143 Brooks Street, Suite 2
Brighton, MA 02135
Phone: 1-866-886-0452, Fax: 1-617-249-2116
info@telerik.com

The telerik r.a.d.controls Suite: Above and Beyond

Buying a suite of controls for ASP.NET is a big decision. You need to consider not just the initial expense of buying the controls, but the long-range impact of taking on a particular vendor as your partner in ASP.NET development. Does the vendor have a well-rounded suite of controls? Do they support features that are useful? Do they substantially improve on and surpass the controls that Microsoft ships as part of ASP.NET?

In this paper, you'll get a high-level overview of one of the premier ASP.NET suites, the **t**elerik r.a.d.**controls** suite. This set of 17 controls empowers you to deliver feature-rich, standards-compliant, and cross-browser compatible Web applications.

The telerik Way

As you work with the r.a.d.**controls**, you'll discover that they all have some features in common. This consistency helps you make better use of the suite and makes the controls work better together. Here are some of the key features shared across the various controls:

- Separate versions for ASP.NET 1.x and ASP.NET 2.0. The r.a.d.**controls** work in all versions of ASP.NET, but having separate versions allows them to take advantage of the new 2.0 features (such as declarative databinding and theme support) without dragging down the 1.x codebase.
- Extensive support for client-side programming. The r.a.d.**controls** come with a documented client-side API for fast and responsive Web sites. In addition, they fully support Microsoft's "Atlas" AJAX implementation for richly interactive sites.
- Standards compliance, including XHTML 1.1 support and support for W3C and Section 508 accessibility guidelines
- Extensive help and samples, including source code for all samples

Above: telerik Controls That Do More Than Their ASP.NET Counterparts

It's convenient to divide the r.a.d.**controls** suite into two sets of controls. The first set of controls superficially resembles various stock ASP.NET controls from Microsoft, but rise above them in functionality and flexibility:

- r.a.d.**calendar**
- r.a.d.**combobox**
- r.a.d.**dock**
- r.a.d.**grid**
- r.a.d.**input**
- r.a.d.**menu**
- r.a.d.**panelbar**
- r.a.d.**rotator**
- r.a.d.**treeview**

143 Brooks Street, Suite 2
Brighton, MA 02135
Phone: 1-866-886-0452, Fax: 1-617-249-2116
info@telerik.com

- r.a.d.**upload**

The r.a.d.calendar Control

The r.a.d.**calendar** control is an attractive and high-performance replacement for the standard ASP.NET calendar control. With superior customization and support options for client-side operation, r.a.d.**calendar** offers an easy way to improve the user experience anywhere that your application requires date selection.

The r.a.d.**calendar** gives you both visual and functional enhancements over its ASP.NET counterpart. On the visual side, r.a.d.**calendar** fully supports customization for different parts of the control, such as Title, WeekendDay, Today, and so on, as well as ASP.NET 2.0 themes and skinning. (Nine attractive pre-built themes ship with the control.) Beyond this level of customization (which matches what you get from the standard calendar control), r.a.d.**calendar** lets you define a collection of special days that get their own styles, display more than one month at a time, show weeks vertically rather than horizontally, and use header and footer templates to dress up the control.

On the functional side, r.a.d.**calendar** adds several features not seen in the standard control:

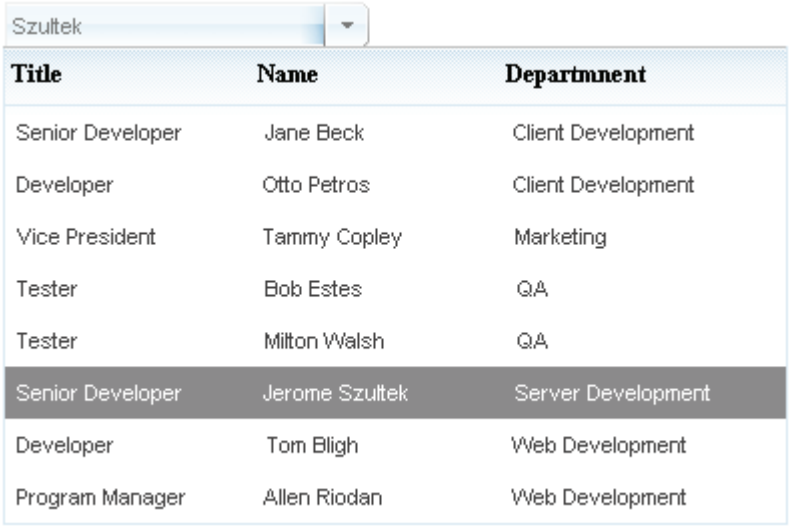
- Multiple-month fast navigation
- Month and year navigation from a drop-down menu activated by clicking on the month title
- A DatePicker control, which couples a TextBox with a popup calendar control

The most important extended features, though, are the choice of operating mode and the support for a client-side API. You can choose to run the entire control in callback or AJAX mode, running much of the logic on the client to avoid page postbacks and thus providing a smoother user experience. This feature is completely absent in the ASP.NET calendar control.

143 Brooks Street, Suite 2
Brighton, MA 02135
Phone: 1-866-886-0452, Fax: 1-617-249-2116
info@telerik.com

The ASP.NET calendar also has no client-side API. The r.a.d.**calendar** has a client-side API that lets you select and unselect dates, or find out what dates the user has selected, using javascript. The API also raises events when the control state changes, giving you a javascript hook for updating any dependent controls.

The r.a.d.combobox Control



Selecting from a list is one of the most common user interface conventions in all applications, Windows and Web alike. The ASP.NET DropDownList control gives you a very basic version of this convention by providing a thin wrapper around the standard HTML select control. If you need anything beyond the most basic options, you'll need to turn to a more full-featured alternative, such as the r.a.d.**combobox**.

The ASP.NET DropDownList control allows the user to select one item from a simple list of alternatives. A user can click on the control to see the list and click again to select the item. The keyboard can also be used to choose an item.

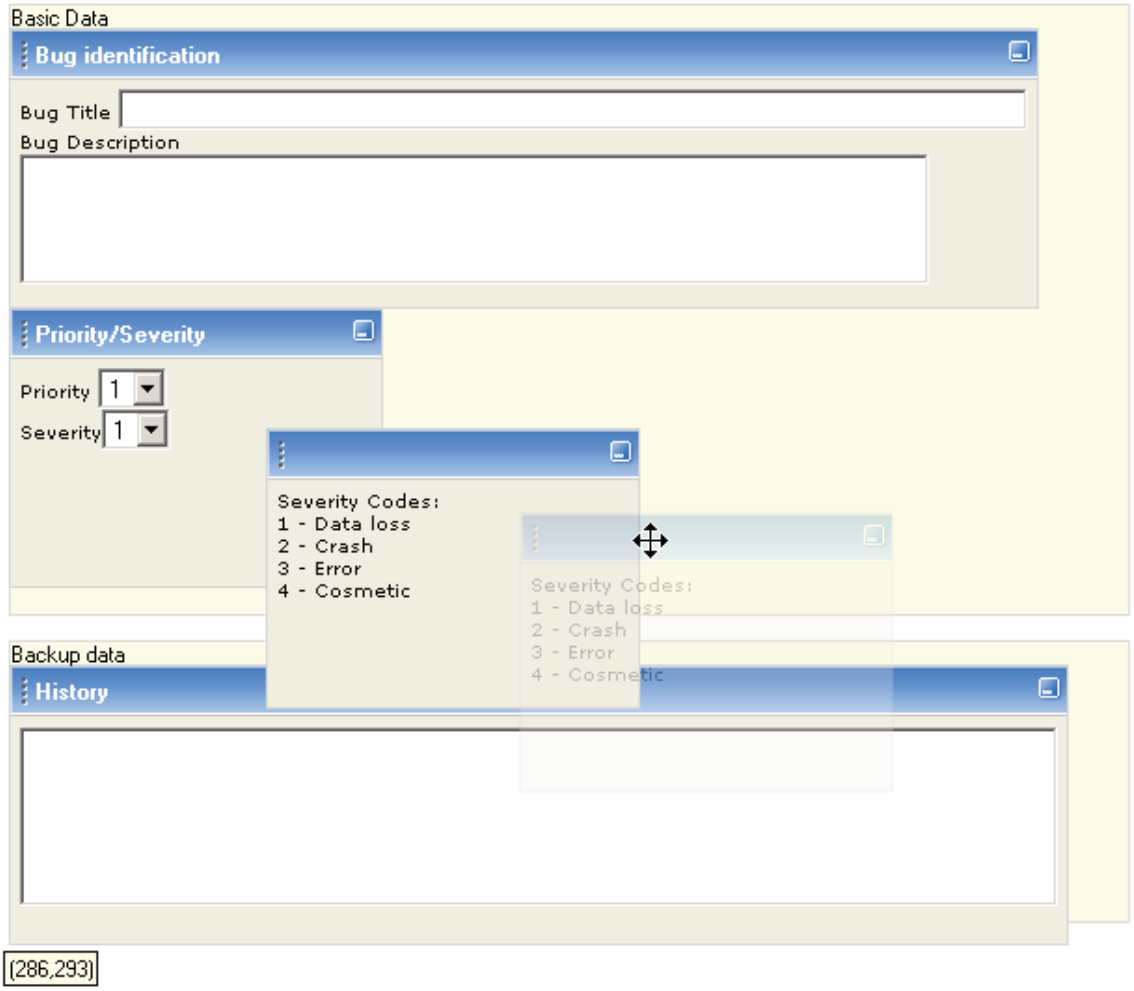
The r.a.d.**combobox** can work similarly to the ASP.NET control if configured to do so, but has many more options that increase your application's flexibility. First off, it can be configured to allow a user to enter items not in the list, or even to allow entering multiple items (separated by special characters such as commas or semicolons). But the real treat here is in the drop down portion of the control. Instead of a simple single column list, the r.a.d.**combobox** control allows multiple columns of data to be shown. In fact, because the drop down is a templated control, you can place anything you like there, including graphics or hyperlinks. A separate template allows you to add column headers to the drop down area.

Another area where the r.a.d.**combobox** far surpasses the ASP.NET DropDownList is in styling and effects. With the DropDownList, you get a simple "one style fits all" approach. With the r.a.d.**combobox**, you'll find full support for preset styles, as well as

143 Brooks Street, Suite 2
Brighton, MA 02135
Phone: 1-866-886-0452, Fax: 1-617-249-2116
info@telerik.com

full customization for the look and feel of the drop down area. You even get to select from among special effects, such as checkboard or wipe, for displaying the drop down. In addition to AJAX support, the r.a.d.**combobox** provides a client-side javascript API that includes autocomplete as well as a number of useful events. It fully supports databinding to arrays, collections, DataTables, and DataSets, as well as complete programmatic operation using resources embedded directly in your ASP.NET pages. By supporting a true superset of DropDownList functionality, r.a.d.**combobox** can grow with you as quickly as you need to go beyond the stock control's limited functionality.

The r.a.d.dock Control



Starting with the release of SharePoint, Microsoft has been working toward a standard way of creating user-customizable Web pages through the use of WebParts. WebParts are regions of HTML that can be dragged around and docked in specific "zones" on the underlying page. **t**elerik's r.a.d.**dock** product is aimed at developers and users who have become frustrated by the limitations of the standard WebParts interface and who are looking for something more customizable.

Although WebParts provide personalization and customizability to Web pages, the interface that they use is somewhat intrusive. WebParts are strongly modal by nature. If you want to move a WebPart around, you need to use a menu to specify where to move it,

143 Brooks Street, Suite 2
Brighton, MA 02135
Phone: 1-866-886-0452, Fax: 1-617-249-2116
info@telerik.com

or put the whole page into a special "moving parts" state that temporarily exposes drag-and-drop functionality.

By contrast, when you use r.a.d.**dock,** the moveable regions of your Web page (represented by RadDockableObject controls) are always active. You build a page out of a collection of these controls and corresponding RadDockingZone objects, and at either run time or design time, you can simply drag the object controls between the zones. You can also (depending on how you have the properties of the RadDockableObject controls set) leave them floating free, roll them up so only their title bar is showing, pin them in particular spots, or resize them. Docking zones can be resizable too. It's also marvelously easy to save and load the state of a page containing the r.a.d.**dock** controls (one line of code each), so you can let people store their own site personalizations for later re-use.

Everything about the r.a.d.**dock** objects is customizable. You can control the highlighting that shows users where they can drop things, the appearance of the dockable objects, the position of the dragging grips, whether things can be resized, and so on. You've got programmatic access to expand and collapse objects, reorder them, or even create them dynamically. Compared to using WebParts, you'll find yourself writing far less code to enable the same rich end-user customization functionality if you use r.a.d.**dock**.

The r.a.d.grid Control

The ASP.NET GridView control is a solid starting point for the display of tabular data in a Web application. But, like most of the standard controls, it has room for improvement. The **t**elerik r.a.d.**grid** control brings a variety of advanced capabilities to the grid metaphor without adding unnecessary bulk to your applications.

A key advance of the r.a.d.**grid** is its support for hierarchical data. Not only can the r.a.d.**grid** display an entire hierarchy, with expand and collapse buttons to let you explore

143 Brooks Street, Suite 2
Brighton, MA 02135
Phone: 1-866-886-0452, Fax: 1-617-249-2116
info@telerik.com

the structure of the data, it can also accommodate multiple tables at a single level of hierarchy. This is simply beyond the capabilities of the GridView without extensive custom programming, and is definitely a boon for database applications.

The r.a.d.**grid** is also easy for users to interact with, thanks to its extensive selection of client-side behaviors. With little or no code, you can enable sorting (including sorting on multiple columns), grouping by dragging column headers to a grouping panel, rearranging columns by drag-and-drop, resizing rows and columns with the mouse, and multiple-row selection. A client-side API lets you perform these operations programmatically without server postbacks as well. And when you do need to execute a postback, the user's client-side customizations are all preserved when the page is rebuilt. Like the other r.a.d. controls, you can fully customize the appearance of the r.a.d.**grid**. The full grid is skinnable, and in a hierarchical grid, you can skin each table separately. There's also full design-time support to let you build your grid in the Visual Studio WYSIWYG environment.

Perhaps best of all, it's simple to migrate from the GridView to the r.a.d.**grid**. The syntax of the two controls is largely compatible. Just change a few tags and your GridView becomes a r.a.d.**grid**, ready to take advantage of the enhanced capabilities instantly.

The r.a.d.input Control

The humble TextBox control may not seem like a strong candidate for improvement, but even here, **t**elerik adds value. By using the r.a.d.**input** controls, you can help ensure trouble-free data entry in applications that collect data from the user in particular formats.

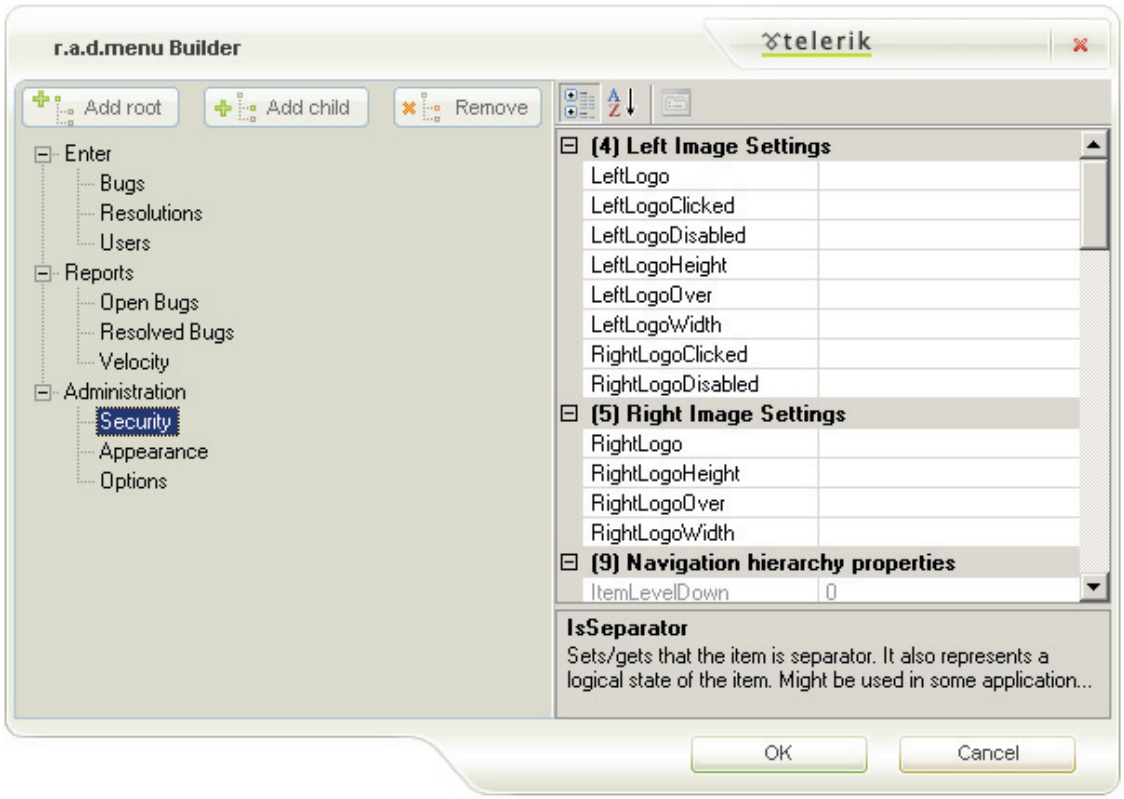
The r.a.d.**input** package contains two controls: the RadMaskedTextBox control and the RadDateInput control. The first offers "masked" data entry. That is, it lets you define a particular pattern of characters, and limits the users' input to that pattern. For example, you might allow only numeric input with dashes in the pattern ###-##-#### if you were collecting social security numbers.

The RadMaskedTextBox control is fully compatible with the ASP.NET validation controls and supports the same keyboard interface as the standard TextBox control. It also lets you define enumerations (for example, a "day of the week" data entry control might accept only the three-letter abbreviations Sun, Mon, Tue, Wed, Thu, Fri, Sat) and lets you move through them using the up and down arrow keys.

The RadDateInput control is even more specialized. It's designed solely for date input, and lets you set both the culture and format for the dates that it accepts. You can also easily synchronize a RadDateInput control with a r.a.d.**calendar** control so that the two always reflect the same date.

143 Brooks Street, Suite 2
Brighton, MA 02135
Phone: 1-866-886-0452, Fax: 1-617-249-2116
info@telerik.com

The r.a.d.menu Control



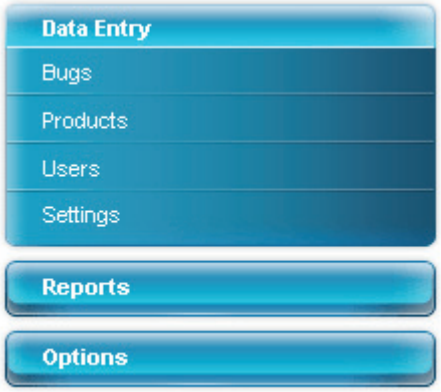
With ASP.NET 2.0, Microsoft includes a menu control in the box to construct menus for Web pages. The **t**elerik r.a.d.**menu** control is a more powerful version of this control that brings the full power of desktop menus to your Web applications. In addition to a more flexible set of appearance options, r.a.d.**menu** offers greater power to the user and an advanced client-side API to avoid unnecessary postbacks and improve application speed.

When you're designing menus, you'll find that the r.a.d.menu designer gives you true freedom. You can create a menu that looks and behaves almost exactly like a standard Windows desktop application menu, right down to icons, shortcut keys, and support for arrow-key navigation. Alternatively you can use r.a.d.**menu** to build context menus for Web pages, or to build vertical or horizontal menus that have a more organic, panel-oriented design. You can customize the appearance of all parts of the menu using themes, CSS, and images, with a much finer degree of control than the standard menu control offers.

When you're ready to hook up the menu, you also gain flexibility with r.a.d.**menu**. This includes support for shortcut keys (such as Ctrl+C for copy), access keys (such as Alt+V to open the View menu) and the arrow keys to navigate the menu structure. You can set the menu to open only when clicked, or when the mouse moves over a menu items. There's also a client-side API that lets you run code in response to clicks and other menu events without round trips to the server, as well as support for Atlas and r.a.d.**callback** for superior performance.

143 Brooks Street, Suite 2
Brighton, MA 02135
Phone: 1-866-886-0452, Fax: 1-617-249-2116
info@telerik.com

The r.a.d.panelbar Control



The Internet has developed its own metaphors for conveying information. One of these is the collapsible side menu for navigation, where headings expand and collapse to reveal subheadings that are hyperlinks to pages within the site. Microsoft's implementation of this metaphor in ASP.NET 2.0 uses the TreeView control, which can be bound to an XML SiteMap to give you a visual representation of your site's structure. This is a functional approach to the problem, but it is difficult to customize or make attractive.

The **t**elerik r.a.d.**panelbar** control frees you from the limits of the TreeView as a navigation control by providing a component designed specifically for collapsible side menus. Implemented as a series of root and child items, the r.a.d.**panelbar** can be expanded and collapsed through either the mouse or keyboard for speed of use regardless of the user's preferences. Like the TreeView, the items in the r.a.d.**panelbar** can be simple hyperlinks, but you can also use content templates to nest any other control you like within an item, including other r.a.d.**panelbar** controls for multi-level menu structures. This gives you the ability to use the control for any situation where you want to hide and display content, not just for menu structures.

Customizable appearance is also a strong point of the r.a.d.**panelbar**. You can use themes and CSS to completely change the appearance of the r.a.d.**panelbar** getting anything from an Outlook-style sidebar to a Windows menu look and feel with the change of a few properties. Client- and server-side APIs allow you to populate and interact with the r.a.d.**panelbar** on the fly, enabling complex scenarios with programmatic menus that depend on other page content.

The r.a.d.rotator Control

Microsoft's AdRotator is basically unchanged from ASP.NET 1.0. The AdRotator is designed to display one out of a selection of banner advertisements pulled from an XML file or other data source. It can accommodate an image, alternate text, and link for each ad—and that's it.

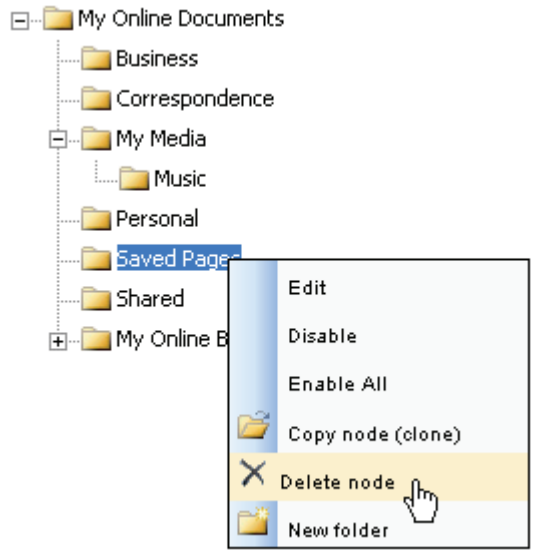
For a truly flexible rotator control, look to **t**elerik's r.a.d.**rotator**. You can use r.a.d.**rotator** for simple banner ad rotation if you wish, but you can also do much more with it. For example, because the contents of the r.a.d.**rotator** can be customized with CSS, templates, and images, you can display nearly anything in rotation: ads, slides, notices, calendar pages, maps—whatever you can build from HTML or ASP.NET

143 Brooks Street, Suite 2
Brighton, MA 02135
Phone: 1-866-886-0452, Fax: 1-617-249-2116
info@telerik.com

controls is fair game. There's also built-in support for "ticker" display of text, making it easy to build an interface that simulates typing news or bulletins one character at a time. This is a great way to draw attention to sales and other important notices on your Web site.

A full client-side API makes it easy to interact with the user in real time. You can slow down, speed up, or stop the rotation when the mouse passes over the rotator, for example, or manipulate the state of the control one frame at a time. A server-side ItemClick event makes it easy to process mouse clicks on ads.

The r.a.d.treeview Control



Microsoft includes a TreeView Web server control with ASP.NET 2.0. This control can display hierarchical data in the form of a tree, and it handles expanding and collapsing the nodes of the tree. Clicking nodes can raise selection events or hyperlink to pages within the Web site. The TreeView control can also be customized with themes and node images to achieve a variety of attractive looks. What could be done to improve on this?

The **t**elerik answer is simple: the r.a.d.**treeview** control encompasses all of the visual functionality of the built-in control and goes on to add rich client interactivity as well. If your users expect a treeview to display the complex behavior normally found in desktop applications, this is the control for you. In addition to a comprehensive set of visual customization properties, the r.a.d.**treeview** adds numerous facilities for user interaction.

- You can load data on demand as nodes are expanded, using either client-side or server-side code, to speed up initial load times and keep the page response snappy.
- You can enable drag-and-drop between r.a.d.**treeview** controls, or from the r.a.d.**treeview** to any other HTML element on the page.
- You can add custom context (right-click) menus to any node in the treeview.
- You can allow the user to edit the text of nodes by clicking a selected node or pressing F2.

143 Brooks Street, Suite 2
Brighton, MA 02135
Phone: 1-866-886-0452, Fax: 1-617-249-2116
info@telerik.com

- You can interact with node selection and checkbox settings using a comprehensive client-side API, as well as the traditional server-side API provided by ASP.NET.

In addition to this all-new interactivity, the r.a.d.**treeview** provides finer control over the look and feel of the treeview than does its ASP.NET analog. For example, each node can be assigned separate CSS classes for the default, hover, and selected states. You can even set the treeview to display in combobox mode, where the tree is not visible until a drop-down arrow is clicked—ideal for displaying hierarchical data on a page where screen real estate is scarce.

The r.a.d.upload Controll

ASP.NET 2.0 includes the FileUpload control, which provides a thin wrapper around the HTML InputFile control. Although FileUpload is easy for .NET developers to use, **t**elerik's r.a.d.**upload** control expands the upload functionality to make uploading files easier than ever.

In contrast to the FileUpload control, the r.a.d.**upload** control allows the user to specify multiple files for uploading in a single operation. The developer can limit the number of files, or can allow this number to be increased dynamically at runtime. This helps keep your application's user interface clean even when you need to accommodate the transmission of multiple files.

The r.a.d.**upload** control also adds significant file validation ability. By setting a few simple properties, the r.a.d.**upload** control lets you limit the uploadable file extensions, MIME types, and file sizes. In addition, these properties can be quickly integrated with the ASP.NET custom validator control for client-side validation.

The **t**elerik control also optimizes the uploading process. The standard ASP.NET FileUpload control stores the entire file in memory, which can cause problems on the server if the user chooses a large file. The r.a.d.**upload** control alleviates stress on the server memory by processing the uploaded files in variably sized chunks. The user experience during uploading is also better, because r.a.d.**upload** provides a special component called the Upload Progress Area. Using this component, a developer can pass back upload progress information on the page while the upload occurs using AJAX technologies. This adds a very professional look to any uploading process, especially when larger file sizes are involved.

Once the upload is complete, you're not quite finished. The file or files have to be stored or processed. The ASP.NET FileUpload control offers only manual methods for storing the file with code. Although r.a.d.**upload** is compatible with this approach, it also offers automatic file processing options. If you set the TargetFolder property, files are automatically saved without manual coding. You also have complete control over whether existing files should be overwritten.

143 Brooks Street, Suite 2
Brighton, MA 02135
Phone: 1-866-886-0452, Fax: 1-617-249-2116
info@telerik.com

By paying attention to these little "fit and finish" issues, **t**elerik's r.a.d.**upload** takes a simple HTML wrapper and turns it into a truly useful server control that makes your application looks better and saves you development time.

Beyond: The telerik Controls Add New Zing To Your Projects

The second group of controls go beyond those that Microsoft ships to bring new functionality to your ASP.NET projects:

- r.a.d.**ajax**
- r.a.d.**chart**
- r.a.d.**editor**
- r.a.d.**spell**
- r.a.d.**tabstrip**
- r.a.d.**toolbar**
- r.a.d.**window**

The r.a.d.ajax Control

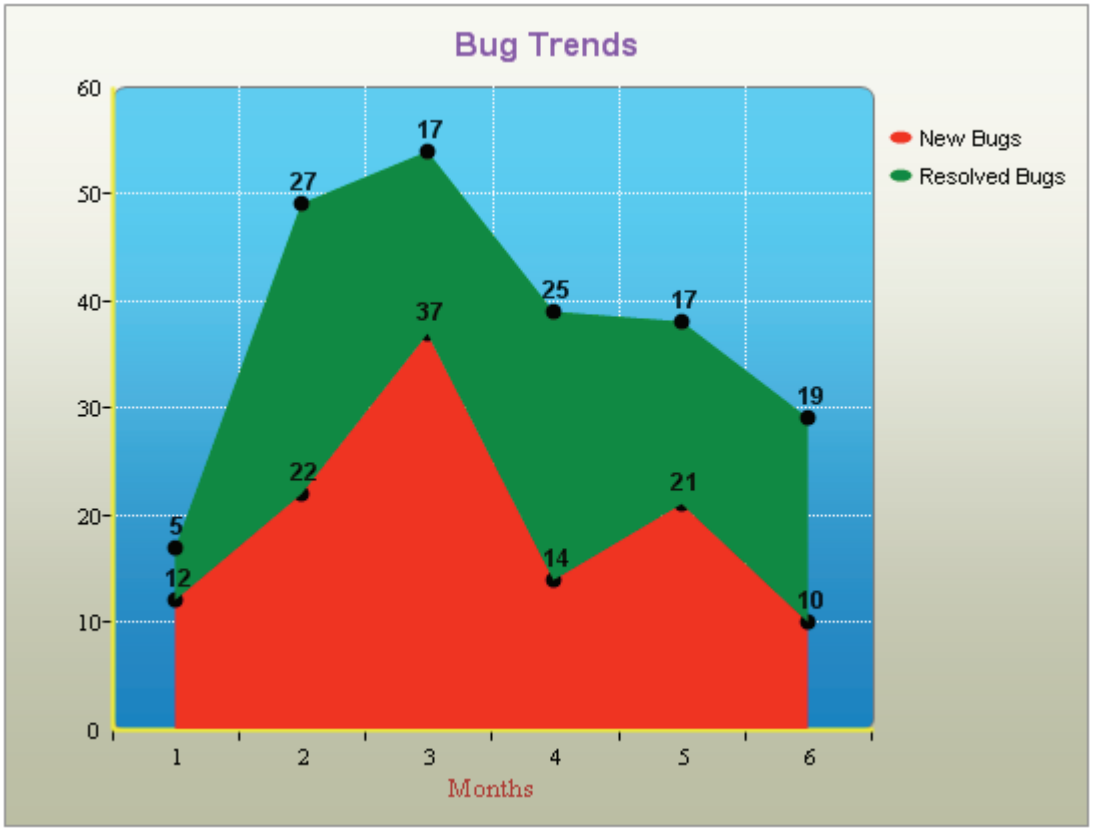
The r.a.d.**ajax** control is **t**elerik's second-generation AJAX product. AJAX, as you may know, stands for Asynchronous Javascript And XML, but it's less important to understand the plumbing than it is to understand what it can do for you. Without r.a.d.**ajax**, executing any code on the Web server requires posting back the entire page and waiting for it to refresh—a time-consuming process that requires a page reload. With r.a.d.**ajax**, you can choose any specific piece of server code corresponding to a single client event to execute, leaving the rest of the page unchanged. The effect is that you can quickly and easily update only a piece of your page—as small as the caption on a single button or as large as the contents of an entire panel containing multiple controls—while leaving the rest of the page unchanged. This gives the effect of a smooth update of only the relevant parts of the user interface in response to some action taken by the user.

The r.a.d.**ajax** control goes beyond AJAX products from other vendors by enabling these smooth updates without requiring you to write a single line of JavaScript code or changing any of your existing server-side code. You can do this simply by placing a RadAjaxManager control on your ASP.NET form along side the existing controls. Telerik supplies a visual editor to let you specify AJAX relations at design time. Just pick which controls initiate the AJAX request, and which controls should be updated - it's that easy.

You also get a set of specialized controls, including a LoadingPanel that gives the user a visual indication of an operation in progress and a Timer that can trigger callbacks at regular intervals. With r.a.d.**ajax**, anyone can take advantage of the latest AJAX techniques without steep learning curves or tedious code rewrites. And, of course, you get the accessibility, XHTML compliance, and cross-browser compatibility benefits of the r.a.d.**controls** product line.

143 Brooks Street, Suite 2
Brighton, MA 02135
Phone: 1-866-886-0452, Fax: 1-617-249-2116
info@telerik.com

The r.a.d.chart Control



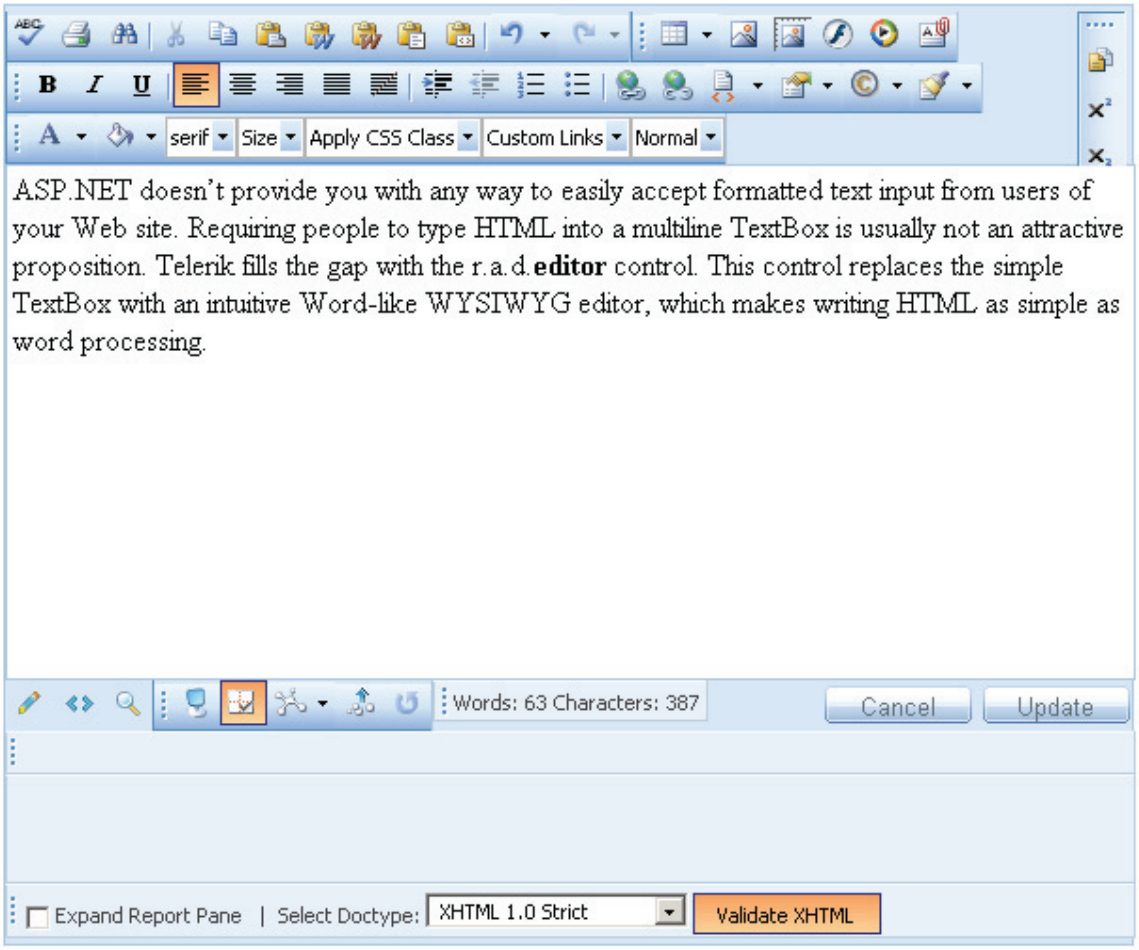
If you're really determined, you can create primitive charts with nothing more than the classes in the System.Drawing namespace and a lot of trial and error. When you need a polished, professional-looking chart with serious options for customization and interactivity, you need to turn to r.a.d.**chart**. The r.a.d.**chart** control supports 13 different chart types, including line charts, pie charts, Gantt charts, bubble charts, and bar charts. You can plot multiple series on a single chart, and use different types for different series (for example, combining a line chart and a bar chart into a single, unified display).

The r.a.d.**chart** control offers extensive customization options to allow you to get just the look you want. You can specify background colors, gradients, and transparency for every chart element, from the background to the data series. You can control the placement, font, and wording of the title and legend, as well as the formatting of numeric values. On pie charts, you can create exploded pieces, and on bar charts you can overlap the bars if you desire. Chart axes can automatically shrink or expand to fit their data, or you can set them to an exact origin, step, and range to control the display of the data more precisely.

Charts can be based on data entered directly through the ASP.NET designer or created programmatically. Alternatively, you can persist a chart to an XML content file or bind it to a database or a live data feed such as an RSS feed. The r.a.d.**chart** is also designed for easy interactivity. Any series can be turned into an image map just by setting a property, which makes doing drill-down charting painless. You can also determine the clicked series, item, or legend item at postback time.

143 Brooks Street, Suite 2
Brighton, MA 02135
Phone: 1-866-886-0452, Fax: 1-617-249-2116
info@telerik.com

The r.a.d.editor Control



ASP.NET doesn't provide you with any way to easily accept formatted text input from users of your Web site. Requiring people to type HTML into a multiline TextBox is usually not an attractive proposition. You'll find that **t**elerik fills the gap with the r.a.d.**editor** control. This control replaces the simple TextBox with an intuitive Word-like WYSIWYG editor, which makes writing HTML as simple as word processing.

In addition to the normal features you expect from a word processor (such as font and size selection, bold, italic, and underline, alignment and indentation options, bullets and numbered lists), r.a.d.**editor** also includes a number of options designed specifically for cleanly editing HTML. For example, you can easily apply CSS styles to text in the editing area, or select custom links from a drop-down list provided by the developer. An extensive table designer provides the user with control over every aspect of HTML tables, and there's also an image manager that handles advanced operations such as creating thumbnails. The r.a.d.**editor** control is also able to strip excess markup from text pasted from Microsoft Word, which can be a serious problem with other Web editors.

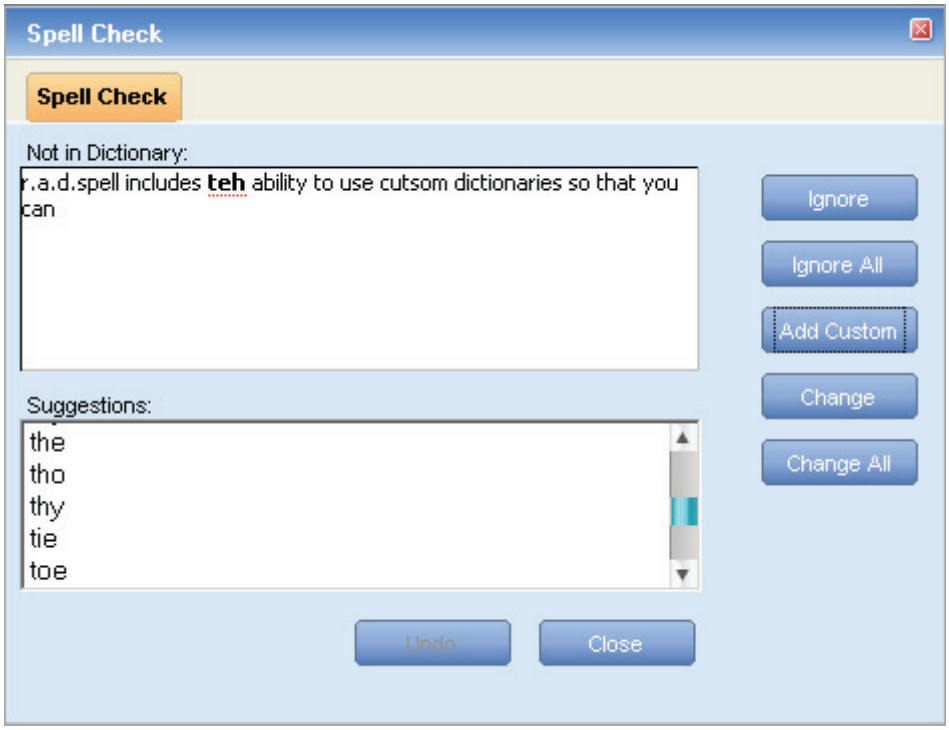
The r.a.d.**editor** control also provides extensive user interface amenities for end users. The editor's toolbars can be floated or docked, so the user can rearrange the workspace to suit their own working habits. At **t**elerik, we've implemented multi-level undo and redo for flexible editing, and context menus and shortcut keys make it easy to perform editing

143 Brooks Street, Suite 2
Brighton, MA 02135
Phone: 1-866-886-0452, Fax: 1-617-249-2116
info@telerik.com

operations without needing to use the toolbars. A full-screen mode makes it possible to temporarily increase the size of the editing area without having the editor take up an excessive amount of screen real estate when it's not in use.

You'll find that **t**elerik has also pioneered advances in programmability to make it easy to integrate r.a.d.**editor** into your own applications. A client-side API and extensive customization options make it easy to change the look and feel of the editor and make it suitable for your own needs. You can use the built-in dialog boxes or create your own for things like the image manager. The r.a.d.**editor** control can even edit text embedded directly in ASPX files so that you can use it to manage things like FAQ pages without creating separate files and a loading mechanism.

The r.a.d.spell Control



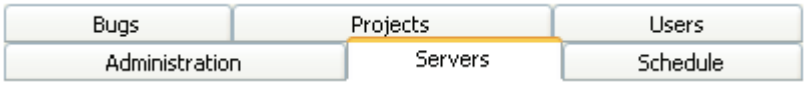
If you're going to let people add content to your site, it's helpful to offer them editing tools. In addition to r.a.d.**editor**, the r.a.d.**controls** suite includes r.a.d.**spell**, a standalone component that you can attach to any editable control. This means that you can use r.a.d.**spell** with simple TextBox controls, but you can also seamlessly integrate it with r.a.d.**editor** (or for that matter, with any other text-editing component).

The r.a.d.**spell** control includes support for 19 languages through standard dictionaries, as well as the ability to import your own custom dictionaries to extend its word list. You can control the algorithm used in suggesting replacement words to suggest more or fewer replacements, as well as to ignore doubled words or words containing numbers or only capital letters. The spelling algorithm is smart enough to ignore inline script and style definitions when you're spell-checking HTML, greatly easing the problem of detecting code as misspelled words. If you have Microsoft Word installed on your Web server, r.a.d.**spell** can also use Word's dictionaries and custom dictionaries.

143 Brooks Street, Suite 2
Brighton, MA 02135
Phone: 1-866-886-0452, Fax: 1-617-249-2116
info@telerik.com

The appearance of the r.a.d.**spell** control itself can be changed by setting its user interface language, and by using CSS-based skins. You can also completely redesign the dialog box used by the control if you like. The control uses a no-postback algorithm for superior performance, letting it complete the checking of an entire control without requiring round trips to the server.

The r.a.d.tabstrip Control



Tabbed interfaces are well-known as an excellent way to organize and optimize a busy user interface. It's hard to see why Microsoft didn't build a tab control into ASP.NET, but **t**elerik's r.a.d.**tabstrip** steps neatly into the gap.

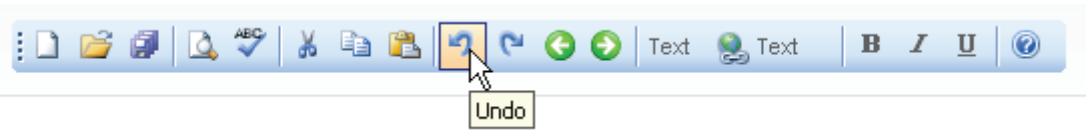
With r.a.d.**tabstrip**, you can create a wide variety of tabbed interfaces. Tabs can be arranged vertically or horizontally, and you can control their appearance through setting text, icons, or skins. You can also build scrolling sets of tabs if you have many tabs to fit into a constrained horizontal space. The tabs themselves represent only the labels and not the associated content; a separate r.a.d.**multipage** control can be hooked up to handle switching the visibility of contained controls in a content area depending on the selected tabs. This makes it easy to design flexible layouts where the tabs are not necessarily directly beside the content area.

There are a number of ways to define the structure of a r.a.d.**tabstrip** control. The simplest is to use inline markup directly in the ASPX file, but that's not the only way. You can also build up the structure programmatically using the server-side API and object model, or encapsulate it in an XML file. Finally, you can use declarative data binding to flat or hierarchical data sources.

A full client-side API lets you handle events including mouse movement and tab selection without server postbacks. You can also enable or disable the entire tabstrip on the client side. On the server side, there's full support for tab click events with auto postback.

Using the r.a.d.**tabstrip** gives you one more way to make your ASP.NET applications stand out from the crowd, and to make them behave a bit more like desktop applications. Your users will appreciate the extra flexibility and you won't have to do a lot of work to deliver it.

The r.a.d.toolbar Control



143 Brooks Street, Suite 2
Brighton, MA 02135
Phone: 1-866-886-0452, Fax: 1-617-249-2116
info@telerik.com

The r.a.d.**toolbar** control lets you add another piece of the standard Windows desktop user interface to your Web applications: the ubiquitous toolbar. A toolbar, of course, presents a rectangular area with a collection of buttons to perform different functions - typically things like open, save, new, and so on. You can have as many buttons as you like on a r.a.d.**toolbar** control, and each button can perform a different action in either client- or server-side code.

It's easy to customize the appearance of the r.a.d.**toolbar**. Of course, you can select the icons for the individual buttons, but you can also control their sizes, and the overall look and feel of the toolbar, by applying CSS-based skins. Pre-built skins are available for common looks such as Office 2000, Office 2003, Visual Studio .NET, and the "glassy" look popularized by MacOS X. If you want to customize things further, individual styles can be applied to such things as the appearance of buttons when the mouse hovers over them and the grab handles used to position the toolbar.

Buttons can be set to display text, images, or both. You can arrange buttons in groups, and assign keyboard shortcuts to them. When you're designing a toolbar, you can embed the description in the ASPX file, use data-binding to any IEnumerable or IListSource, or store the definitions in an XML file. If you combine r.a.d.**toolbar** with r.a.d.**dock**, floating toolbars are easy to achieve.

The r.a.d.window Control

If you've ever tried to show a dialog box in an ASP.NET application, you know that it's not exactly a simple thing to do. You can play with JavaScript alerts or you can open a separate browser window, but you need to worry about things like z-ordering and popup blockers. Alternatively, you can use r.a.d.**window** to create modal or non-modal dialog boxes that behave very much like their desktop application cousins. They're not affected by popup blockers, they can be modal or non-modal, and they work without any code at all.

Creating a new window with r.a.d.**window** is a simple matter of setting properties and then using the RadWindowManager component to show the window. You can show a window when your page loads or in response to a client-side action such as clicking a button. Windows can be as simple as an alert or prompt, or they can display complex

143 Brooks Street, Suite 2
Brighton, MA 02135
Phone: 1-866-886-0452, Fax: 1-617-249-2116
info@telerik.com

HTML. You can also retrieve results from a r.a.d.**window**, just as you can with a Windows dialog box.

The r.a.d.**window** control lets you completely customize the look and feel of individual windows. You can use skins and properties to control the title bar, status bar, icon, and buttons. You have complete control over whether windows can be moved, pinned, maximized, minimized, or closed. It's possible to specify elements on the page that will "catch" minimized windows, and so that a table cell or other element can function like the Windows taskbar. Windows can even be shown with no surrounding elements at all, functioning as timer-based splash screens for your application.

Next Steps

If you'd like to see just how easy it is to build cutting-edge ASP.NET user interfaces with the **t**elerik r.a.d.**controls** suite, we invite you to spend some time working with the controls on your own, without obligation:

- Explore our online demos, which range from simple examples to a full-fledged help desk application and a DotNetNuke Web site. The demos are available for either .NET 1.0 or .NET 2.0, and you can view all of the source code directly in our demo browser.
- Download trial versions of any individual control or the entire suite, as well as our demos. If you prefer, you can download the product help files separately. You can create 30-day trial keys directly from the download page.
- Use our support and community forums to ask any questions you might have about the controls, or contact our friendly and helpful sales department.

When you're ready to make a purchase, a single-developer license to the entire r.a.d.**controls** suite starts at $799 for use on an unlimited number of servers and domains. If you buy five or more licenses, you automatically receive source code to the suite as well.

Mike Gunderloy, Senior Technology Partner, Adaptive Strategy