Getting Started with WebExtenders for ASP.NET

WebExtenders for ASP.NET from TMG Development Ltd extend the capabilities of various Web Server controls which ship with the Microsoft Visual Studio .NET design environment. Each component in the library adds a different behavior to controls when they are rendered by the clients browser.

At designtime, they use Visual Studio's *extender provider* mechanism to add new properties to controls such as Button, LinkButton, Textbox etc.
If you are unfamiliar with this concept, see the following link:
http://msdn.microsoft.com/library/default.asp?url=/library/en-us/vbcon/html/vbconextenderobjects.asp for further information. At runtime, each typically sends a small amount of javascript to the browser which alters the controls behavior accordingly. For example:

- **The ConfirmProvider component adds the *ConfirmText* property to all Button, HyperLink, ImageButton and LinkButton controls on a Web Form. Any control whose ConfirmText property has been set, invokes the javascript *confirm* function in response to the user clicking the control; the specified text is displayed in a modal OK/Cancel dialog box.**
- **The StatusProvider component adds the *StatusText* property to numerous controls including the Web Form itself. Any control whose StatusText property has been set alters the javascript *status* property as the user moves the mouse over the control; the specified text is displayed in the status bar. When the StatusText is set on the Web Form, this determines the *defaultStatus* for the clients browser (that is, the text which appears by default in the status bar).**
- **The AlertProvider component adds the *AlertText* property to button controls, invoking the javascript *alert* function in response to the user clicking the control.**
- **The MaskProvider component adds the *MaskExpression* property to TextBox controls. The mask is applied in response to text entry, with disallowed characters being ignored.**

WebExtenders are easily integrated into exisiting Web Forms without the need to replace controls with more specialized versions. The extra functionality is simply attached at runtime and as we will see, two or more WebExtenders can combine their functionality when used to extend the same Web Server control.

Overview

In this document we will outline the steps necessary to extend both a Web Form (Sytem.Web.UI.Page) and a LinkButton (System.Web.UI.WebControls.LinkButton) Web Server control using the Microsoft Visual Studio .NET IDE, before taking a look at the results when rendered by the clients browser.

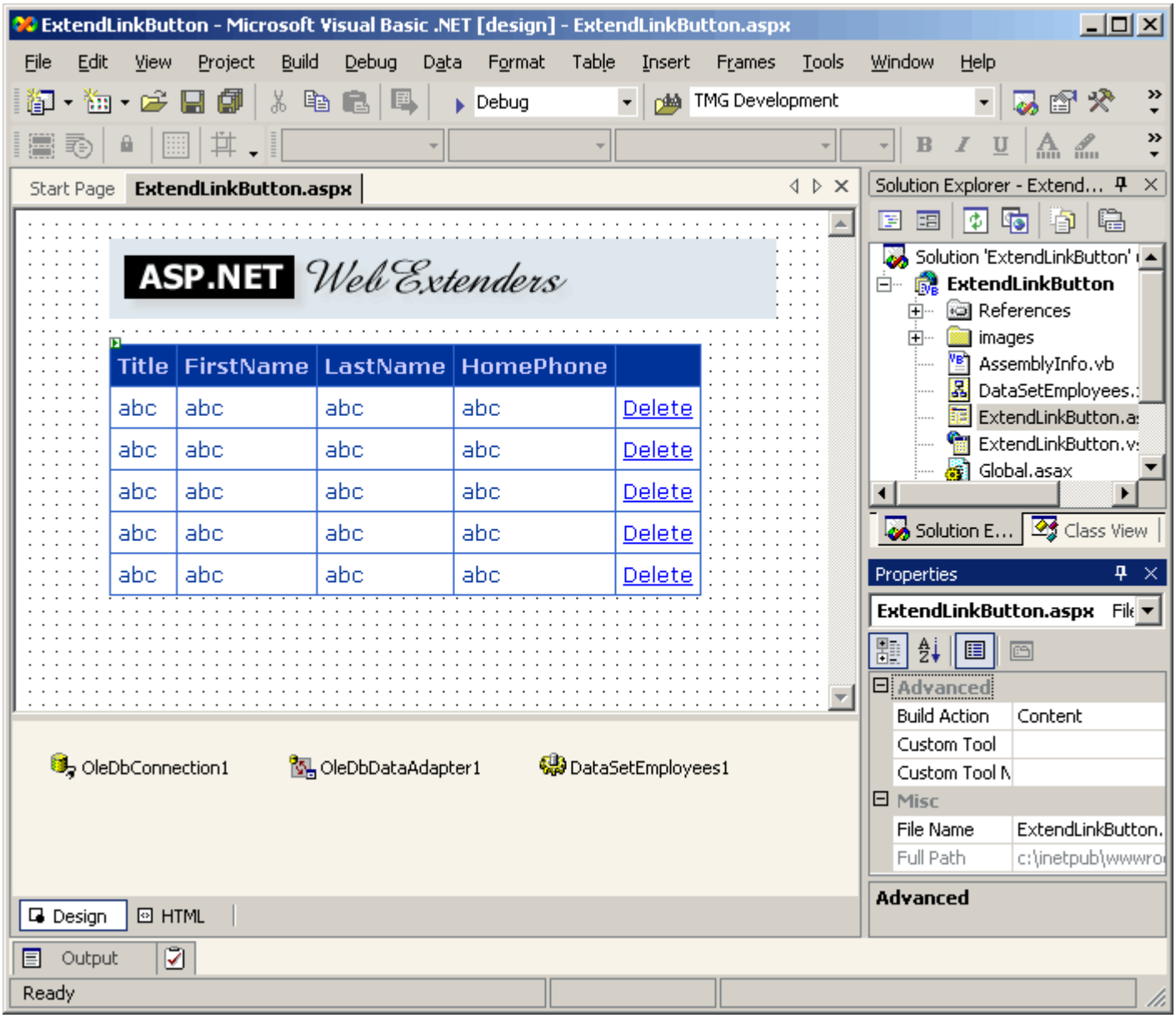
We're going to create a very simple web application that consists of a typical data-bound web page and then show how we add a StatusProvider component, extending the properties of the controls on the form as well as the form itself. Then we will add a ConfirmProvider component, thus extending the LinkButton control with a second new property.

Using WebExtenders at Design Time

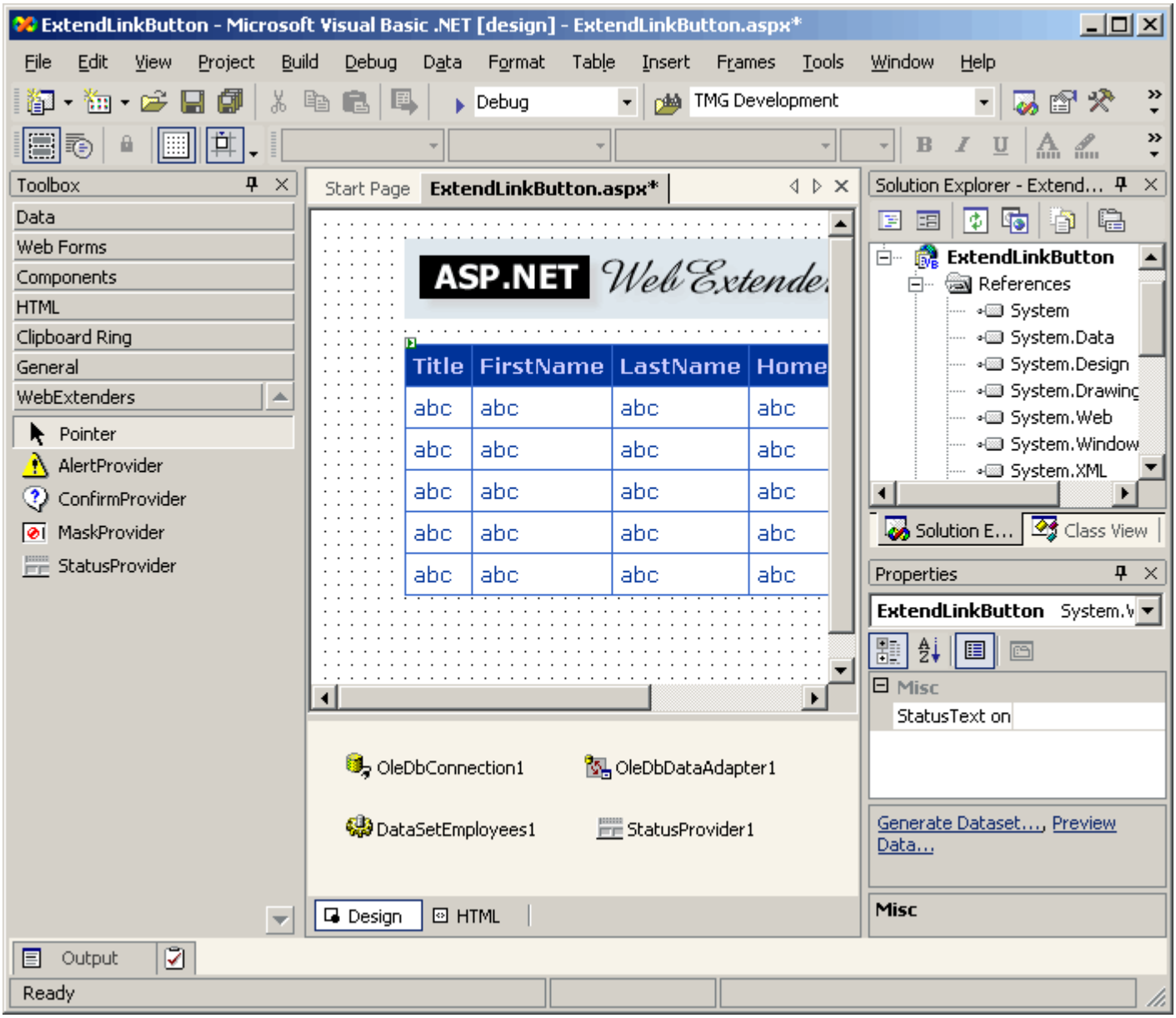
To start with, we've created a basic page based around a DataGrid Web Server control. It has already been configured with a source of data and some basic formatting to improve its appearance. The OleDbConnection, OleDbDataAdapter and typed DataSet components in the component tray will populate the DataGrid at runtime. Their configuration is unrelated to use of the WebExtender components however, so we will not discuss it any further here.

In addition to the four columns which correspond to columns within the DataSet (Title, FirstName, LastName & HomePhone) we have added a 'Delete' button column. We will use the StatusProvider to extend the links in this column to show custom text messages in the clients web-browser at runtime.

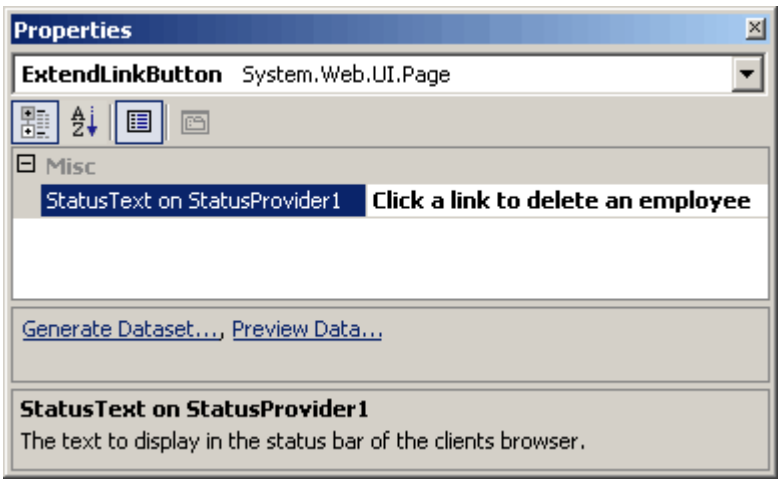
Note that in order to be able to extend the links in this column, the column must be converted to a 'Template Column'.



The first step is to add a StatusProvider component. Dragging it straight onto the form from the Toolbox causes it to appear in the component tray:



Before we customize the links in the DataGrid, lets consider the simpler case of the Web Form (System.Web.UI.Page). StatusProvider extends the form to allow a means of specifying the *defaultStatus* for the browser. Selecting the form from the dropdown list in the Properties Window, reveals:

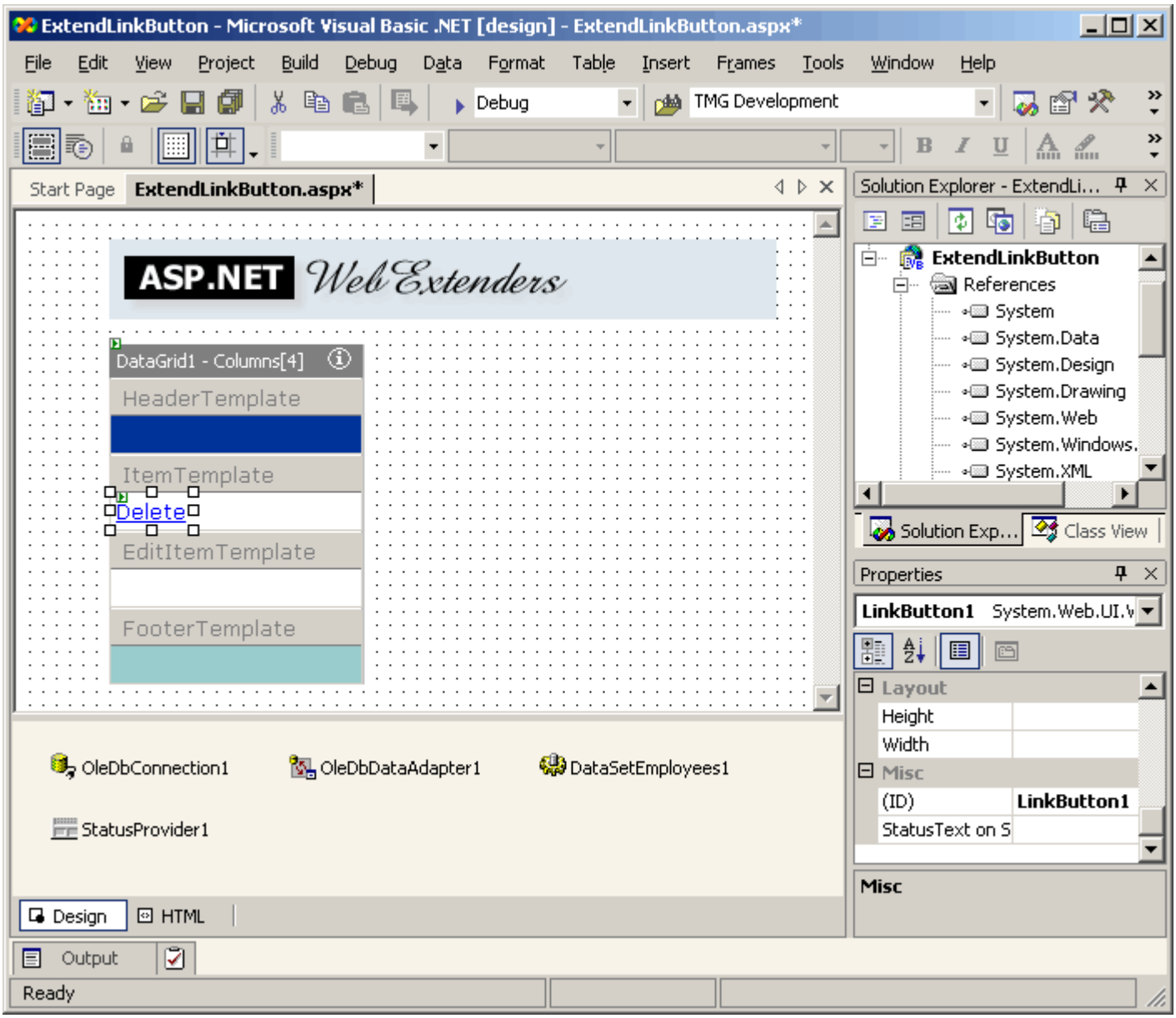


Highlighted in the 'Misc' section, is the StatusText property which has been provided by the StatusProvider component. This is where we can specify the text which will appear in the status bar of the clients browser. Because this is a form, the text will appear

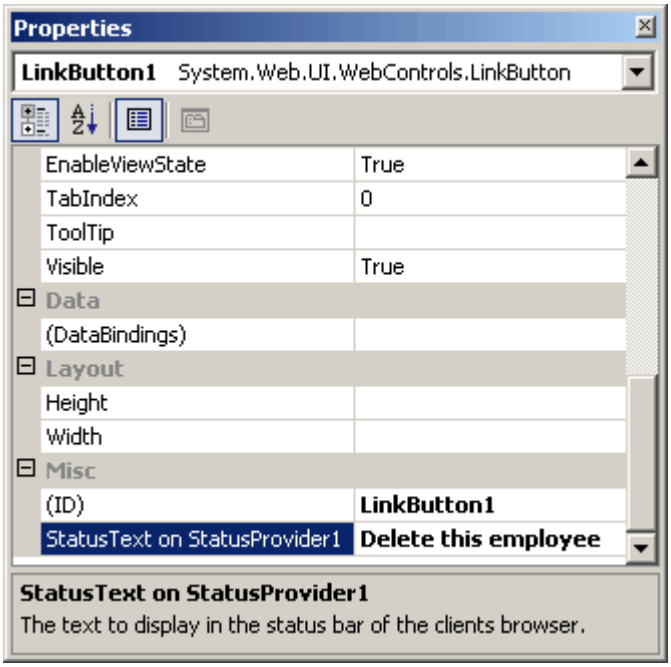
until **the user drives over a link in the browser.**

The StatusProvider component extends any other link controls on the form in the same way (although of course their text would not appear until the user drives over them).

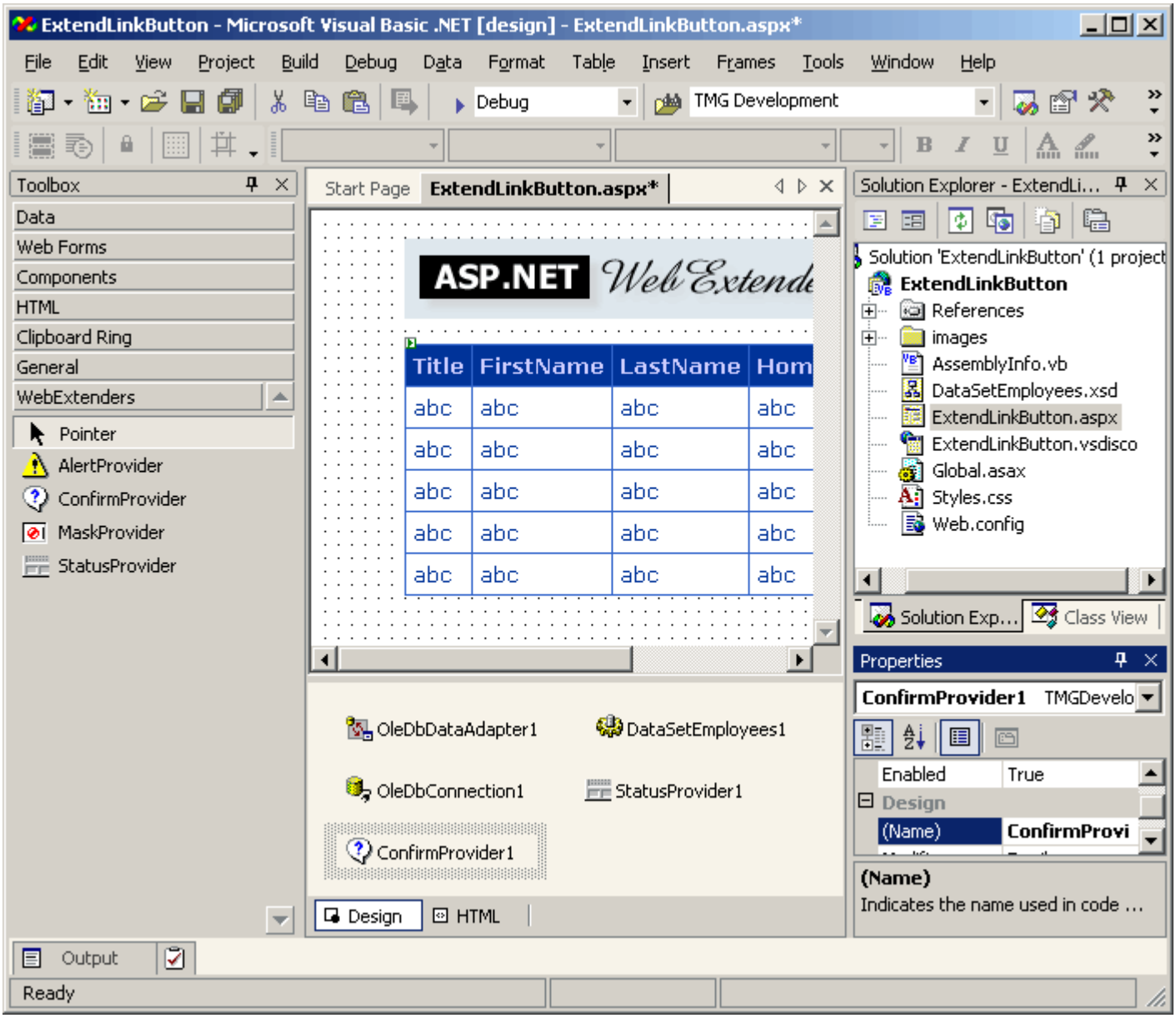
Before we can specify the status bar text for a control which is hosted by the DataGrid control, we need to enter *template editing mode*. Right-clicking anywhere on the surface of the DataGrid and selecting 'Edit Template' >> 'Columns[X]' from the popup menu causes the DataGrid to alter its appearance, as shown below:



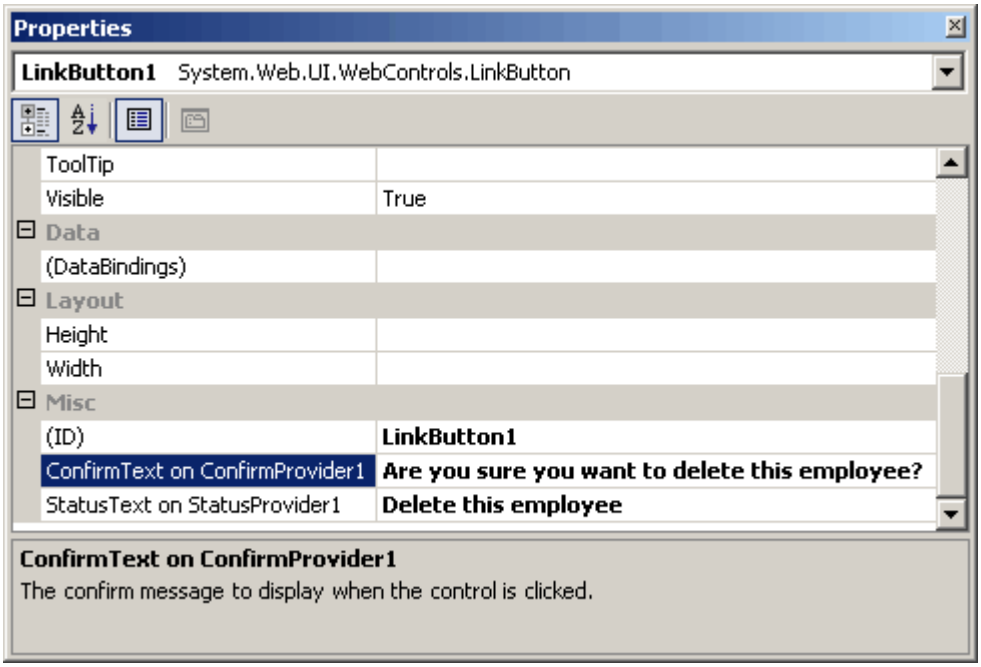
Notice that the Web Form designer has created a LinkButton control so that we may edit its properties. As before, the Properties Window shows the new property which has been supplied by the StatusProvider:



In this example, *Delete this employee* would seem to be an appropriate message. Having entered it into the property window, we'll conclude our template editing session: right-clicking the DataGrid once more and selecting 'End Template Editing' returns the DataGrid to its original state. Let's now add a ConfirmProvider component from the Toolbox. Dragging it onto the form as before:



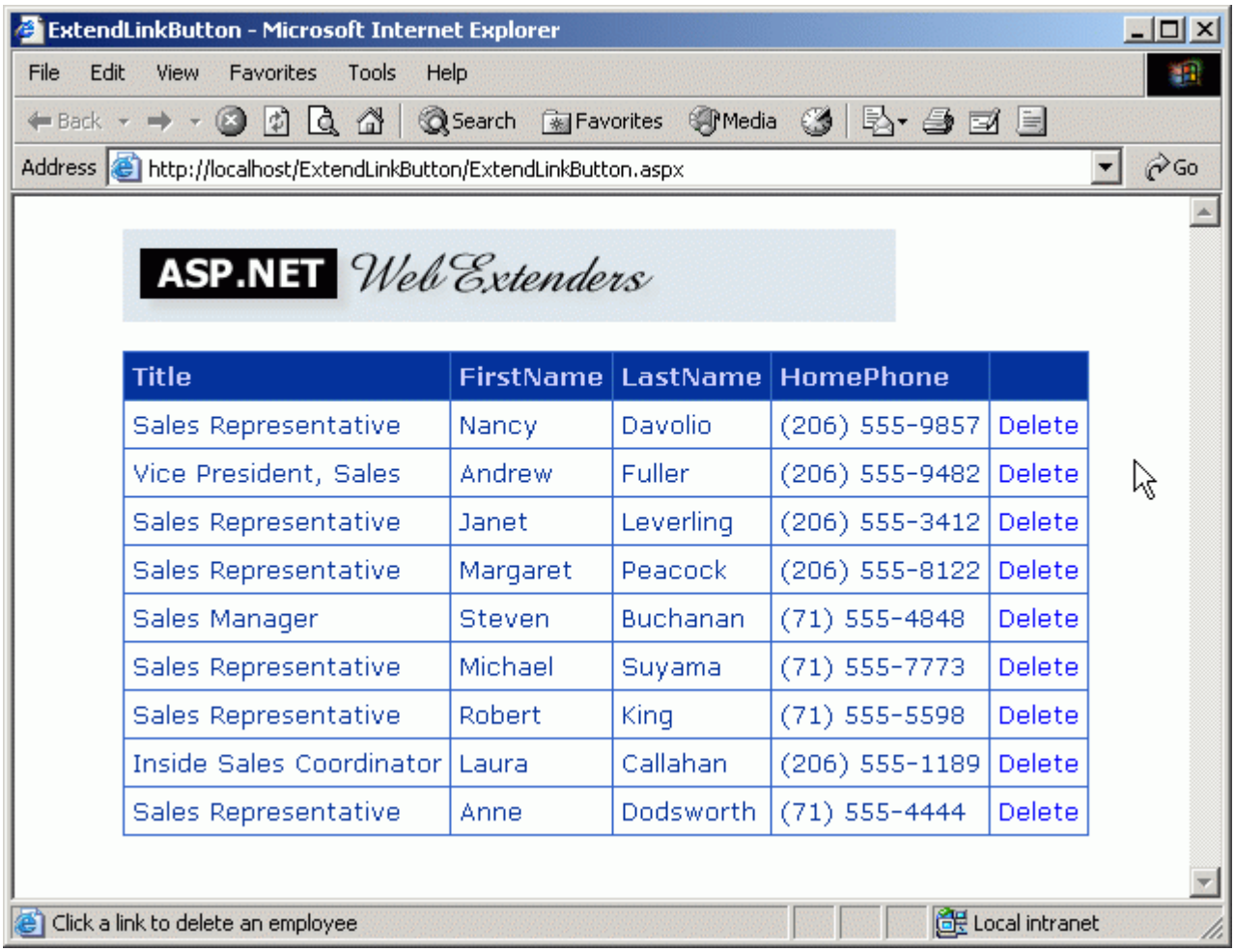
Going back into template editing mode and viewing the LinkButton properties again, we see that there is now a second extended property:



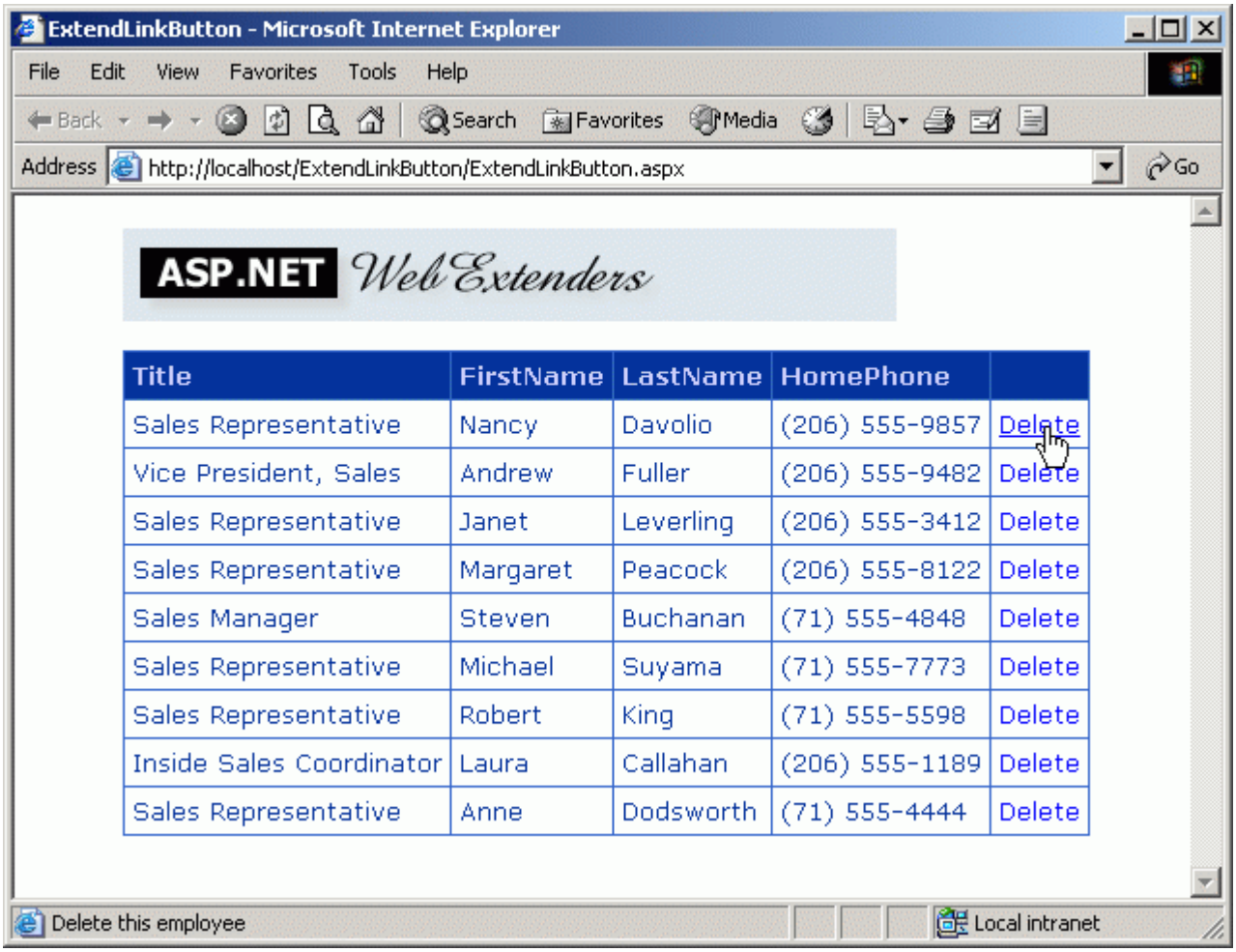
Setting the new property to *Are you sure you want to delete this employee?* will give us an appropriate 'last chance' confirmation dialog, with the option to abort the delete operation.

Running the Web Application

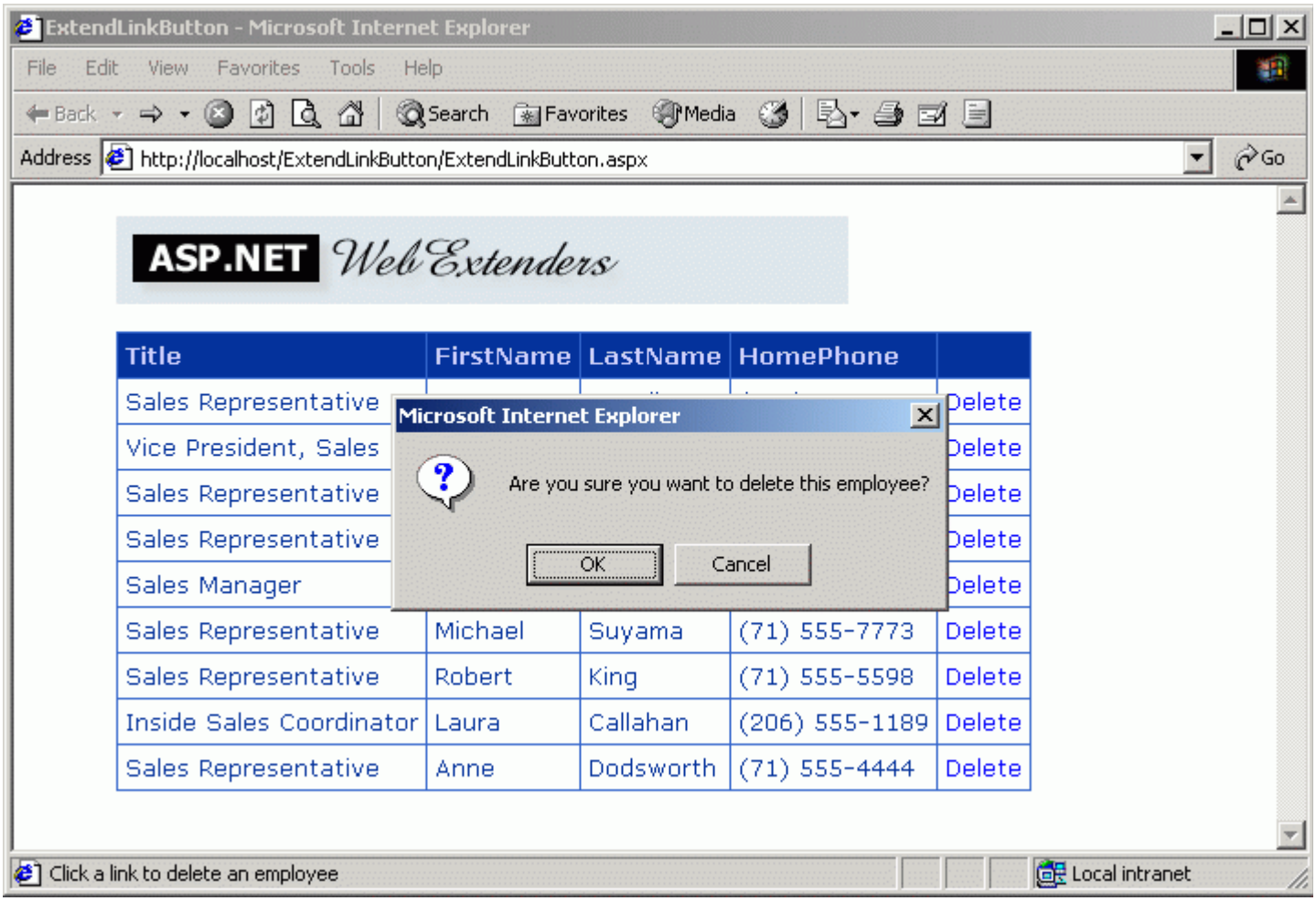
That's all that needs to be done at designtime. All that remains now is to run the application by hitting F5 to see the results...



… note the default status bar text before the cursor is moved over a link.



Moving the pointer over the record belonging to the hapless Nancy Davolio alters the status bar text accordingly. Her continued employment hangs by a thread …



Thanks to the ConfirmProvider however, she gets a reprieve ... cancelling the dialog prevents a postback, and her subsequent removal from the company database.

Summary

We've shown how WebExtenders can extend the capabilities of Web Server controls, and how they are used within the design environment. We have also seen how two WebExtenders can extend the behavior of the same Web Server control to give even more flexibility at runtime.

To try a free trial version of WebExtenders please visit the WinformReports website at http://www.winformreports.co.uk

July 15, 2003