CAN THIRD-PARTY ASP.NET CONTROLS MAKE ME MORE PRODUCTIVE?

See the Answer—in Code

Telerik

CONTENTS

THE 30-MINUTE CHALLENGE

As a Developer Evangelist representing Telerik products at tech events, many people ask about the productivity gained by using third-party controls over the Microsoft ASP.NET controls. Then I thought, what better way to answer the question than with code?

So I gave myself a 30-minute challenge: build the same application two ways, once working with the default Visual Studio toolset, then again using **Telerik UI for ASP.NET AJAX**. For each attempt, I gave myself 30 minutes.

At the end, I planned to compare the results of working with each toolset on the basis of:

- How much functionality I could implement in the allotted 30 minutes
- How close the two applications resembled the original idea and look of the application
- The amount of code I had to write

What's in it for you?

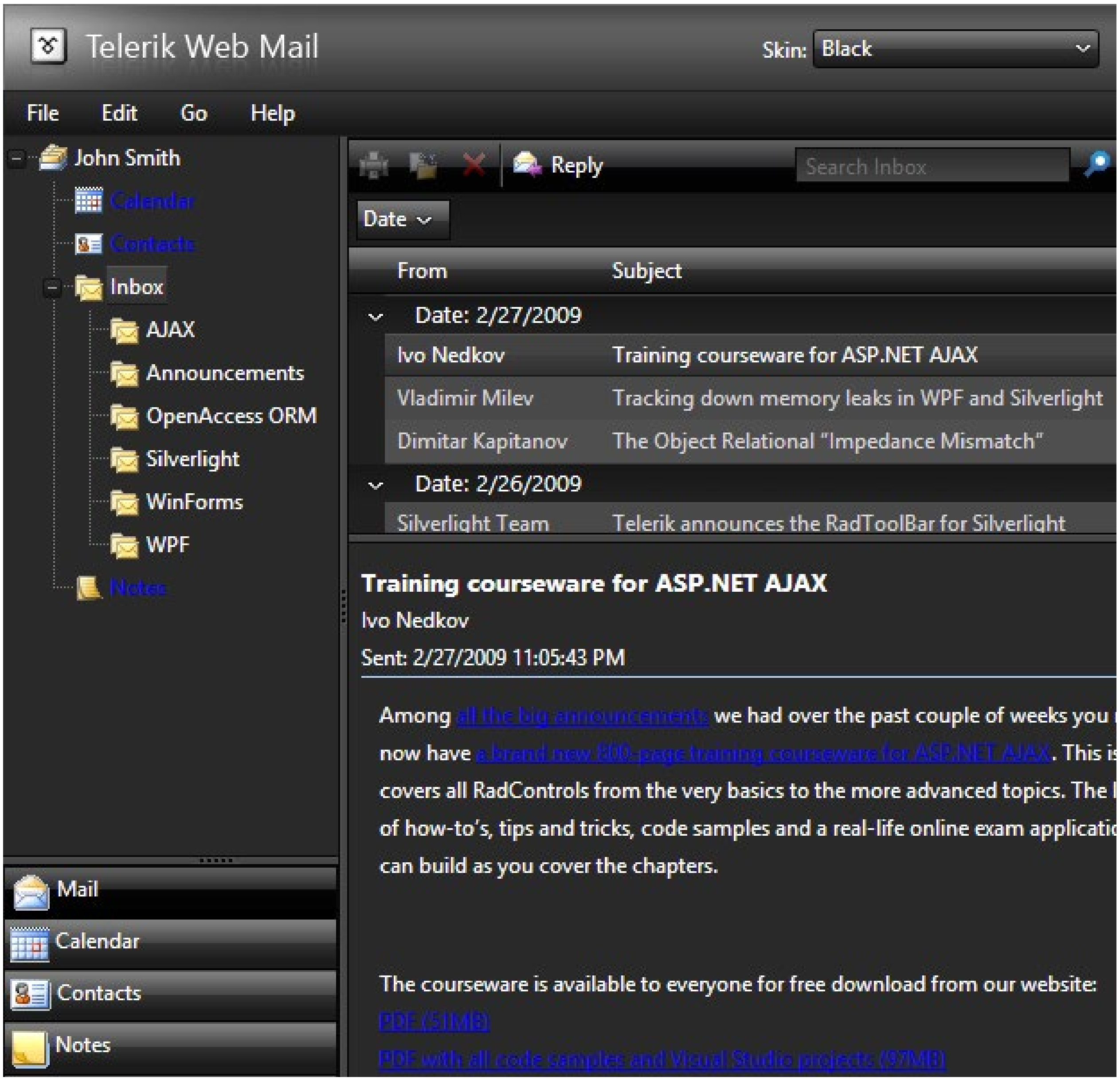
You will learn how to build one of the screens of a WebMail application with both toolsets, as well as see the gains produced by using third-party controls.

THE APPLICATION REQUIREMENTS

The web form I planned to develop was a WebMail application. For the sake of simplicity, I reused the design, already available CSS files and data access components of the **Telerik WebMail sample application.**

Clearly, I wasn't going to be able to build a full-blown application in 30 minutes, so I set a realistic goal to implement as much as possible from the home screen look and functionality. I used the Black Telerik Theme.

Here's the desired result:



From	Subject
Date: 2/27/2009	
Ivo Nedkov	Training courseware for ASP.NET AJAX
Vladimir Milev	Tracking down memory leaks in WPF and Silverlight
Dimitar Kapitanov	The Object Relational "Impedance Mismatch"
Date: 2/26/2009	
Silverlight Team	Telerik announces the RadToolBar for Silverlight

Training courseware for ASP.NET AJAX
Ivo Nedkov
Sent: 2/27/2009 11:05:43 PM

Among all the big announcements we had over the past couple of weeks you
now have a brand new 800-page training courseware for ASP.NET AJAX. This is
covers all RadControls from the very basics to the more advanced topics. The
of how-to's, tips and tricks, code samples and a real-life online exam applicati
can build as you cover the chapters.

The courseware is available to everyone for free download from our website:
PDF (51MB)
PDF with all code samples and Visual Studio projects (97MB)

LET'S CODE

Layout and Header

DEFAULT ASP.NET CONTROLS:

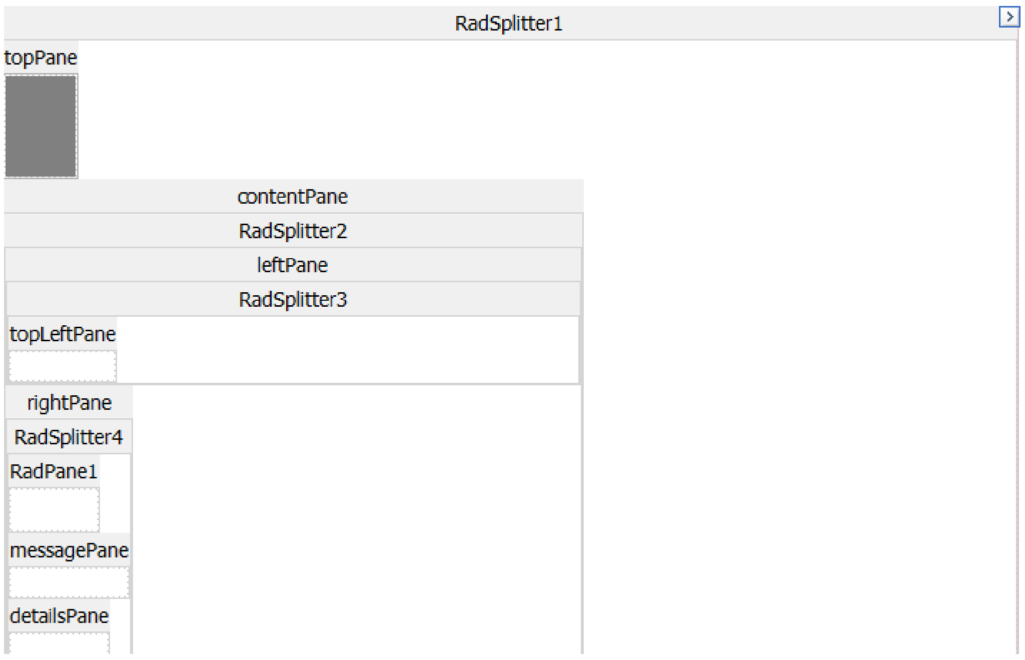
The layout of the page is intended to be responsive, meaning it should shrink and grow with the browser. To accomplish this with the Microsoft ASP.Net controls, I wrapped my page in series of DIVs that allowed for resizing. I would look at the JavaScript and CSS, only if I had time remaining at the end. This was a layout with five sections, so I typed the HTML by hand with some copy-paste help to quickly generate the below few lines of markup:

```
<div id="headerTopNavigation" style="height: 74px; display: block; border-bottom: 1px solid #808080; background-color: #000;">
</div>
<div id="treeNavigation" style="width: 240px; float: left; border-right: 1px solid #000; display: block; height: 100%;">
</div>
<div id="toolbar" style="display: block;">
</div>
<div id="messagegrid" style="display: block; float: left; width: 480px; height: 50px;">
</div>
<br />
<div id="messagecontent" style="display: block; margin-top: 30px;">
</div>
```

CODE LISTING 1 - THE DIV MARKUP FOR THE MICROSOFT ASP.NET LAYOUT

TELERIK UI FOR ASP.NET AJAX:

To achieve this resizing capability with the Telerik controls, I nested them inside of the **Splitter control** which provides it automatically. I will build this layout by using the control toolbox to drag the appropriate panels and splitters onto the design surface in Visual Studio. My initial layout with this tool looks like this:

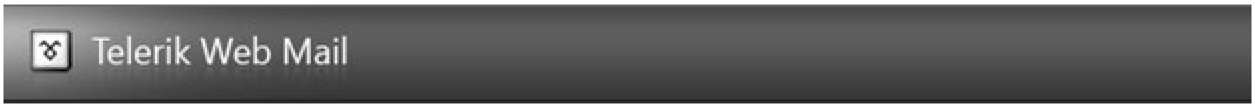


By defining some attributes in the property window for these controls, I very quickly got the following markup that more clearly defined my intentions for this layout:

```
<telerik:RadSplitter runat="server" ID="RadSplitter1" Height="100%" Width="100%" Orientation="Horizontal" 
BorderStyle="None"
    BorderSize="0" PanesBorderSize="0">
    <telerik:RadPane runat="server" ID="topPane" Height="74px" BackColor="Gray">
    </telerik:RadPane>
    <telerik:RadPane runat="server" ID="contentPane" Scrolling="None">
        <telerik:RadSplitter runat="server" BorderStyle="None" PanesBorderSize="0">
            <telerik:RadPane runat="server" ID="leftPane" Width="240px" MinWidth="240" MaxWidth="240" 
Scrolling="None">
                <telerik:RadSplitter runat="server" BorderStyle="None" PanesBorderSize="0" Height="100%"
                    Orientation="Horizontal">
                    <telerik:RadPane runat="server" ID="topLeftPane">
                    </telerik:RadPane>
                </telerik:RadSplitter>
            </telerik:RadPane>
            <telerik:RadPane runat="server" ID="rightPane" CssClass="right-pane" Scrolling="None">
                <telerik:RadSplitter runat="server" Width="100%" BorderSize="0" BorderStyle="None" 
PanesBorderSize="0"
                    Height="100%" Orientation="Horizontal">
                    <telerik:RadPane runat="server" Height="32px" EnableViewState="false" Scrolling="None">
                    </telerik:RadPane>
                    <telerik:RadPane runat="server" ID="messagePane">
                    </telerik:RadPane>
                    <telerik:RadPane runat="server" ID="detailsPane">
                    </telerik:RadPane>
                </telerik:RadSplitter>
            </telerik:RadPane>
        </telerik:RadSplitter>
   </telerik:RadPane>
</telerik:RadSplitter>
```

CODE LISTING 2 - THE RADSPLITTER MARKUP FOR THE TELERIK LAYOUT

The header area of this screen was a simple bar with a background gradient and logo image.



In both scenarios, this was a standard DIV element with format provided in the shared CSS stylesheet.

```
<div class="header">

    <div class="logo"></div>
</div>
```

CODE LISTING 3 – THE SHARED HEADER WITH LOGO MARKUP

PRODUCTIVITY RECAP

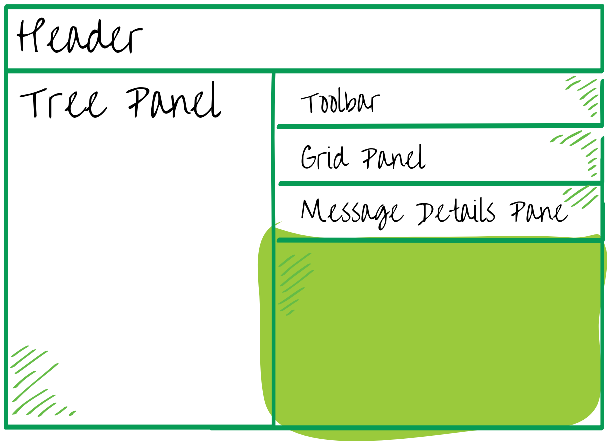
At the end of each section, I have included small, preliminary comparisons between the results I achieved with the default ASP.NET controls versus those using Telerik UI for ASP.NET AJAX.

To make the feature and time comparisons fair, I decided to stay after the 30 minute challenge and implement in the MS sample app the same functionality I have out-of-the-box in the Telerik one. These items are marked with a star to denote that they were completed after the challenge.

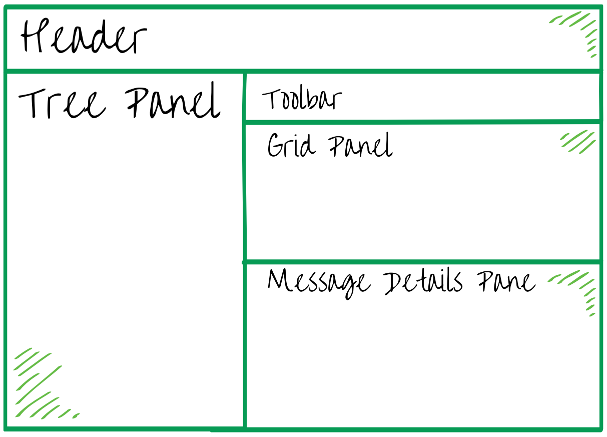
Functionality Needed	Time to Achieve With the Default ASP.Net Controls	Time to Achieve With Telerik UI for ASP.NET AJAX
Markup Generation	1 minute	5 minutes
Styling and Formatting	4 minutes	0 minutes
Automatic Resizing on Browser Resize	20 minutes to adjust CSS*	0 minutes (out-of-the-box functionality)
Total:	**25 minutes**	**5 minutes**

UI RESULTS

I added coloring to the layout DIVs to see how the layout would turn out with each toolkit. Since I am not a designer, I struggled to make the layout fill the entire browser window with the Default ASP.NET controls. I set the Telerik layout with RadSplitter which automatically filled the entire viewport without showing an undesired "blank space" at the bottom of the ASP.NET layout.



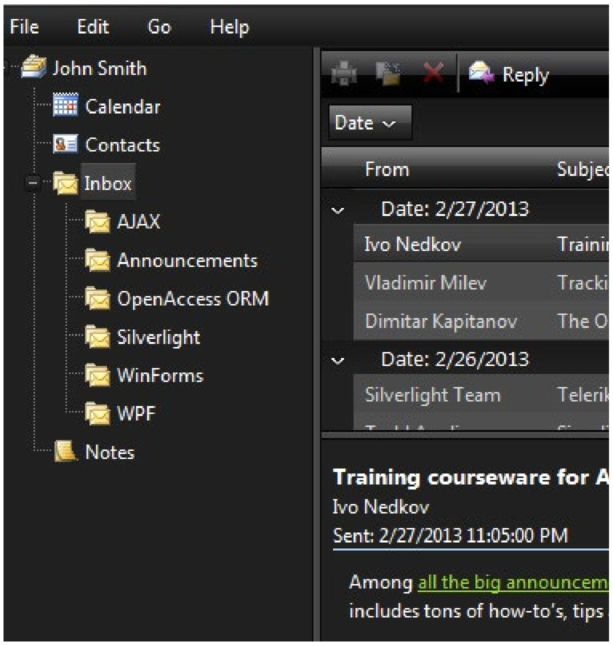
DEFAULT ASP.NET LAYOUT WITH HIGHLIGHTING TO SHOW EACH PANEL



TELERIK LAYOUT WITH HIGHLIGHTING TO SHOW EACH PANEL

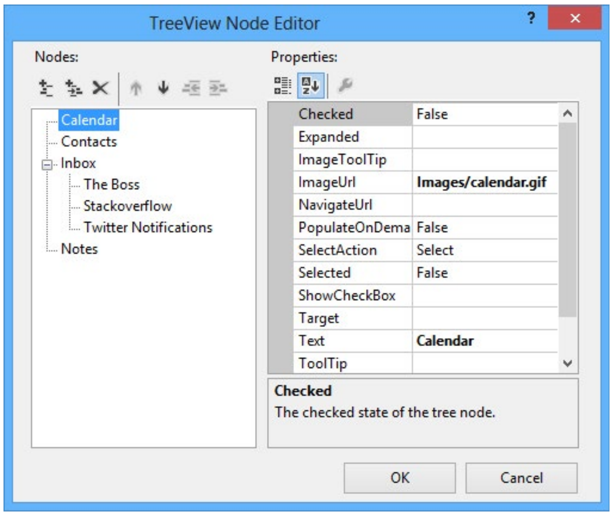
Creating the Navigation TreeView

The next area on the screen was a pair of panels, with a tree control on the left and a main content area on the right:



DEFAULT ASP.NET CONTROLS:

With the default controls, I created another DIV and added an asp:TreeView control to that. The DIV allocated the entire height of the page for the TreeView and provided the ability for it to scroll vertically if necessary. I defined the content nodes of the TreeView using the node editor available from the control's smart tag:



Checked	False
Expanded	
ImageToolTip	
ImageUrl	Images/calendar.gif
NavigateUrl	
PopulateOnDema	False
SelectAction	Select
Selected	False
ShowCheckBox	
Target	
Text	Calendar
ToolTip	

Checked
The checked state of the tree node.

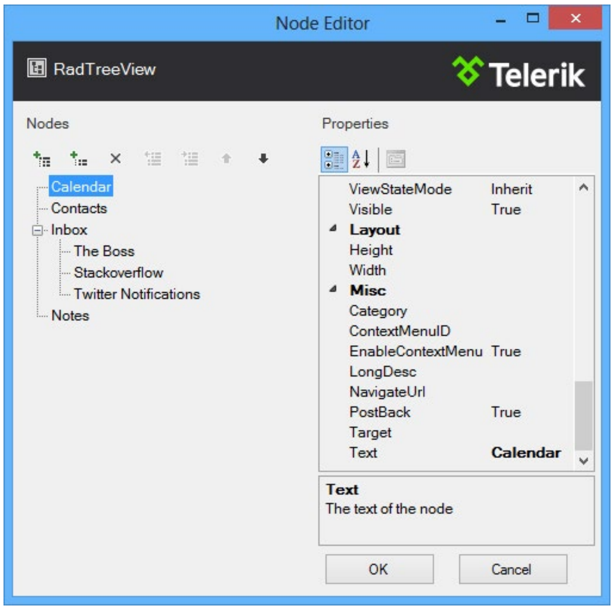
With a little bit of mousing and typing in this dialog, I generated the following markup:

```
<div id="treeNavigation" style="width: 240px; float: left; border-right: 1px solid #000; display: block;
height: 100%;">
    <asp:TreeView runat="server" ID="folders" ShowLines="true">
        <Nodes>
            <asp:TreeNode Text="Calendar" ImageUrl="Images/calendar.gif"></asp:TreeNode>
            <asp:TreeNode Text="Contacts" ImageUrl="Images/contacts.gif"></asp:TreeNode>
            <asp:TreeNode Text="Inbox" ImageUrl="Images/inbox.gif" Expanded="true">
                <asp:TreeNode Text="The Boss" ImageUrl="Images/inbox.gif"></asp:TreeNode>
                <asp:TreeNode Text="Stackoverflow" ImageUrl="Images/inbox.gif"></asp:TreeNode>
                <asp:TreeNode Text="Twitter Notifications" ImageUrl="Images/inbox.gif"></asp:TreeNode>
            </asp:TreeNode>
            <asp:TreeNode Text="Notes" ImageUrl="images/notes.gif"></asp:TreeNode>
        </Nodes>
    </asp:TreeView>
```

CODE LISTING 4: THE MICROSOFT ASP.NET TREEVIEW

TELERIK UI FOR ASP.NET AJAX:

The **Telerik ASP.NET Tree control** has a similar structure, with the only difference being the presence of a "Nodes" element to define the hierarchy of tree nodes. I used the similar Telerik RadTreeView Node editor to generate the same tree nodes:



ViewStateMode	Inherit
Visible	True
Layout	
Height	
Width	
Misc	
Category	
ContextMenuID	
EnableContextMenu	True
LongDesc	
NavigateUrl	
PostBack	True
Target	
Text	**Calendar**

Text
The text of the node

After mousing and clicking through the menus and options, the following markup was generated:

```
<telerik:RadTreeView runat="server" ID="folders">
  <Nodes>
    <telerik:RadTreeNode Text="Calendar" ImageUrl="Images/Calendar.gif" />
    <telerik:RadTreeNode Text="Contacts" ImageUrl="Images/contacts.gif" />
    <telerik:RadTreeNode Text="Inbox" ImageUrl="Images/Inbox.gif" Expanded="true" Selected="true">
      <Nodes>
        <telerik:RadTreeNode Text="The Boss" ImageUrl="Images/Inbox.gif" />
        <telerik:RadTreeNode Text="Stackoverflow" ImageUrl="Images/Inbox.gif" />
        <telerik:RadTreeNode Text="Twitter Notifications" ImageUrl="Images/Inbox.gif" />
      </Nodes>
    </telerik:RadTreeNode>
    <telerik:RadTreeNode Text="Notes" ImageUrl="Images/notes.gif" />
  </Nodes>
</telerik:RadTreeView>
```

CODE LISTING 5 – THE TELERIK RADTREEVIEW

In a production application, I might need to have the content of the Tree come from a data source. This was possible with both controls, and binding to a hierarchical data source was a fully supported feature of the Telerik TreeView control.

PRODUCTIVITY RECAP

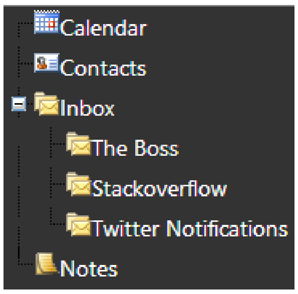
Functionality Needed	Time to Achieve With the Default ASP.NET Controls	Time to Achieve With Telerik UI for ASP.NET AJAX
Markup Generation	1 minute	2 minutes
Styling and Formatting	0 minutes (used the default formatting)	0 minutes (comes from the built-in skin)
TreeView Layout and Content	2 minutes	2 minutes
TreeView: Image Alignment	10 minutes to fix CSS*	0 (comes from the built-in skin)
Total:	13 minutes	4 minutes

*ITEMS MARKED WITH A STAR WERE COMPLETED AFTER THE CHALLENGE

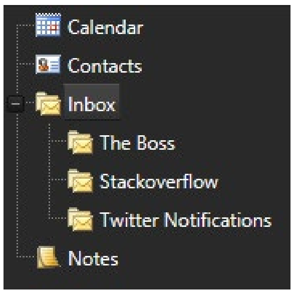
UI RESULTS

When we compared the two TreeView controls, we saw some immediate differences:

- The dotted lines representing the hierarchy in the MS controls broke at some points and were not clearly visible on the black background
- The MS TreeView had no margins between the icons and the text, while this benefit comes out-of-the-box with the Telerik controls
- The MS TreeView lacked hovered and selected state styles, while this was included with the Telerik skin



DEFAULT ASP.NET TREEVIEW



TELERIK TREEVIEW

PERFORMANCE RESULTS

The most interesting comparison between the two Tree controls was the HTML generated from this markup. When I examined the source of the two tree controls, I saw that the standard ASP.NET tree rendered a series of nested tables. The Telerik tree rendered a set of unordered lists with the expected structure that mirrored the node markup defined earlier. This choice of HTML layout in ASP.NET was reflected in the volume of markup sent to the browser:

ASP.Net TreeView HTML size	Telerik RadTreeView HTML size
6.28kb	1.58kb

Building the MenuBar

DEFAULT ASP.NET CONTROLS

Below is the markup I needed to implement the navigation with the Microsoft ASP.NET Menu:

```
<asp:Menu runat="server" ID="menu" Orientation="Horizontal" style="background-color: #000;">
    <Items>
        <asp:MenuItem Text="File">
            <asp:MenuItem Text="New" Enabled="false"></asp:MenuItem>
            <asp:MenuItem Text="Open" Enabled="false"></asp:MenuItem>
            <asp:MenuItem Text="--" Selectable="false"></asp:MenuItem>
            <asp:MenuItem Text="Save As" Enabled="false"></asp:MenuItem>
        </asp:MenuItem>
        <asp:MenuItem Text="Edit">
            <asp:MenuItem Text="Cut" Enabled="false"></asp:MenuItem>
            <asp:MenuItem Text="Copy" Enabled="false"></asp:MenuItem>
        </asp:MenuItem>
        <asp:MenuItem Text="Go">
            <asp:MenuItem Text="Mail"></asp:MenuItem>
            <asp:MenuItem Text="Calendar"></asp:MenuItem>
            <asp:MenuItem Text="Contacts"></asp:MenuItem>
            <asp:MenuItem Text="Notes"></asp:MenuItem>
        </asp:MenuItem>
        <asp:MenuItem Text="Help"></asp:MenuItem>
    </Items>
</asp:Menu>
```

CODE LISTING 6: MENU FORMATTED WITH THE DEFAULT ASP.NET CONTROLS

TELERIK UI FOR ASP.NET AJAX

The T**elerik menu bar control** had similar markup syntax to the Microsoft ASP.NET Menu, so the code looked alike between the two sets of controls. However, the output was vastly different:

```
<telerik:RadMenu runat="server" EnableRoundedCorners="true" style="float: none; position: absolute; top: 48px; z-index: 2000"
    Width="100%">
    <Items>
        <telerik:RadMenuItem Text="File">
            <Items>
                <telerik:RadMenuItem Text="New" Enabled="false"></telerik:RadMenuItem>
                <telerik:RadMenuItem Text="Open" Enabled="false"></telerik:RadMenuItem>
                <telerik:RadMenuItem IsSeparator="true"></telerik:RadMenuItem>
                <telerik:RadMenuItem Text="Save As" Enabled="false"></telerik:RadMenuItem>
            </Items>
         </telerik:RadMenuItem>
         <telerik:RadMenuItem Text="Edit">
             <Items>
                 <telerik:RadMenuItem Text="Cut" Enabled="false"></telerik:RadMenuItem>
                 <telerik:RadMenuItem Text="Copy" Enabled="false"></telerik:RadMenuItem>
             </Items>
         </telerik:RadMenuItem>
         <telerik:RadMenuItem Text="Go">
             <Items>
                  <telerik:RadMenuItem Text="Mail"></telerik:RadMenuItem>
                  <telerik:RadMenuItem Text="Calendar"></telerik:RadMenuItem>
                  <telerik:RadMenuItem Text="Contacts"></telerik:RadMenuItem>
                  <telerik:RadMenuItem Text="Notes"></telerik:RadMenuItem>
             </Items>
         </telerik:RadMenuItem>
         <telerik:RadMenuItem Text="Help"></telerik:RadMenuItem>
    </Items>
</telerik:RadMenu>
```
CODE LISTING 7: THE TELERIK RADMENU MARKUP

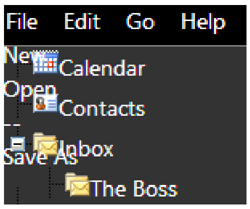
PRODUCTIVITY RECAP

Functionality Needed	Time to Achieve With the Default ASP.NET Controls	Time to Achieve With Telerik UI for ASP.NET AJAX
Markup Generation	2 minutes	2.5 minutes
Styling and Formatting	1 minute (used additional CSS)	0 (comes from the built-in skin)
Menu: Layout and Content	2 minutes	2 minutes
Menu: Rounded Corners	30 minutes to find and apply CSS templates *	0.16 minutes (turned on one property)
Menu: Separator between Items	30 minutes *	0.5 minutes (added markup and turned on one property)
Menu: Item Selection and Hover Highlighting	10 minutes to add CSS*	0 minutes (comes with the built-in skin)
Total:	75 minutes	5.16 minutes

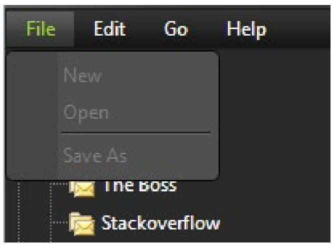
*ITEMS MARKED WITH A STAR WERE COMPLETED AFTER THE CHALLENGE.

UI RESULT

Clearly, I needed to spend more time writing CSS to make the Microsoft ASP.NET menu appear with a non-transparent background and show some sort of highlight on the MenuBar to indicate which top-level element was selected. The Telerik MenuBar had these options configured by default. Here was what I achieved:



DEFAULT ASP.NET

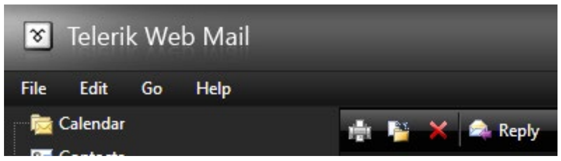


TELERIK MENU

FIGURE 1 – SIDE-BY-SIDE COMPARISON OF THE TWO MENUS, DEMONSTRATING THE TRANSPARENCY PROBLEM IN THE DEFAULT ASP.NET MENU AND THE DIFFERENCES IN THE SEPARATOR FUNCTIONALITY.

Building the Toolbar Control

On the right side of the desired application, there was a toolbar followed by a list of messages, and an area for the selected message content.



THE DESIRED HEADER WITH MENUBAR AND TOOLBAR

DEFAULT ASP.NET CONTROLS

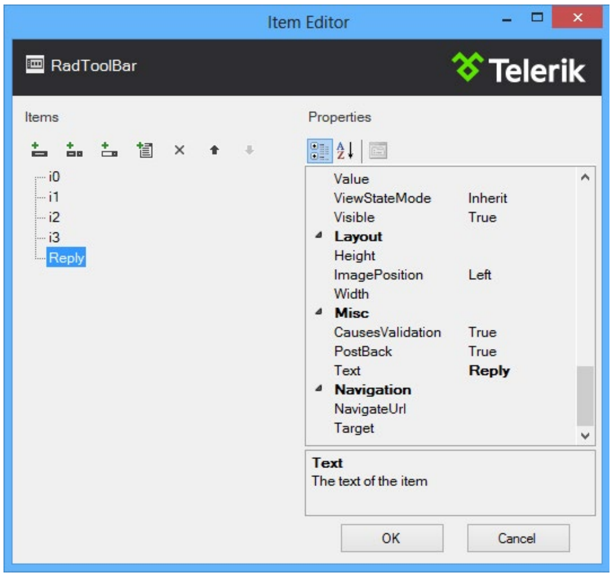
The toolbar looked like a simple collection of image buttons, so I hand-wrote some markup for the four image buttons:

```
<div id="toolbar" style="display: block;">
    <asp:ImageButton runat="server" ID="printButton" ImageUrl="images/print.gif" ToolTip="Print" />
    <asp:ImageButton runat="server" ID="moveButton" ImageUrl="images/move.gif" ToolTip="Move" />
    <asp:ImageButton runat="server" ID="deleteButton" ImageUrl="images/delete_inbox.gif" ToolTip="Delete"
/>
    &nbsp;
    <asp:ImageButton runat="server" ID="replyButton" ImageUrl="images/reply.gif" ToolTip="Reply"
AlternateText="Reply" />
</div>
```

CODE LISTING 8: THE MICROSOFT ASP.NET TOOLBAR

TELERIK UI FOR ASP.NET AJAX:

The **Telerik ASP.NET Toolbar component** was an easy choice to build a true toolbar on this screen. Once again, I used the Item Editor dialog (available from the Toolbar control's smart tag) to configure the items I wanted. I even managed to make the Reply button appear as a button with an image and text by just defining the text for it in the Item Editor:



Value
ViewStateMode Inherit
Visible True
Layout
Height
ImagePosition Left
Width
Misc
CausesValidation True
PostBack True
Text Reply
Navigation
NavigateUrl
Target

Text
The text of the item

This results in the following markup being generated:

```
<telerik:RadToolBar runat="server" CssClass="inbox-search-toolbar">
    <Items>
        <telerik:RadToolBarButton ImageUrl="Images/print.gif" ToolTip="Print" />
        <telerik:RadToolBarButton ImageUrl="Images/move.gif" ToolTip="Move" />
        <telerik:RadToolBarButton ImageUrl="Images/delete_inbox.gif" ToolTip="Delete" />
        <telerik:RadToolBarButton IsSeparator="true" />
        <telerik:RadToolBarButton ImageUrl="Images/reply.gif" Text="Reply" />
    </Items>
</telerik:RadToolBar>
```

CODE LISTING 9 – THE TELERIK RADTOOLBAR

PRODUCTIVITY RECAP

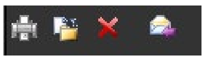
Functionality Needed	Time to Achieve With the Default ASP.NET Controls	Time to Achieve With Telerik UI for ASP.NET AJAX
Markup Generation	1 minute	1 minute
Styling and Formatting	0 minutes (used the default formatting)	0 minutes (comes from the built-in skin)
Toolbar: Layout and Content	1 minute	1 minute
Toolbar: Button with Text and Image	2 minutes to convert button into a LinkButton with image and text content*	0.25 minutes (set two properties)
Toolbar: Item Separator	2 minutes to add vertical bar*	0.5 minutes (added the markup and turned on one property)
Total:	6 minutes	2.75 minutes

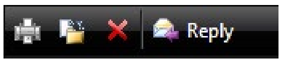
*ITEMS MARKED WITH A STAR WERE COMPLETED AFTER THE CHALLENGE.

UI RESULTS

The toolbar built with Telerik controls looked and felt like a real toolbar, rather than a collection of images provided by the Microsoft ASP.Net controls.

The vertical separator between the delete and reply buttons is not something that can be easily created in ASP.NET, but with the Telerik control it was as easy as setting the "IsSeperator" attribute to true. Adding the Reply text as part of the button is also not trivial with the standard ASP.NET controls.


DEFAULT ASP.NET TOOLBAR



TELERIK TOOLBAR

Configuring the Message Grid

I got to the point when I had to build the message grid. The additional behavior I had to implement was to display the selected message content in the box at the bottom of the screen.

DEFAULT ASP.NET CONTROLS:

I added the AJAX Extensions to the project with NuGet and added an asp:ScriptManager at the top of the page. With this element in place, I added an asp:UpdatePanel and wrapped an asp:GridView to show the list of messages without a full postback.

I had to do some style definition here to get the grid rows to appear with proper coloring. I also had to add some extra attributes to enable sorting, and a command column to allow the user to select a row in the grid. I referred to the standard LINQ data source, originally crafted for the webmail sample, and re-used it here.

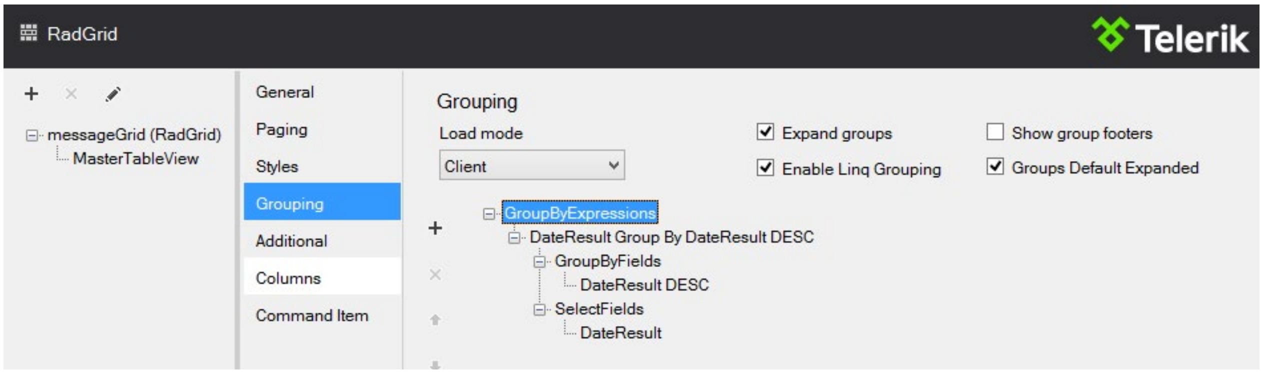
```
<asp:UpdatePanel ID="gridPanel" runat="server" UpdateMode="Conditional">
    <ContentTemplate>
        <asp:GridView runat="server" ID="grid" DataSourceID="dataSource" Width="800px" Height="300px"
            RowStyle-CssClass="gridRow" DataKeyNames="MessageID" AutoGenerateColumns="False"
            AlternatingRowStyle-BackColor="#494949" RowStyle-BackColor="#272727" SelectedRowStyle-
BackColor="#303030"
            HeaderStyle-ForeColor="Silver" AllowSorting="True">
            <Columns>
                <asp:CommandField ShowSelectButton="true" />
                <asp:BoundField DataField="From" HeaderText="From" SortExpression="From" />
                <asp:BoundField DataField="Received" HeaderText="Date" SortExpression="Received" />
                <asp:BoundField DataField="Subject" HeaderText="Subject" SortExpression="Subject" />
            </Columns>
        </asp:GridView>
    </ContentTemplate>
</asp:UpdatePanel>
```

CODE LISTING 10: MICROSOFT ASP.NET GRIDVIEW FOR MESSAGES

TELERIK UI FOR ASP.NET AJAX

To achieve the same thing with the Telerik controls, I took advantage of [Telerik DataGrid for ASP.NET AJAX.] I used the Design-Time Wizard of the grid and built a control featuring sorting, row selection and grouping of fields in just over a minute. All I have to do was check a few boxes, define how the group-by data would be calculated and I had my grid.

I then used the Design-Time Wizard of **RadAJAX** to quickly AJAX-enable the Grid.



Grouping

Load mode: Client

☑ Expand groups
☐ Show group footers
☑ Enable Linq Grouping
☑ Groups Default Expanded

```
GroupByExpressions
  DateResult Group By DateResult DESC
    GroupByFields
      DateResult DESC
    SelectFields
      DateResult
```

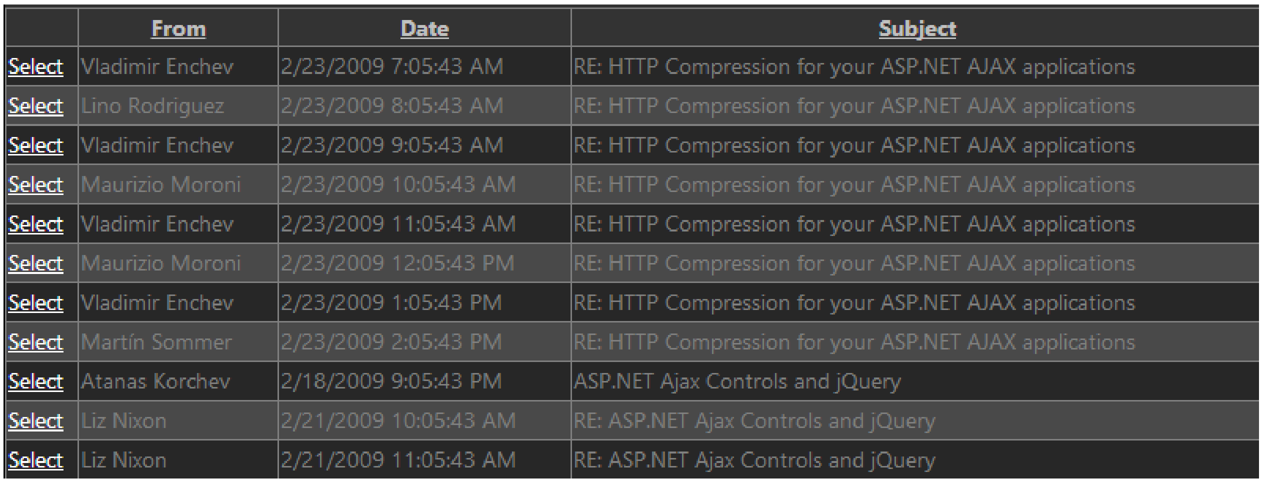
PRODUCTIVITY RECAP

Functionality Needed	Time to Achieve With the Default ASP.NET Controls	Time to Achieve With Telerik UI for ASP.NET AJAX
Markup Generation	5 minutes	2 minutes
Styling and Formatting	0.5 minutes (set row background colors)	0 minutes (comes from the built-in skin)
Grid: Layout and Content	5 minutes	2 minutes
Grid: Row Selection	0.5 minutes (added a new command column)	0.5 minutes (turned on three properties)
	60 minutes to implement a plugin with jQuery*	
Grid: Sorting	1 minute (used the Design-time Wizard and added SortExpressions to each column)	0.15 minutes (used the Design-time Wizard to check a box)
Grid: Grouping	180 minutes (re-wrote grid using a jQuery grid plugin)*	0.15 minutes (used the Design-time Wizard to check a box)
Grid: Automatic Resizing on Browser Resize	20 minutes to adjust CSS of all elements*	0 minutes (out-of-the-box functionality)
Grid: AJAX-Enable	2 minutes (added an UpdatePanel)	1 minute (used the Design-time Wizard)
Total:	274 minutes	5.8 minutes

*ITEMS MARKED WITH A STAR WERE COMPLETED AFTER THE CHALLENGE.

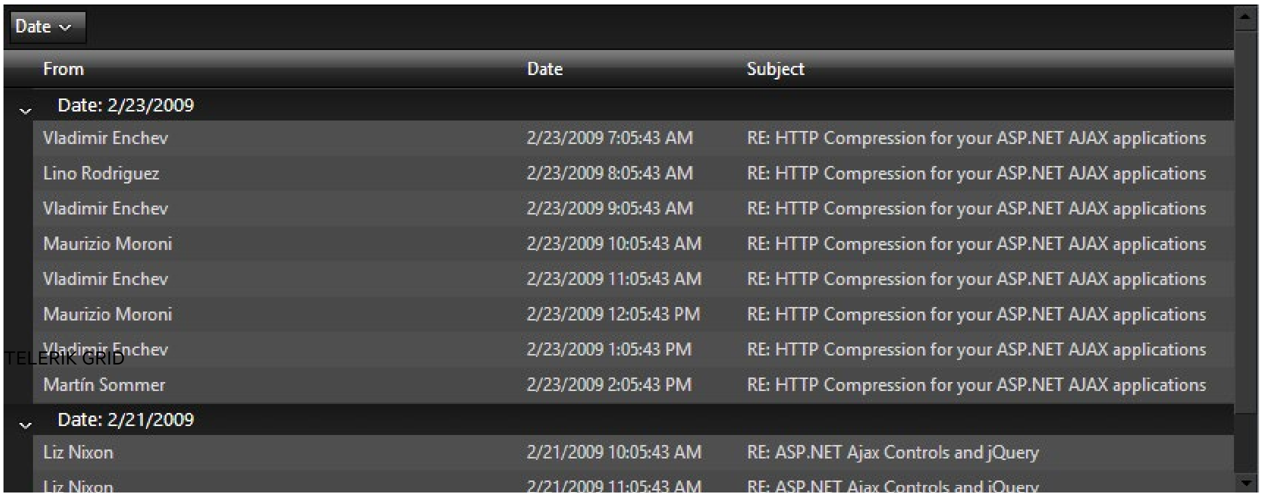
UI RESULTS

I spent no time styling Telerik Grid, just used the Black skin as is. I tried to apply custom CSS to the MS GridView to make it look the same and here's what I got:



	From	Date	Subject
Select	Vladimir Enchev	2/23/2009 7:05:43 AM	RE: HTTP Compression for your ASP.NET AJAX applications
Select	Lino Rodriguez	2/23/2009 8:05:43 AM	RE: HTTP Compression for your ASP.NET AJAX applications
Select	Vladimir Enchev	2/23/2009 9:05:43 AM	RE: HTTP Compression for your ASP.NET AJAX applications
Select	Maurizio Moroni	2/23/2009 10:05:43 AM	RE: HTTP Compression for your ASP.NET AJAX applications
Select	Vladimir Enchev	2/23/2009 11:05:43 AM	RE: HTTP Compression for your ASP.NET AJAX applications
Select	Maurizio Moroni	2/23/2009 12:05:43 PM	RE: HTTP Compression for your ASP.NET AJAX applications
Select	Vladimir Enchev	2/23/2009 1:05:43 PM	RE: HTTP Compression for your ASP.NET AJAX applications
Select	Martín Sommer	2/23/2009 2:05:43 PM	RE: HTTP Compression for your ASP.NET AJAX applications
Select	Atanas Korchev	2/18/2009 9:05:43 PM	ASP.NET Ajax Controls and jQuery
Select	Liz Nixon	2/21/2009 10:05:43 AM	RE: ASP.NET Ajax Controls and jQuery
Select	Liz Nixon	2/21/2009 11:05:43 AM	RE: ASP.NET Ajax Controls and jQuery

DEFAULT ASP.NET GRIDVIEW



From	Date	Subject
Date: 2/23/2009		
Vladimir Enchev	2/23/2009 7:05:43 AM	RE: HTTP Compression for your ASP.NET AJAX applications
Lino Rodriguez	2/23/2009 8:05:43 AM	RE: HTTP Compression for your ASP.NET AJAX applications
Vladimir Enchev	2/23/2009 9:05:43 AM	RE: HTTP Compression for your ASP.NET AJAX applications
Maurizio Moroni	2/23/2009 10:05:43 AM	RE: HTTP Compression for your ASP.NET AJAX applications
Vladimir Enchev	2/23/2009 11:05:43 AM	RE: HTTP Compression for your ASP.NET AJAX applications
Maurizio Moroni	2/23/2009 12:05:43 PM	RE: HTTP Compression for your ASP.NET AJAX applications
Vladimir Enchev	2/23/2009 1:05:43 PM	RE: HTTP Compression for your ASP.NET AJAX applications
Martín Sommer	2/23/2009 2:05:43 PM	RE: HTTP Compression for your ASP.NET AJAX applications
Date: 2/21/2009		
Liz Nixon	2/21/2009 10:05:43 AM	RE: ASP.NET Ajax Controls and jQuery
Liz Nixon	2/21/2009 11:05:43 AM	RE: ASP.NET Ajax Controls and jQuery

TELERIK GRID

TELERIK ASP.NET GRID

Showing the Message Content

DEFAULT ASP.NET CONTROLS

For the message content box in the webmail sample I will use the standard asp:DetailsView control, so there is nothing really to see here other than the additional CSS work that I need to make it fit in with the look and feel of the screen. I'll define that control and add it to the screen in an asp:UpdatePanel to get the select–update effect from the grid:

```
<div id="messagecontent" style="display: block; margin-top: 30px;">
<asp:UpdatePanel ID="UpdatePanel1" runat="server" UpdateMode="Conditional">
    <ContentTemplate>
        <asp:DetailsView runat="server" ID="details" DataSourceID="detailsData" Style="color: #fff; display: block;
            margin-left: 240px; margin-top: 300px; width: 800px; height: 400px; overflow-y: scroll" AutoGenerateRows="false"
            CssClass="message-view" GridLines="None" EnableViewState="false">
            <Fields>
                <asp:TemplateField ShowHeader="false">
                    <ItemTemplate>
                        <ul>
                            <li>
                                <h3 id="subject"><%# Eval("Subject") %></h3>
                            </li>
                            <li><span id="from"><%# Eval("From") %></span></li>
                            <li>Sent: <span id="sent"><%# Eval("Received") %></span></li>
                        </ul>
                        <div id="message-body">
                            <%# ((string)Eval("Body")).Replace("\n", "<br />") %>
                        </div>
                    </ItemTemplate>
                </asp:TemplateField>
            </Fields>
        </asp:DetailsView>
    </ContentTemplate>
    <Triggers>
        <asp:AsyncPostBackTrigger ControlID="grid" />
    </Triggers>
</asp:UpdatePanel>
</div>
```

CODE LISTING 11: SHOWING THE MESSAGE CONTENT WITH THE MICROSOFT ASP.NET CONTROLS

With those pieces in place, I had the ASP.NET grid loading data and selection of a message presenting content at the bottom properly.

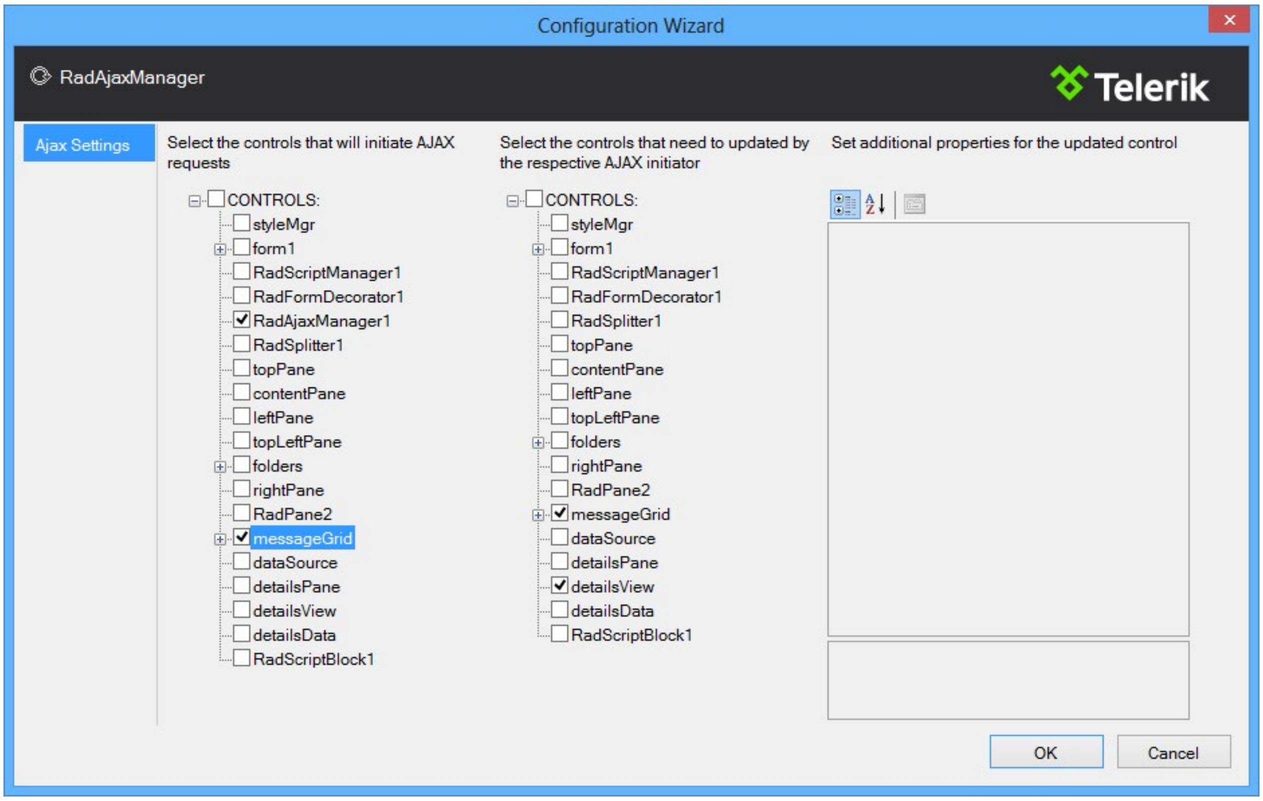
TELERIK UI FOR ASP.NET AJAX

To draw these same features on the Telerik app, I defined a details pane at the bottom of the screen. With all of the styling and layout provided, I was able to focus on just the control features to present the data. The content pane code looked like this:

```
<telerik:RadPane runat="server" ID="detailsPane">
    <asp:DetailsView runat="server" ID="detailsView" DataSourceID="detailsData" AutoGenerateRows="false"
        CssClass="message-view" GridLines="None">
        <Fields>
            <asp:TemplateField ShowHeader="false">
                <ItemTemplate>
                    <ul>
                        <li>
                            <h3 id="subject"><%# Eval("Subject") %></h3>
                        </li>
                        <li><span id="from"><%# Eval("From") %></span></li>
                        <li>Sent: <span id="sent"><%# Eval("Received") %></span></li>
                    </ul>
                    <div id="message-body">
                        <%# ((string)Eval("Body")).Replace("\n","<br/>") %>
                    </div>
                </ItemTemplate>
            </asp:TemplateField>
        </Fields>
    </asp:DetailsView>
</telerik:RadPane>
```

CODE LISTING 12: SHOWING THE MESSAGE CONTENT WITH THE TELERIK ASP.NET

I then used the Design-Time Wizard of RadAjax to quickly AJAX-enable the DetailsView:



Select the controls that will initiate AJAX requests

- CONTROLS:
  - styleMgr
  - form1
  - RadScriptManager1
  - RadFormDecorator1
  - [✓] RadAjaxManager1
  - RadSplitter1
  - topPane
  - contentPane
  - leftPane
  - topLeftPane
  - folders
  - rightPane
  - RadPane2
  - [✓] messageGrid
  - dataSource
  - detailsPane
  - detailsView
  - detailsData
  - RadScriptBlock1

Select the controls that need to updated by the respective AJAX initiator

- CONTROLS:
  - styleMgr
  - form1
  - RadScriptManager1
  - RadFormDecorator1
  - RadSplitter1
  - topPane
  - contentPane
  - leftPane
  - topLeftPane
  - folders
  - rightPane
  - RadPane2
  - [✓] messageGrid
  - dataSource
  - detailsPane
  - [✓] detailsView
  - detailsData
  - RadScriptBlock1

Set additional properties for the updated control

In this case, there was no big difference in the amount of effort it took to AJAX-enable the Grid and DetailsView with UpdatePanels and RadAjax. However, in applications where more Update Panels need to be used, I would have spent more time setting the update mode and adding the proper triggers to each panel to tell it when to update and when not. On the other hand, RadAjax would have done this for me and the only thing I would have needed to do is check a few boxes in the Design-Time Wizard.

PRODUCTIVITY RECAP

Functionality Needed	Time to Achieve With the Default ASP.NET Controls	Time to Achieve With Telerik UI for ASP.NET AJAX
Markup Generation	2.5 minutes	2.5 minutes
Styling and Formatting	0 minutes (used the default formatting)	0 minutes (comes from the built-in skin)
Markup Generation	2 minutes	2 minutes
DetailsView: AJAX-Enable	2 minutes (added an UpdatePanel)	1 minute (used the Design-time Wizard)
Total:	6.5 minutes	5.5 minutes

*ITEMS MARKED WITH A STAR WERE COMPLETED AFTER THE CHALLENGE.

OVERALL RESULTS

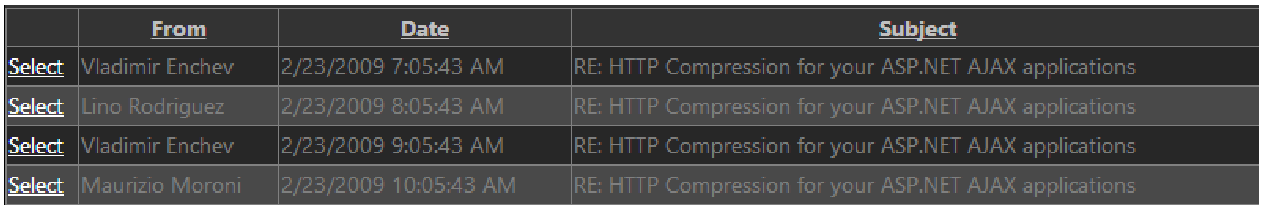
How Much Functionality Did I Manage to Implement With Each Toolset?

As you saw, I implemented significantly more functionality in the Telerik sample app in the given timeframe. Still, to make the feature and time comparison fair, I stayed after the 30 minute challenge expired to implement the Telerik sample app functionality in the MS version, tracking the time needed to do so. The overall results are as follows:

Functionality Needed	Time to Achieve With the Default ASP.NET Controls	Time to Achieve With Telerik UI for ASP.NET AJAX
Markup Generation	12.5 minutes	15.0 minutes
Styling and Formatting	8.5 minutes	0 minutes
Automatic Resizing on Browser Resize	20 minutes to adjust CSS*	0 minutes (out-of-the-box functionality)
TreeView: Image Alignment	10 minutes to fix CSS*	0 minutes (comes from the built-in skin)
Menu: Rounded Corners	30 minutes to find and apply CSS templates *	0.16 minutes (turned on one property)
Menu: Separator between Items	30 minutes *	0.5 minutes (added markup and turned on one property)
Menu: Item Selection and Hover Highlighting	10 minutes to add CSS*	0 minutes (comes with the built-in skin)
Toolbar: Button with Text and Image	2 minutes to convert button into a LinkButton with image and text content*	0.25 minutes (set two properties)
Toolbar: Item Separator	2 minutes to add vertical bar*	0.5 minutes (added the markup and turned on one property)
Grid: Row Selection	0.5 minutes (added a new command column) 60 minutes to implement a plugin with jQuery*	0.5 minutes (turned on three properties)
Grid: Sorting	1 minute (used the Design-time Wizard and added SortExpressions to each column)	0.15 minutes (used the Design-time Wizard to check a box)
Grid: Grouping	180 minutes (re-wrote grid using a jQuery grid plugin*)	0.15 minutes (used the Design-Time Wizard to check a box)
Grid: Automatic Resizing on Browser Resize	20 minutes to adjust CSS of all elements*	0 minutes (out-of-the-box functionality)
Grid: AJAX-Enable	2 minutes (added an UpdatePanel)	1 minute (used the Design-Time Wizard)
DetailsView: AJAX-Enable	2 minutes (added an UpdatePanel)	1 minute (used the Design-Time Wizard)
Total:	390.5 minutes	19.21 minutes

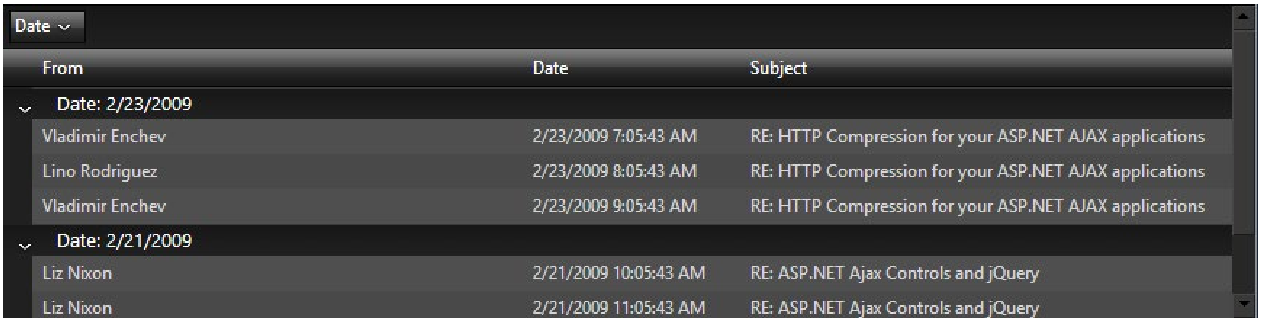
I added jQuery to the ASP.NET GridView, and after a few trials, was able to configure the whole row selection feature to look identical to the Telerik Grid. This effort took me about an hour to configure and test to ensure that it met the same level of quality as the Telerik Grid. This effort is clearly outside of the time allocated for this challenge, but we wanted to document it for the record.

I added sorting to the ASP.NET grid with some additional work, but it does not show any indicator in the header when a column is sorted. By contrast, the Telerik Grid has sorting with an indicator in the header and a highlight for the entire sorted column. The same amount of effort went into configuring each of these grids, but the Telerik Grid has the extra display features configured by default.



	From	Date	Subject
Select	Vladimir Enchev	2/23/2009 7:05:43 AM	RE: HTTP Compression for your ASP.NET AJAX applications
Select	Lino Rodriguez	2/23/2009 8:05:43 AM	RE: HTTP Compression for your ASP.NET AJAX applications
Select	Vladimir Enchev	2/23/2009 9:05:43 AM	RE: HTTP Compression for your ASP.NET AJAX applications
Select	Maurizio Moroni	2/23/2009 10:05:43 AM	RE: HTTP Compression for your ASP.NET AJAX applications

DEFAULT ASP.NET GRIDVIEW



From	Date	Subject
Date: 2/23/2009		
Vladimir Enchev	2/23/2009 7:05:43 AM	RE: HTTP Compression for your ASP.NET AJAX applications
Lino Rodriguez	2/23/2009 8:05:43 AM	RE: HTTP Compression for your ASP.NET AJAX applications
Vladimir Enchev	2/23/2009 9:05:43 AM	RE: HTTP Compression for your ASP.NET AJAX applications
Date: 2/21/2009		
Liz Nixon	2/21/2009 10:05:43 AM	RE: ASP.NET Ajax Controls and jQuery
Liz Nixon	2/21/2009 11:05:43 AM	RE: ASP.NET Ajax Controls and jQuery

TELERIK GRID

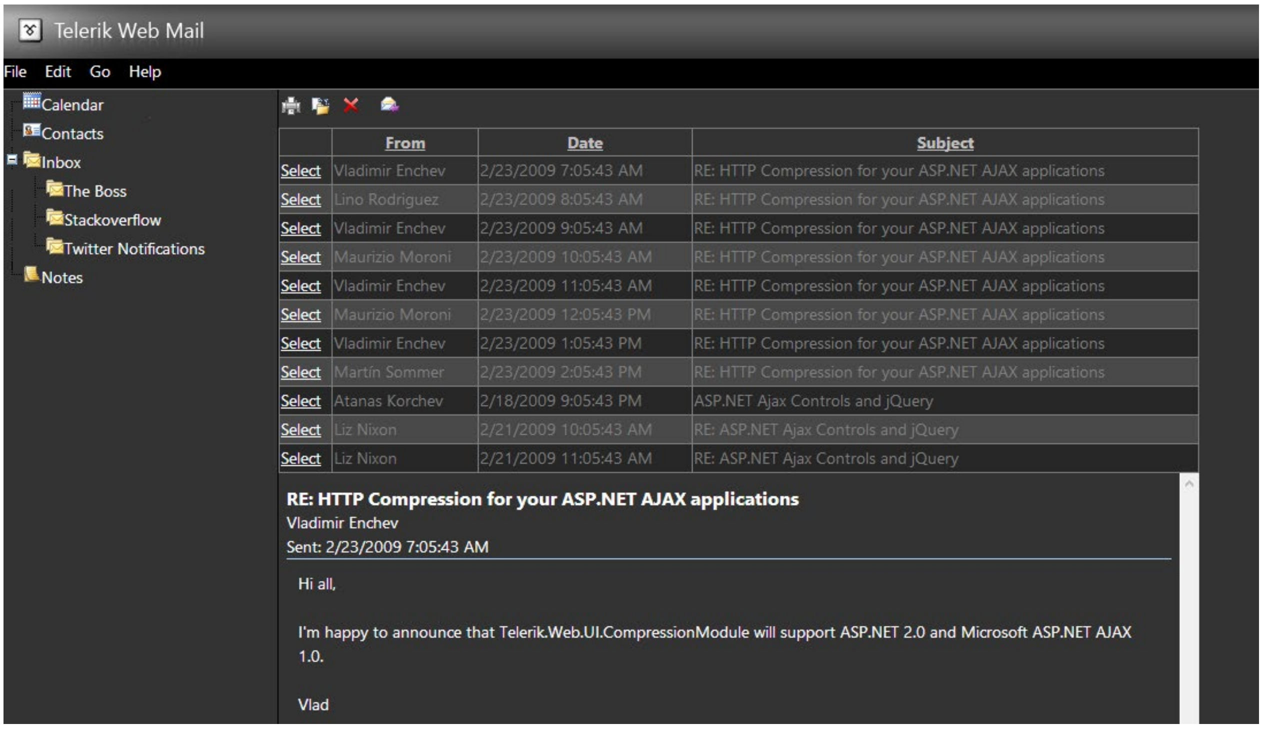
Grouping is not available at all in the ASP. Net GridView. With the Telerik Grid, I enabled it with a few configuration options and it was available within a minute or two. I was able to find options to use jQuery to add grouping capabilities to an HTML table with a WebAPI service supplying data. This would have required a complete rewrite of the table and an estimated two to four hours to complete. Once again, this was outside of the bounds of this comparison, but we are documenting it here to create a comprehensive view.

The Telerik screens were built with the RadSplitter control, allowing for dynamic resizing of the controls on screen as the browser changes shape. Additionally, users will be able to drag the separator bars between the elements on screen to resize some of the panels and give more room to others. The default ASP.Net controls have no such feature, and would need this functionality written with JavaScript. I estimated another 15-30 minutes to add the jQueryUI resizable widget and configure it properly for the default ASPX controls.

Both screens were able to make the “click and display the correct message” functionality work properly with Ajax.

How Close Are The Two Applications to the Original Idea?

Looking at the two apps side-by-side, initial feedback was that they looked very similar:



	From	Date	Subject
Select	Vladimir Enchev	2/23/2009 7:05:43 AM	RE: HTTP Compression for your ASP.NET AJAX applications
Select	Lino Rodriguez	2/23/2009 8:05:43 AM	RE: HTTP Compression for your ASP.NET AJAX applications
Select	Vladimir Enchev	2/23/2009 9:05:43 AM	RE: HTTP Compression for your ASP.NET AJAX applications
Select	Maurizio Moroni	2/23/2009 10:05:43 AM	RE: HTTP Compression for your ASP.NET AJAX applications
Select	Vladimir Enchev	2/23/2009 11:05:43 AM	RE: HTTP Compression for your ASP.NET AJAX applications
Select	Maurizio Moroni	2/23/2009 12:05:43 PM	RE: HTTP Compression for your ASP.NET AJAX applications
Select	Vladimir Enchev	2/23/2009 1:05:43 PM	RE: HTTP Compression for your ASP.NET AJAX applications
Select	Martín Sommer	2/23/2009 2:05:43 PM	RE: HTTP Compression for your ASP.NET AJAX applications
Select	Atanas Korchev	2/18/2009 9:05:43 PM	ASP.NET Ajax Controls and jQuery
Select	Liz Nixon	2/21/2009 10:05:43 AM	RE: ASP.NET Ajax Controls and jQuery
Select	Liz Nixon	2/21/2009 11:05:43 AM	RE: ASP.NET Ajax Controls and jQuery

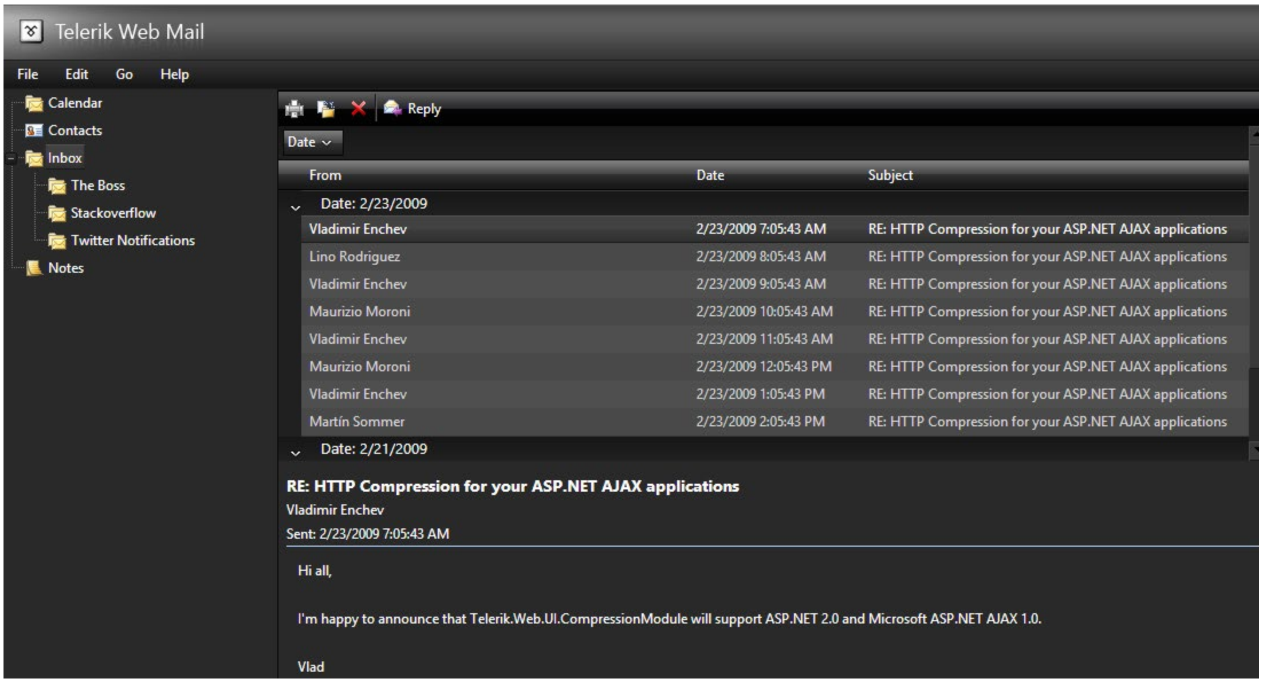
RE: HTTP Compression for your ASP.NET AJAX applications
Vladimir Enchev
Sent: 2/23/2009 7:05:43 AM

Hi all,

I'm happy to announce that Telerik.Web.UI.CompressionModule will support ASP.NET 2.0 and Microsoft ASP.NET AJAX 1.0.

Vlad

DEFAULT ASP.NET CONTROLS



From	Date	Subject
Date: 2/23/2009		
Vladimir Enchev	2/23/2009 7:05:43 AM	RE: HTTP Compression for your ASP.NET AJAX applications
Lino Rodriguez	2/23/2009 8:05:43 AM	RE: HTTP Compression for your ASP.NET AJAX applications
Vladimir Enchev	2/23/2009 9:05:43 AM	RE: HTTP Compression for your ASP.NET AJAX applications
Maurizio Moroni	2/23/2009 10:05:43 AM	RE: HTTP Compression for your ASP.NET AJAX applications
Vladimir Enchev	2/23/2009 11:05:43 AM	RE: HTTP Compression for your ASP.NET AJAX applications
Maurizio Moroni	2/23/2009 12:05:43 PM	RE: HTTP Compression for your ASP.NET AJAX applications
Vladimir Enchev	2/23/2009 1:05:43 PM	RE: HTTP Compression for your ASP.NET AJAX applications
Martín Sommer	2/23/2009 2:05:43 PM	RE: HTTP Compression for your ASP.NET AJAX applications
Date: 2/21/2009		

RE: HTTP Compression for your ASP.NET AJAX applications
Vladimir Enchev
Sent: 2/23/2009 7:05:43 AM

Hi all,

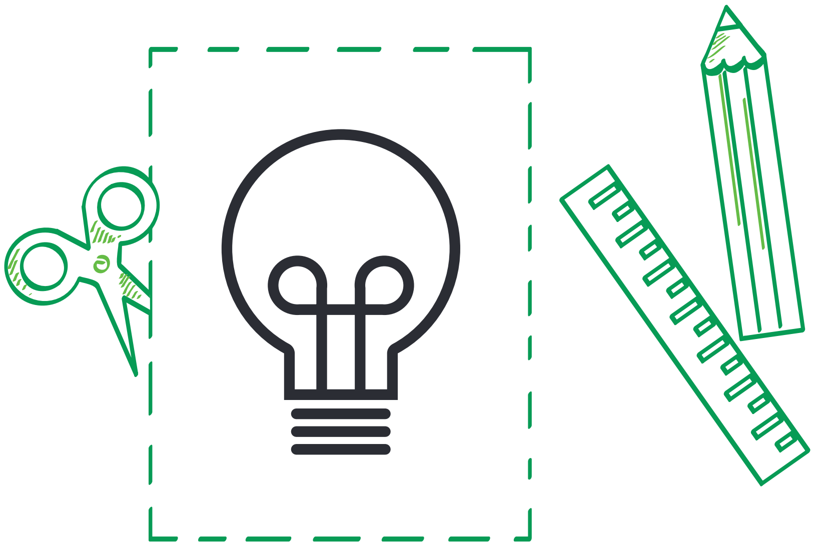
I'm happy to announce that Telerik.Web.UI.CompressionModule will support ASP.NET 2.0 and Microsoft ASP.NET AJAX 1.0.

Vlad

TELERIK ASP.NET CONTROLS

As one starts to dissect them though, it becomes clear that I couldn't apply a unified theme across all of the Microsoft ASP.NET controls:

- The fonts changed size and face through the application.
- The color scheme was not consistent. There were bits and pieces that did not quite fit together(i.e. the menu bar was solid black while the background was gray).
- The scrollbars were not themed.
- The Telerik grid enabled items to be selected by clicking anywhere, while the ASP.NET grid had a specific "Select" button. I tried to adjust this with some JavaScript, but ran out of time.
- The tree in the ASP.Net sample had poor margins and the lines connecting the branches were hard to see.
- The toolbar in the ASP.NET sample was incomplete, as there was no simple ImageAndText button in ASP.NET that could easily replicate the Reply button in the Telerik sample. Also, the watermarked Search textbox was missing.
- No gradients at all: our developer evangelist was just that, a developer, not a designer.
- No rounded corners. The ASP.NET controls all displayed within standard HTML squares, while the Telerik controls showed highlights in smooth rounded boxes.
- The menubar on the Telerik sample had a great animation to display the drop-down elements, while the MS menubar had no animation and presented a menu with a transparent background.
- The Telerik RadGrid had a hover effect when the user moved the mouse over items, while the MS grid did not have this effect by default. I did not have time to add the custom code.



How Much Code Did I Have to Write?

DEFAULT ASP.NET CONTROLS

I found very little tooling support to achieve the style and formatting in the MS controls. Most of the ASP.NET controls were built through the content wizards provided, then the HTML was customized by hand to add the styles and layout. This customization was what took me the longest, and prevented me from even attempting to add the CSS transitions or JavaScript interactions that were built into the Telerik controls.

In the MS ASP.NET sample, every control had some extra CSS defined in the ASPX page to achieve the overall look and theme presented. By contrast, the Telerik sample had zero CSS defined in the ASPX page. All of the styling was delivered from the theme specified in the external web.config file.

TELERIK UI FOR ASP.NET AJAX

You should know that the Telerik screen was built almost entirely with a mouse. The built-in wizards and templates enabled RadTreeView, RadMenuBar, RadToolBar and RadGrid to be built without hand-coding HTML.

NOW I EXTEND THE 30-MIN CHALLENGE TO YOU!

Productivity is something I don't take lightly. I know developers don't have the luxury of time in many cases, and need to deliver significant business value in short timeframes. Take 30 minutes to try building part of your current application with Telerik UI for ASP.NET AJAX and see just how much MORE you can get done in less time.

Download a fully-functional trial of Telerik's **more than 80 Telerik controls** for every project need. You will find controls for data visualization and management, navigation and layout, editing, interactivity and more.

In addition, the trial comes with 30 days of dedicated technical support from the people who build the controls.

APPLICATION SOURCE CODE

The code for both samples is available for download **here**

About the Author

Jeffrey T. Fritz is a Microsoft MVP in ASP.NET, an ASPInsider and Developer Evangelist for Telerik with over 15 years of experience building large-scale multi-tenant web applications in the software-as-a-service model. A Penn State graduate with a degree in Management Sciences and Information Systems, he speaks regularly at developer events through the INETA speaker program and maintains blogs at www.csharpfritz.com and **blogs.telerik.com/jefffritz**. You can find him on Twitter at **@csharpfritz** and can reach him at **jeff.fritz@telerik.com**.