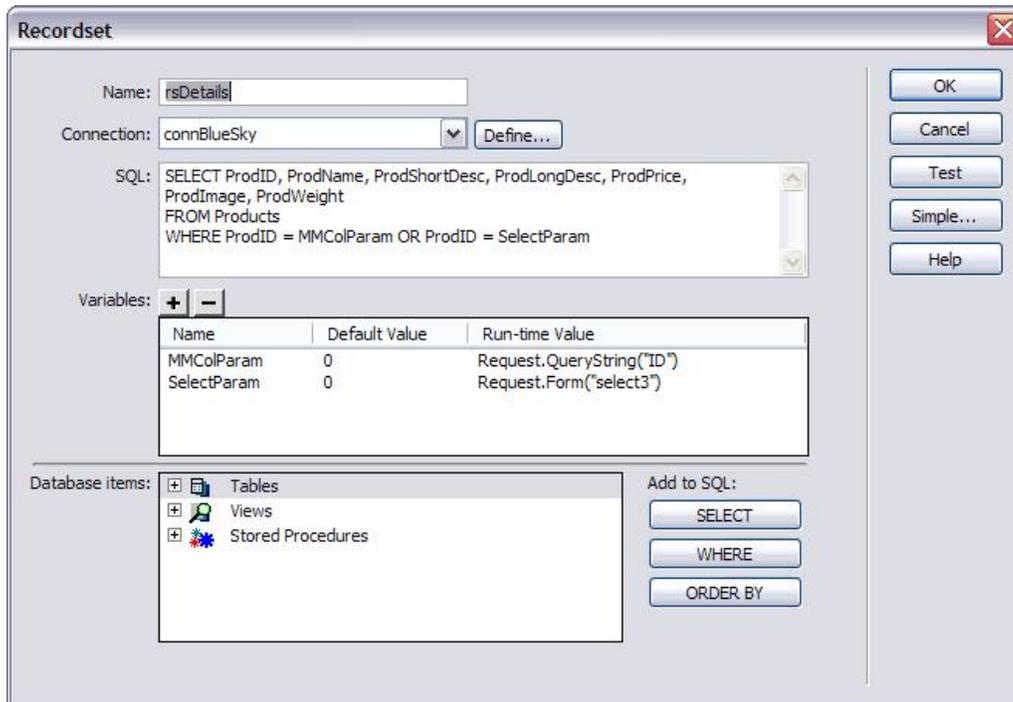
Step 6: Adding the code to the recordset to filter the search:

Lastly you'll need to add a filter on the recordset that will display the selected product based on the ID passed from the select3 form element.

Open the Catalog_Details page for your server language and open the Bindings panel. Double-click on the recordset, rsDetails which is already on the page to open it's interface. Add the following to the WHERE clause:

OR ProdID = SelectParam

Then in the Variables section, change the Default Value for the first parameter, MMColParam to 0 and then click the Add (+) button and enter the Name SelectParam, Default Value 0 and Run-time Value Request.Form("select3") to enable the recordset to filter on the value passed from the third List/Menu in your Dynamic Dropdowns. Setting the Default Value to 0 will not affect the view links from the catalog_mens and catalog_womens pages.



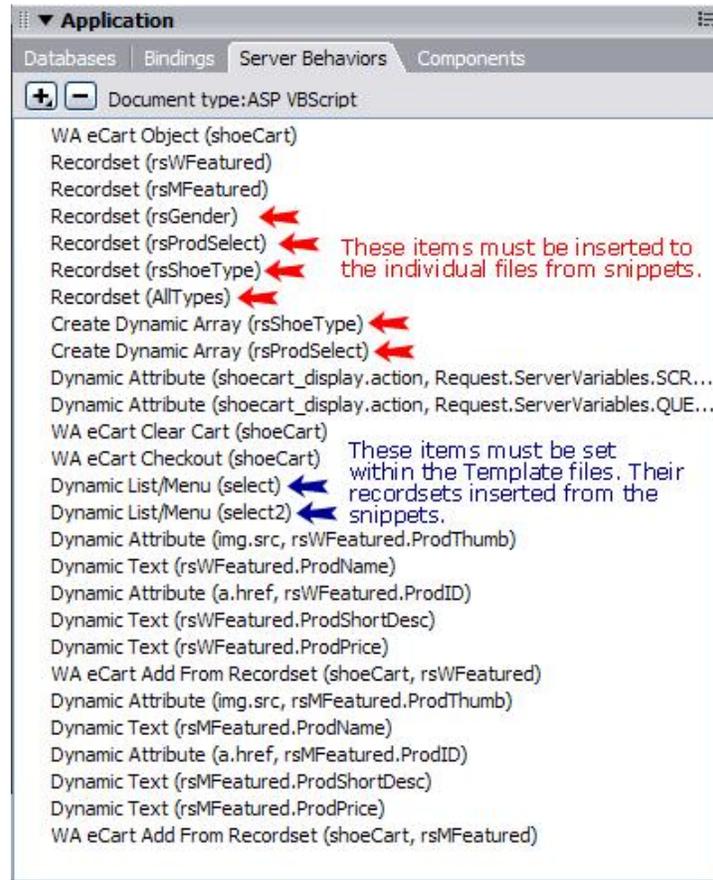
Step 7: Implementing throughout the site:

Because the Dynamic Dropdowns are implemented separately in seven Template files, you'll need to go into the various template files that use this feature and add the code inside the HTML that will enable the feature for each of these Templates. Fortunately we've saved the recordsets and Server Behavior code in snippets, so you should be able to quickly apply the necessary code for each template file. Here is a list of the template files:

- Catalog.dwt.(your server language extension)
- Catalog_Details.dwt.(your server language extension)
- Catalog_Home.dwt.(your server language extension)
- Catalog_Mens.dwt.(your server language extension)
- Catalog_Search.dwt.(your server language extension)
- Catalog_Womens.dwt.(your server language extension)
- Login.dwt.(your server language extension)

Once you've updated the Template files, you'll need to open each of the pages that those Templates are applied to and add the sequence of Server Behaviors

and code chunks needed to enable the Dynamic Dropdowns for each page. Here's what you should see in the catalog page, for instance:



Note: If you don't use snippets, you can copy Server Behaviors from the Template file's Server Behaviors panel by right-clicking (Control Click on Macintosh) on the Server Behavior and choosing Copy from the contextual menu that appears. Then in the individual pages or other Template pages, right-click (Control-click on Macintosh) inside the Server Behaviors panel and choose Paste from the contextual menu.

Still need help?

If these instructions do not allow you to use the extension, log in to our site and click the Support button. On the Support Overview page, choose Support > Premium Support > Submit Incident. We will reply back to you within four business hours. If you need to purchase an incident, choose Support > Premium Support > Purchase. Incidents you purchase do not expire and can be used for any WebAssist products you own, so your support incidents can be a valuable tool when you need help. Please go to:

<http://www.webassist.com/Support/AboutPremium.asp>

for a full description of our Premium Support Service.