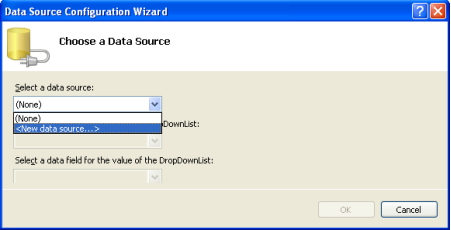
## Master/Detail Filtering With a DropDownList

In this tutorial we'll see how to display the master records in a DropDownList control and the details of the selected list item in a GridView.

## 

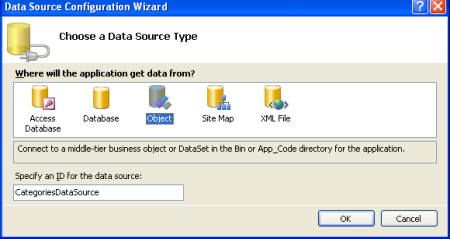
## Step 1: Displaying the Categories in a DropDownList

Open the FilterByDropDownList.aspx page in the Filtering folder, drag on a DropDownList, and set its ID property to Categories. Next, click on the Choose Data Source link from the DropDownList's smart tag. This will display the Data Source Configuration wizard.



**Figure 1**: Specify the DropDownList's Data Source

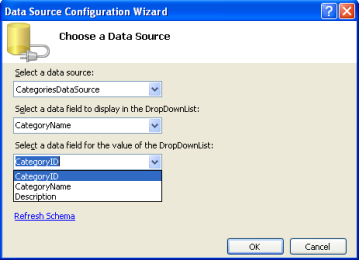
Choose to add a new ObjectDataSource named CategoriesDataSource that invokes the CategoriesBLL class's GetCategories() method.



**Figure 2**: Add a New ObjectDataSource Named CategoriesDataSource

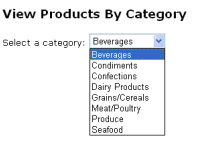
|  |  |
| --- | --- |
| Choose a Business Object: CategoriesBLL | Select method GetCategories() |
|  |  |

After configuring the ObjectDataSource we still need to specify what data source field should be displayed in DropDownList and which one should be associated as the value for the list item. Have the CategoryName field as the display and CategoryID as the value for each list item.



**Figure 5**: Have the DropDownList Display the CategoryName Field and Use CategoryID as the Value

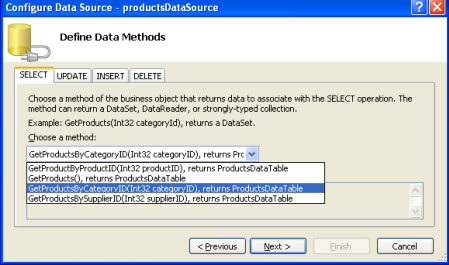
At this point we have a DropDownList control that's populated with the records from the Categories table. Figure 6 shows our progress thus far when viewed through a browser.



**Figure 6**: A Drop-Down Lists the Current Categories

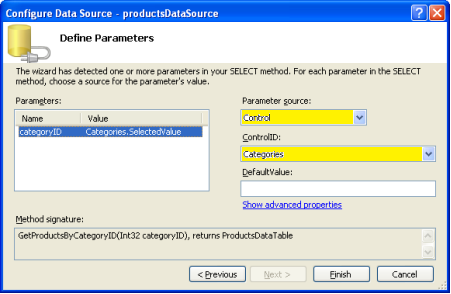
## Step 2: Adding the Products GridView

Add a GridView to the page and create a new ObjectDataSource named **productsDataSource.** Have the **productsDataSource** control call its data from the **ProductsBLL** class's **GetProductsByCategoryID(categoryID)** method.



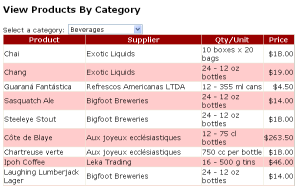
**Figure 7**: Select the GetProductsByCategoryID(categoryID) Method

After choosing this method, the ObjectDataSource wizard prompts us for the value for the method's *categoryID* parameter. To use the value of the selected categories DropDownList item set the Parameter source to Control and the ControlID to Categories.



**Figure 8**: Set the ***categoryID*** Parameter to the Value of the **Categories** DropDownList

When first visiting the page, those products belong to the selected category (Beverages) are displayed (as shown in Figure 9), but changing the DropDownList doesn't update the data. This is because a postback must occur for the GridView to update. To accomplish this **Set the categories DropDownList's** [AutoPostBack property](http://msdn2.microsoft.com/en-US/library/system.web.ui.webcontrols.listcontrol.autopostback%28VS.80%29.aspx) **to True.** This will trigger a postback whenever the DropDownList's selected item is changed by the user.

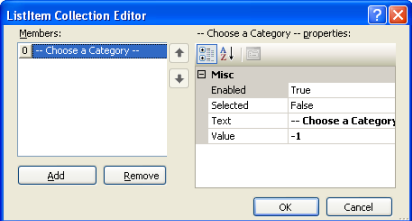


**Figure 9**: When First Visiting the Page, the Beverage Products are Displayed

## Adding a "-- Choose a Category --" List Item

Rather than showing the first category's products, we may want to instead have a DropDownList item selected that says something like, "-- Choose a Category --".

To add a new list item to the DropDownList, go to the Properties window and click on the ellipses in the **Items** property. Add a new list item with the Text "-- Choose a Category --" and the Value -1.



**Figure 11**: Add a -- Choose a Category -- List Item

Alternatively, you can add the list item by adding the following markup to the DropDownList:

<asp:DropDownList ID="categories" runat="server" AutoPostBack="True"

DataSourceID="categoriesDataSource"

DataTextField="**CategoryName**"

DataValueField="**CategoryID**"

EnableViewState="False"

**AppendDataBoundItem="True">**

**<asp:ListItem Value="-1"> -- Choose a Category – </asp:ListItem>**

</asp:DropDownList>

Additionally, we need to set the DropDownList control's **AppendDataBoundItems** to True because when the categories are bound to the DropDownList from the ObjectDataSource they'll overwrite any manually-added list items if AppendDataBoundItems isn't True.

After these changes, when first visiting the page the "-- Choose a Category --" option is selected and no products are displayed.

**The reason no products are displayed when because the "-- Choose a Category --" list item is selected is because its value is -1 and there are no products in the database with a CategoryID of -1.**

However, you want to display all of the categories when the "-- Choose a Category --" list item is selected, return to the ProductsBLL class and customize the GetProductsByCategoryID(categoryID) method so that it invokes the GetProducts() method if the passed in *categoryID* parameter is less than zero:

public Northwind.ProductsDataTable GetProductsByCategoryID(int categoryID)

{

if (categoryID < 0)

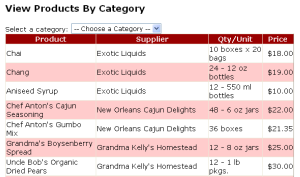
return GetProducts();

else

return Adapter.GetProductsByCategoryID(categoryID);

}

Figure 14 shows a screen shot of FilterByDropDownList.aspx when the "-- Choose a Category --" option is selected. Here, all of the products are displayed by default, and the user can narrow the display by choosing a specific category.



**Figure 14**: All of the Products are Now Listed By Default