Getting Started with Dynamic Dropdowns

Dynamic Dropdowns allows you to create dynamic lists/menus that work together to provide advanced functionality to your site. A few common examples are described at the end of this document to help you get an idea of how you can apply this to your site.

This Getting Started Guide will show you how to have two menu lists (parent and child) that then are used to add specific selections to a target list. This also used Dynamic Dropdowns behaviors to create Add to List, Remove, Select All, and Move Up or Down buttons to your page.

What Do You Need to Start?

- Dreamweaver 8, CS3, or CS4

- Dynamic Dropdowns

- Dynamic page (PHP, ASP-JS, ASP-VB, ColdFusion)

- Compatible database

Configuring your database

Before you begin using Dynamic Dropdowns, you must ensure that your database is configured properly to display the data you need. The following details what you should have set up in your database for this to work.

1. Once you have decided on how you will implement Dynamic Dropdowns, you will need the database tables to support this.

- Parent table: One table will be required as the first menu list.

- Child table: Another table will be required as the Child table. *Depending on your needs, your database may need more than two tables.*

Example: If this were an automotive web site, the Parent table would contain all the manufacturers in the content (text) column. The Child table would contain the various models with the manufacturer ID from the Parent table.

2. The Parent table only requires two columns, one for the ID, another for content.
More columns are permitted, but not required.

3. The Child table must include an ID column, a Secondary key to the Parent table's ID, and one for the content.
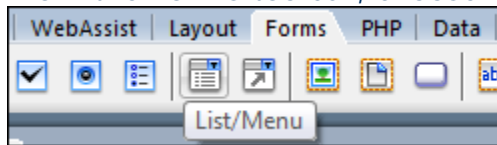
Note: More details regarding configuring your database can be found in the Context-sensitive help documentation built into the product.

Creating your form lists

With your database configured, we can get started with adding the menu/lists to the page. For this example, we will create a menu list for the parent table, another for the child table, and a final list which we will add our selections to.

Creating the Parent List

4. Place your cursor where you wish to insert the Parent list.

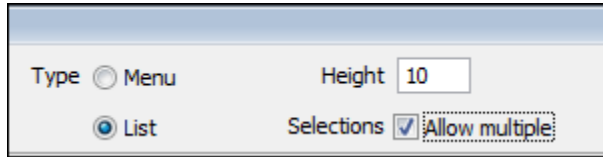5. From the Forms toolbar, choose List/Menu.

6. In the ID field enter Parent.

7. In the Label field enter Parent.
   The names for the IDs and Labels are for this example only.

8. Click **OK**.

9. If asked to *Add form tag?* click **Yes**.

10.   Select your new list.

11.   In the Properties Inspector, choose the List radio button

12.   Set the Height to 10.

Creating the Child List

1. From the Forms toolbar, choose the List/Menu option again.

2. Enter Child in the ID field.

3. Enter Child in the Label field.

4. Click OK.

5. Select the Child list that was just created.

6. In the Properties Inspector, choose the **List** radio button.

7. In the Height field, enter **10**.

8. Choose the **Allow multiple checkbox** to allow you to select multiple items from this list.

Creating the Target List

The target list is not necessary for all applications of Dynamic Dropdowns. For this example, we will create the target list to show implementing the various buttons.

1. From the Forms toolbar, choose **Menu/List**.

2. Enter **Target** in the ID field.

3. Enter **Target** in the Label field.
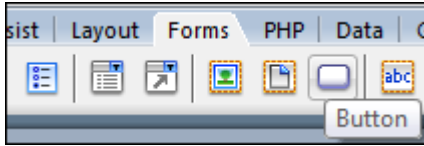
4. Click **OK**.

Creating the buttons

Dynamic Dropdowns can apply various behaviors to form buttons to add some useful functionality to your lists. The following behaviors can be applied to buttons and will be covered in this guide.

- **Add to List**: Adds selections from a list/menu to a target list.

- **Remove**: Removes the selected item(s) in a list.

- **Select All**: Selects all the items that are in the list.

- **Move Up**: Moves the selection above other items in the list.

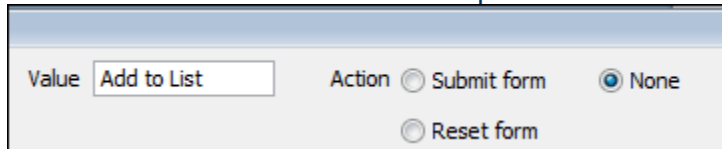- **Move Down**: Moves the selection below other items in the list.

This step shows creating each button using Dreamweaver's built-in functionality. The subsequent steps will show how to use Dynamic Dropdowns to make these buttons active.

Add to List Button

From the Forms toolbar, choose **Button**.

1. In the ID field, enter **add**.

2. Leave the Label field blank, and click **OK**.

3. Select your new button.

4. In the Properties Inspector, replace Submit in the Value field with **Add to List**.

5. Choose **None** from the Action options.

Remove Button

1. Choose the Button option from the Forms toolbar.

2. In the ID field, enter Remove.

3. Leave the Label field blank, and click **OK**.

4. Select the button that was just created.

5. In the Properties Inspector, replace Submit in the Value field with **Remove**.

6. Choose **None** from the Action options.

Select All Button

1. Choose the Button option from the Forms toolbar.

2. In the ID field, enter **Select All**.

3. Leave the Label field blank, and click **OK**.

4. Select the button that was just created.

5. In the Properties Inspector, replace Submit in the Value field with **Select All**.

6. Choose **None** from the Action options.

Move Up Button

1. Choose the Button option from the Forms toolbar.

2. In the ID field, enter **Move Up**.

3. Leave the Label field blank, and click **OK**.

4. Select the button that was just created.

5. In the Properties Inspector, replace Submit in the Value field with **Move Up**.

6. Choose **None** from the Action options.

Move Down Button

1. Choose the Button option from the Forms toolbar.

2. In the ID field, enter **Move Down**.

3. Leave the Label field blank, and click **OK**.

4. Select the button that was just created.

5. In the Properties Inspector, replace Submit in the Value field with **Move Down**.

6. Choose **None** from the Action options.

Creating the Recordsets

Your parent and child lists will both require recordsets before you apply the Dynamic Dropdowns behaviors.
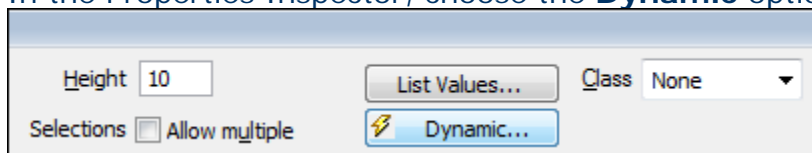
1. From the Server Behaviors panel, choose the **Add** icon.

2. Choose **Recordset**.

3. In the Name field, enter **Parent** (rsParent for ASP users).

4. Choose your database connection from the Connection list.
    If you have not created a database connection, click Define.

5. Choose the parent table from the Table menu list.

6. Click **OK**.

7. Choose the **Add** icon from the Server Behaviors panel, again.

8. In the Name field, enter **Child**.

9. Choose your database connection.

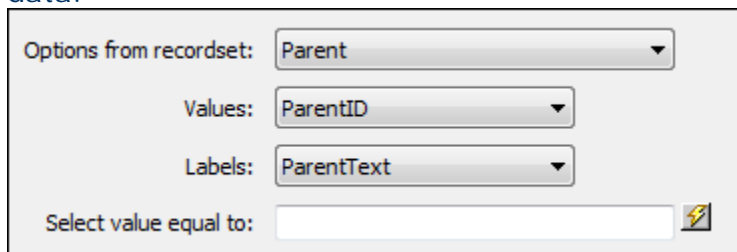10. Select the child table from the Table menu list.

11. Click **OK**.

Creating a Dynamic List

Next, we must turn the Parent list we created earlier, into a dynamic list. As a dynamic list, we can have the Parent list populated with the correct data from the database.

1. Select the parent list on your page.

2. In the Properties Inspector, choose the **Dynamic** option.

3. In the Options from recordset menu, choose the **Parent** table.

4. From the Values menu, choose the **ID column**.

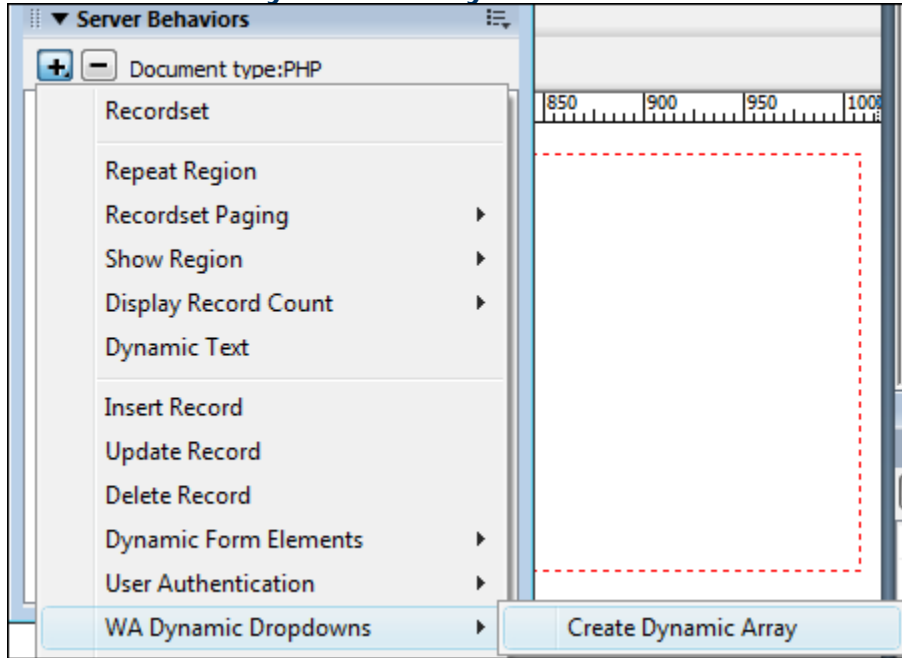5. From the Labels menu, choose the column that contains the text or data.

6. Click **OK**.

Apply Dynamic Dropdowns behaviors

Now that our page contains all the necessary elements for our Dynamic Dropdowns application, we can now start adding the necessary behaviors to these elements.

Creating the Dynamic Array

1. From the Server Behaviors panel, choose the Add icon.

2. Select **WA Dynamic Dropdowns**.
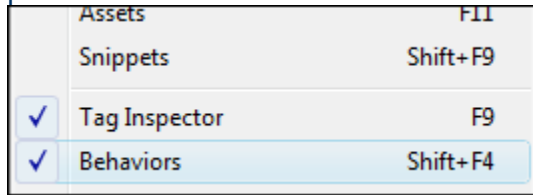
3. Choose **Create Dynamic Array**.

4. From the Recordset menu, choose the **Child table**.

5. From the Parent ID field menu, choose the **ParentID** column.

6. From the Child ID field menu, choose the **ChildID** column.

7. From the Child Text field menu, choose the column that contains the text or data for the child table.
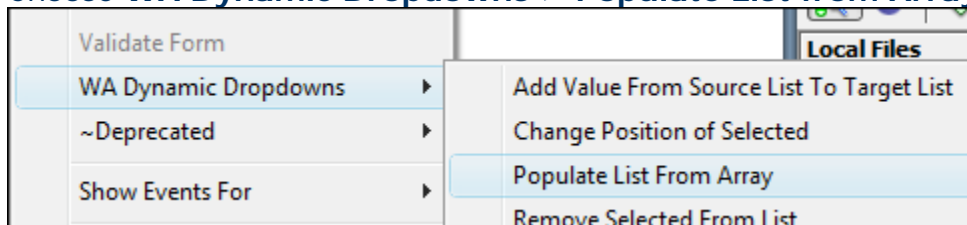
8. Click **OK**.

Populate List from Array

Note: The following steps utilize the Dreamweaver Behaviors panel, which differs from the Server Behaviors panel.

1. From the Window menu, choose Behaviors to open the Behaviors panel.

2. Select the Parent list.

3. Choose **Add** from the Behaviors panel.

4. Choose **WA Dynamic Dropdowns > Populate List from Array**.

5. From the Array source menu, choose the **Child table**.

6. From the Source list, choose the **parent list**.

7. From the Target list, choose the **child list**.

8. Click **OK**.

9. Press **Ctrl (Cmd) + S** to save your page.

Adding Behaviors to Buttons

Add to List Button

1. Select the **Add to List** button on your page.

2. From the Behaviors panel, choose **Add**.

3. Select **WA Dynamic Dropdowns > Add Value From Source List To Target List**.

4. From the source list, choose the **Child** list.

5. From the Target menu list, choose the **Target** list.

6. Click **OK**.

Remove Button

1. Select the **Remove** button on your page.

2. From the Behaviors panel, choose **Add**.

3. Select **WA Dynamic Dropdowns > Remove Selected From List**.

4. From the List menu, choose the **Target** list.

5. Click **OK**.

Select All

1. Select the **Select All** button on your page.

2. From the Behaviors panel, choose Add.

3. Select **WA Dynamic Dropdowns > Select All In List**.

4. From the List menu, choose the **Target** list.

5. Click **OK**.

Move Up

1. Select the **Move Up** button on your page.

2. From the Behaviors panel, choose **Add**.

3. Select **WA Dynamic Dropdowns > Change Position of Selected**.

4. From the List menu, choose the **Target** list.

5. From the Move menu, choose **Up**.

6. Click **OK**.

Move Down

1. Select the **Move Down** button on your page.

2. From the Behaviors panel, choose **Add**.

3. Select **WA Dynamic Dropdowns > Change Position of Selected**.

4. From the List menu, choose the **Target** list.

5. From the Move menu, choose **Down**.

6. Click **OK**.

7. Press **Ctrl (Cmd) + S** to save your page.

8. Press **F12** to preview your dynamic dropdowns and see how they work.

Common Examples

There are a number of applications for a dynamic dropdown, including providing filtering capabilities for an automobile website. This would allow users to choose their vehicle manufacturer in the first menu, which would then populate the second menu list with options related to that maker. This could even include a third dropdown for specifying more specifics regarding the car.

Another common use for dynamic dropdowns is to allow you to easily provide add, remove, select all, options to an existing form menu/list. For example, if you have an email form, which contains a list of all possible email addresses to send to. Using dynamic dropdowns you can specify any or all email addresses to send to, and have those selections added to the To: field of your contact form.