This tutorial is part of a set. Find out more about data access with ASP.NET in the Working with Data in ASP.NET 2.0 section of the ASP.NET site at http://www.asp.net/learn/dataaccess/default.aspx.

Working with Data in ASP.NET 2.0 :: Master/Detail Using a Selectable Master GridView with a Details DetailView

Introduction

In the <u>previous tutorial</u> we saw how to create a master/detail report using two web pages: a "master" web page, from which we displayed the list of suppliers; and a "details" web page that listed those products provided by the selected supplier. This two page report format can be condensed into one page. This tutorial will have a GridView whose rows include the name and price of each product along with a Select button. Clicking the Select button for a particular product will cause its full details to be displayed in a DetailsView control on the same page.

Dutitled Page - Microsoft Internet	Explorer	
Ele Edit View Favorites Iools	Help	10
🗿 Back 🔹 🔘 🔺 📓 🐔 🔎	Search 🍁 Favorites 🙆 🎰 🖗 🕶 💭	× 🕰 🕸
Address 🕘 http://localhost:1563/Code/Filt	ering/DetailsBySelecting_aspx	💌 🛃 Go
Working with Da	ata Tutorials Home > Filter Details of Sele	ing Reports > acted Row
Home	View Product Details	
Basic Reporting	Deaduct Chef Antop's Gumba N	Aise
Simple Display	Category Condiments	11.4
Declarative	Supplier New Orleans Catun Del	lights
Parameters	Qty/Unit 36 boxes	
Satting Parameter	Price \$21.35	
Values	Units In Stock 0	
Filtering Reports	Units On Order 0	
Hitening Reports	Reorder Level 0	
Filter by Drop-Down	Discontinued	
	Product	Linit Drice
Master-Details-	Select Chai	\$18.00
Chi comp	Select Chang	\$19.00
Two Pages	Select Aniseed Syrup	\$10.00
Two Pages	Select Chef Anton's Cajun Seasoning	\$22.00
Details of Selected	Select Chef Anton's Gumbo Mix	\$21.35
NOW.	Select Grandma's Boysenberry Spread	\$25.00
Customized	Select Linde Bob's Organic Dried Pears	\$30.00
e)		S Local Intranet

Figure 1: Clicking the Select Button Displays the Product's Details

Step 1: Creating a Selectable GridView

Recall that in the two-page master/detail report that each master record included a hyperlink that, when clicked, sent the user to the details page passing the clicked row's SupplierID value in the querystring. Such a

hyperlink was added to each GridView row using a HyperLinkField. For the single page master/details report, we will need a Button for each GridView row that, when clicked, shows the details. The GridView control can be configured to include a Select button for each row that causes a postback and marks that row as the GridView's SelectedRow.

Start by adding a GridView control to the DetailsBySelecting.aspx page in the Filtering folder, setting its ID property to ProductsGrid. Next, add a new ObjectDataSource named AllProductsDataSource that invokes the ProductsBLL class's GetProducts() method.

Data Source C	onfiguration	Wizard			? 🛛
	ioose a Data	Source Ty	pe		
Where will the Access Database	Database	get data from	n?	XML File	
Connect to a r	niddle-tier busine	ess object or Da	staSet in the Bin	or App_Code directory	for the application.
Specify an ID f	or the data sour	ce:			
AlProductsDat	aSource				
					OK Cancel

Figure 2: Create an ObjectDataSource Named AllProductsDataSource



Figure 3: Use the ProductsBLL Class

Configure	e Data Source - allProductsDataSource	X
Ļ	Define Data Methods	
SELECT	UPDATE INSERT DELETE	
Choos metho Examp Choos	se a method of the business object that returns data to associate with the SELECT operation. The id can return a DataSet, DataReader, or strongly-typed collection. ple: GetProducts(Int32 categoryId), returns a DataSet. se a method:	
GetPr	roducts(), returns ProductsDataTable	
GetPro GetPro GetPro	oduct8yProductID(Int32 productID), returns ProductsDataTable oducts0, returns ProductsDataTable oducts8yCategoryID(Int32 categoryID), returns ProductsDataTable oducts8ySupplierID(Int32 supplierID), returns ProductsDataTable	
	< Previous Next > Einish Cancel	

Figure 4: Configure the ObjectDataSource to Invoke the GetProducts () Method

Edit the GridView's fields removing all but the ProductName and UnitPrice BoundFields. Also, feel free to customize these BoundFields as needed, such as formatting the UnitPrice BoundField as a currency and changing the HeaderText properties of the BoundFields. These steps can be accomplished graphically, by clicking the Edit Columns link from the GridView's smart tag, or by manually configuring the declarative syntax.



Figure 5: Remove All But the ProductName and UnitPrice BoundFields

The final markup for the GridView is:

```
<asp:GridView ID="ProductsGrid" runat="server"
AutoGenerateColumns="False" DataKeyNames="ProductID"
DataSourceID="AllProductsDataSource" EnableViewState="False">
<Columns>
<asp:BoundField DataField="ProductName"
HeaderText="Product" SortExpression="ProductName" />
<asp:BoundField DataField="UnitPrice"
DataFormatString="{0:c}" HeaderText="Unit Price"
HtmlEncode="False" SortExpression="UnitPrice" />
</Columns>
```

Next, we need to mark the GridView as selectable, which will add a Select button to each row. To accomplish this, simply check the Enable Selection checkbox in the GridView's smart tag.



Figure 6: Make the GridView's Rows Selectable

Checking the Enabling Selection option adds a CommandField to the ProductsGrid GridView with its showSelectButton property set to True. This results in a Select button for each row of the GridView, as Figure 6 illustrates. By default, the Select buttons are rendered as LinkButtons, but you can use Buttons or ImageButtons instead through the CommandField's ButtonType property.

```
<asp:GridView ID="ProductsGrid" runat="server"
AutoGenerateColumns="False" DataKeyNames="ProductID"
DataSourceID="AllProductsDataSource" EnableViewState="False">
<Columns>
<asp:CommandField ShowSelectButton="True" />
<asp:BoundField DataField="ProductName"
HeaderText="Product" SortExpression="ProductName" />
<asp:BoundField DataField="UnitPrice"
DataFormatString="{0:c}" HeaderText="Unit Price"
HtmlEncode="False" SortExpression="UnitPrice" />
</Columns>
</asp:GridView>
```

When a GridView row's Select button is clicked a postback ensues and the GridView's SelectedRow property is updated. In addition to the SelectedRow property, the GridView provides the <u>SelectedIndex</u>, <u>SelectedValue</u>, and <u>SelectedDataKey</u> properties. The SelectedIndex property returns the index of the selected row, whereas the SelectedValue and SelectedDataKey properties return values based upon the GridView's <u>DataKeyNames property</u>.

The DataKeyNames property is used to associate one or more data field values with each row and is commonly used to attribute uniquely identifying information from the underlying data with each GridView row. The SelectedValue property returns the value of the first DataKeyNames data field for the selected row where as the SelectedDataKey property returns the selected row's DataKey object, which contains all of the values for the specified data key fields for that row.

The DataKeyNames property is automatically set to the uniquely-identifying data field(s) when you bind a data source to a GridView, DetailsView, or FormView through the Designer. While this property has been set for us automatically in the preceding tutorials, the examples would have worked without the DataKeyNames property specified. However, for the selectable GridView in this tutorial, as well as for future tutorials in which we'll be examining inserting, updating, and deleting, the DataKeyNames property must be set properly. Take a moment to ensure that your GridView's DataKeyNames property is set to ProductID.

Let's view our progress thus far through a browser. Note that the GridView lists the name and price for all of the products along with a Select LinkButton. Clicking the Select button causes a postback. In Step 2 we'll see how to have a DetailsView respond to this postback by displaying the details for the selected product.

JUntitled Page - Microsoft Interne	t Explorer	
Ele Edit View Favorites Icols	Fielb	
G Fach + Q + 🖻 🖹 🐔 🔎	Search 👷 Favorites 🙆 🎰 💀 🗾 🔹 🧶	* 長 穏
Address Address Address Address Address Address	tering/DetailsBySelecting.aspx	👻 🛃 GO
Working with Da	ata Tutorials Home > Filter Details of Sel	ing Reports > ected Row
Basic Reporting	Droduct	Unit Price
Simple Display	Select Chai	\$18.00
Declarative	Select Chang	\$19.00
Parameters	Select Aniseed Syrup	\$10.00
Setting Parameter	Select Chef Anton's Cajun Seasoning	\$22.00
Values	Select Chef Anton's Gumbo Mix	\$21.35
	Select Grandma's Boysenberry Spread	\$25.00
Filtering Reports	Select Unde Bob's Organic Dried Pears	\$30.00
Filter by Drop-Down	Select Northwoods Cranberry Sauce	\$40.00
List	Select Mishi Kobe Niku	\$97.00
Master-Details-	Select Ikura	\$31.00
	Calant Ouisen Cahralae	421.00
E Done		Local intranet

Figure 7: Each Product Row Contains a Select LinkButton

Highlighting the Selected Row

The ProductsGrid GridView has a SelectedRowStyle property that can be used to dictate the visual style for the selected row. Used properly, this can improve the user's experience by more clearly showing which row of the GridView is currently selected. For this tutorial, let's have the selected row be highlighted with a yellow background.

As with our earlier tutorials, let's strive to keep the aesthetic-related settings defined as CSS classes. Therefore, create a new CSS class in Styles.css named SelectedRowStyle.

```
.SelectedRowStyle
{
    background-color: Yellow;
}
```

To apply this CSS class to the SelectedRowStyle property of *all* GridViews in our tutorial series, edit the GridView.skin Skin in the DataWebControls Theme to include the SelectedRowStyle settings as shown below:

With this addition, the selected GridView row is now highlighted with a yellow background color.



Figure 8: Customize the Selected Row's Appearance Using the GridView's SelectedRowStyle Property

Step 2: Displaying the Selected Product's Details in a DetailsView

With the ProductsGrid GridView complete, all that remains is to add a DetailsView that displays information about the particular product selected. Add a DetailsView control above the GridView and create a new ObjectDataSource named ProductDetailsDataSource. Since we want this DetailsView to display particular information about the selected product, configure the ProductDetailsDataSource to use the ProductsBLL class's GetProductByProductID (*productID*) method.

Configure	Data So	urce - p	roductDeta	ailsDataSour	e			?	×
þ	Define	Data M	lethods						
SELECT	UPDATE	INSERT	DELETE						
Choos metho Examp Ghoos	e a method d can retur xie: GetPro e a method	f of the bu n a Datas ducts(Int3 f:	isiness object iet, DataRead i2 category1d	t that returns da der, or strongly- I), returns a Dat	ta to associat typed collectii aSet.	te with the on.	SELECT ope	ration. The	
GetPro	oductByPro	ductID(In	t32 productIC	D), returns Prod	x v				
GetPro GetPro GetPro	oducts(), re oducts(), re oducts()yCa oducts()ySu	ductID(In sturns Pro stegoryID pplierID(I	CI2 productio ductsDataTab (Int32 catego nt32 supplier]	0), returns Prod ble oryID), returns P ID), returns Prod	uctsDataTable roductsDataT luctsDataTab	able le		<	
				< Previous	<u>N</u> ext :	>	Eirish	Cancel]

Figure 9: Invoke the ProductsBLL Class's GetProductByProductID (productID) Method

Have the *productID* parameter's value obtained from the GridView control's SelectedValue property. As we discussed earlier, the GridView's SelectedValue property returns the first data key value for the selected row. Therefore, it's imperative that the GridView's DataKeyNames property is set to ProductID, so that the selected row's ProductID value is returned by SelectedValue.

Configure Data Source - productDetailsDataS	ource	? 🗙
Define Parameters		
The wizard has detected one or more parameters in yo method, choose a source for the parameter's value.	ur SELECT method. For each parameter in the SELEC	т
Paramgters:	Parameter gource:	
Name Value	Control	
productID ProductsGrid,SelectedValue	<u>C</u> ontrolID:	
	ProductsGrid	~
	DefaultValue:	
	Show advanced properties	
Method signature:		
GetProductByProductID(Int32 productID), returns Pro	ductsDataTable	0.0
< Pre-	vious Next > Einish Cance	1

Figure 10: Set the productID Parameter to the GridView's SelectedValue Property

Once the productDetailsDataSource ObjectDataSource has been configured correctly and bound to the DetailsView, this tutorial is complete! When the page is first visited no row is selected, so the GridView's SelectedValue property returns Nothing. Since there are no products with a NULL ProductID value, no

records are returned by the GetProductByProductID (*productID*) method, meaning that the DetailsView isn't displayed (see Figure 11). Upon clicking a GridView row's Select button, a postback ensues and the DetailsView is refreshed. This time the GridView's SelectedValue property returns the ProductID of the selected row, the GetProductByProductID (*productID*) method returns a ProductSDataTable with information about that particular product, and the DetailsView shows these details (see Figure 12).

Districted Page - Microsoft Internet	LExplorer Help Search ☆ Favorites @ 🙆 • 🏭 🗹 • 🗔 🛛 🧔 ering/DetalsBySelecting.aspx) کے ا ایک ایک (س کے ایک (
Working with Da	ata Tutorials Home > Filter Details of Sel	ing Reports > ected Row
Home	View Product Details	
Basic Reporting	Product	Unit Price
Simple Display	Select Chai	\$18.00
Declarative	Select Chang	\$19.00
Parameters	Select Aniseed Syrup	\$10.00
Eather Daramater	Select Chef Anton's Cajun Seasoning	\$22.00
Values	Select Chef Anton's Gumbo Mix	\$21.35
	Select Grandma's Boysenberry Spread	\$25.00
Filtering Reports	Select Unde Bob's Organic Dried Pears	\$30.00
Filter by Drop-Down	Select Northwoods Cranberry Sauce	\$40.00
List	Select Mishi Kobe Niku	\$97.00
Master-Details-	Select Ikura	\$31.00
Details	Select Queso Cabrales	\$21.00
Master Detail Games	Select Queso Manchego La Pastora	\$38.00
Two Pages	SelectKonbu	\$6.00
Thorages	Select Tofu	\$23.25
Details of Selected	Select Genen Shouyu	\$15.50
ROW	Select Pavlova	\$17.45
Customized	Select Alice Mutton	\$39.00
ù		Local inkranek

Figure 11: When First Visited, Only the GridView is Displayed

Untitled Page - Microsoft Internet	Explorer	
Ele Edit View Fgvorites Icols (jelp	1
🗿 Back • 🔘 - 🖹 🖹 🐔 🔎 5	Search 👷 Favorites 🕢 🍰 - 🍇 🖻]·□
Address () http://localhost:1563/Code/Filt	ering/DetailsBySelecting.aspx	🛩 🛃 Go
Working with Da	ata Tutorials	Home > Filtering Reports > Details of Selected Row
Home	View Product Deta	ils
Basic Reporting	Product Chef Ant	on's Gumbo Mix
Simple Display	Category Condimer	nts
Declarative	Supplier New Orle	ans Cajun Delights
Parameters	Qty/Unit 36 boxes	
Setting Parameter	Price \$21.35	
Values	Units In Stock 0	
Elitarioa Banasta	Units On Order 0	
Filtening Reports	Reorder Level 0	
Filter by Drop-Down List	Discontinued	
Master-Details-	Produ	ct Unit Price
Details	<u>Select</u> Chai	\$18.00
Master/Detail Across	Select Chang	\$19.00
Two Pages	Select Aniseed Syrup	\$10.00
Details of Selected	Select Chef Anton's Cajun	Seasoning \$22.00
Row	Select Cher Anton's Gumb	0 Mix \$21.35
Circtomized	Select Upde Robie Occasio	Dried Pears \$20.00
	select time prices Ornanic	Local intranet

Figure 12: Upon Selecting a Row, the Product's Details are Displayed

Summary

In this and the preceding three tutorials we've seen a number of techniques for displaying master/detail reports. In this tutorial we examined using a selectable GridView to house the master records and a DetailsView to display details about the selected master record on the same page. In the earlier tutorials we looked at how to display master/details reports using DropDownLists and displaying master records on one web page and detail records on another.

This tutorial concludes our examination of master/detail reports. Starting with the next tutorial we'll begin our exploration of customized formatting with the GridView, DetailsView, and FormView. We'll see how to customize the appearance of these controls based on the data bound to them, how to summarize data in the GridView's footer, and how to use templates to obtain a greater degree of control over the layout.

Happy Programming!

About the Author

Scott Mitchell, author of six ASP/ASP.NET books and founder of 4GuysFromRolla.com, has been working with Microsoft Web technologies since 1998. Scott works as an independent consultant, trainer, and writer, recently completing his latest book, <u>Sams Teach Yourself ASP.NET 2.0 in 24 Hours</u>. He can be reached at <u>mitchell@4guysfromrolla.com</u> or via his blog, which can be found at <u>http://ScottOnWriting.NET</u>.

Special Thanks To...

This tutorial series was reviewed by many helpful reviewers. Lead reviewer for this tutorial was Hilton Giesenow. Interested in reviewing my upcoming MSDN articles? If so, drop me a line at <u>mitchell@4GuysFromRolla.com</u>.