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

Building a RibbonBar for Your Application

This article will discuss the Ribbon interface and techniques for effectively using a RadRibbonBar in your applications. First, this article will discuss the Ribbon interface: a brief history of the Ribbon, why the Ribbon is an effective interface for your applications, and what elements make up a Ribbon. Next, this article will move onto strategies for creating an effective interface using the Ribbon interface.

What Is the Ribbon?

For Microsoft Office 2007, the UI engineers at Microsoft created a new interface to replace the existing menu structures in most of the Office applications; Outlook still uses the older menu structures in the main Outlook application, but uses the new interface when the user is composing e-mail. This new interface makes it possible to greatly reduce the number of mouse clicks, and was the result of research to make mouse usage more effective. **t**elerik's RadRibbonBar is a third-party control that lets you build your own Ribbon interface.

The RadRibbonBar is actually a single control area that is composed of numerous controls.

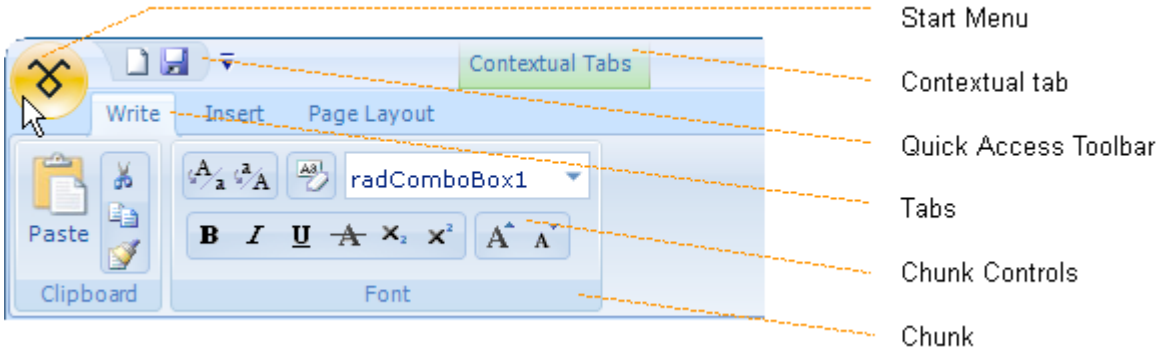

Figure 1. Anatomy of a ribbonbar

In the upper-left corner is the Start Menu, which is similar to the Windows Start menu and is an area for placing the most common options (such as Save or Save As features). The Start Menu can also contain separators, combo boxes, and programmer-created items.

To the right of the Start menu is the Quick Access Toolbar, which typically contains a number of icons that are actually menu items to common features (Save, Undo, or Open are some examples of things you might be able to launch from this area). Like the Start Menu, it can also contain separators, combo boxes, and programmer-created items.

Just below the Quick Access Toolbar is the Tabs. Each Tab opens a set of *Chunks*, which are regions that contain other controls. A developer typically groups related functions within a Chunk. For instance, you may have a Tab group called Write, which contains a Chunk for controlling the font settings of your document. Chunk controls are containers for other controls and can include horizontal button groups, vertical button groups, combo boxes, check-box controls, drop-down buttons, split buttons, repeat buttons, toggle buttons, and check-box buttons. Horizontal button groups and vertical button groups are layout controls for grouping controls giving you the ability to produce a UI with elements of different sizes in rows and columns.

Both drop-down buttons and split buttons display a menu when clicked, but the split button includes an option for selecting a default item whose Click event fires when a user clicks the split button. The repeat button is a special button that repeats the click event as the user holds the

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

mouse down (keeping the button pressed in). The check-box button provides a button that represents two different states by showing a check box next to the option when the item is “on.” In comparison, the toggle button also represents multiple states, but the entire visual presentation of the button can change to indicate the current state.

Techniques for an Effective Ribbon UI using the RadRibbonBar

Designing an effective Ribbon is much like designing a menu with a toolbar. You will go through many of the same steps, but there are also a few additional steps for making an effective interface with a Ribbon.

Before continuing, I will describe the features of a fictional application, and then I will use this fictional application to give you a practical way to see how to apply the concepts. The sample application is a simple blog editor that has the major features of a word processor; it has a spelling checker, formatting features like bold and italic, and features that are related to blog publishing, such as blog categories, tagging, and publishing a post to the blog server.

Step 1: Break down your functions/features with categories

First, you will list the functions and features that you will expose to the end user and place categories next to each one. To illustrate I will use a tabular format with five columns. I find that it is best to list the features and functions first, and then to categorize them after you think you have written down all the categories. I will cover the other columns in the next two steps.

Here is a sample table for some of the features of the blog publishing application:

Feature/Function	**Category**			
Insert Picture	Insert			
Bold text	Style			
Italic text	Style			
Underline text	Style			
Font name	Fonts			
Font size	Fonts			
Spelling Checker	Grammar Tool			
Grammar Checker	Grammar Tool			
Save to file system	File operation			
Publish to blog	File operation			
Manage Blog Categories	Blog Categories			
Apply Blog Category	Blog Categories			
Manage common tags	Tagging			
Apply common tag	Tagging			
Insert custom tag	Tagging			
Font color	Fonts			
Open from local machine	File operation			
Open previously published document	File operation			
Set attachment	Blog Doc Settings			
Insert table	Insert			
Copy	Clipboard			
Paste	Clipboard			
Cut	Clipboard			

Table 1: UI Design Table Step 1

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

You might conclude that the list in Step 1 does not seem organized. This is normal because you will probably forget features as you are listing them, and then remember them later. Do not worry about this. Just list all of the features that you want to expose to the end user.

The next thing you will notice is that there are some seemingly related items that are not in the same category. For instance, Font Name and Insert Table seem related, since they are both formatting functions; however, they are not really so closely related that you would always place them in the same menu or right next to each other on a toolbar every time you designed an interface. Your goal is to create as many categories as possible so that closely related functions are grouped together.

Do not worry if you missed a few features. As you discover things that you have missed, you will know exactly where the item should go in the UI or you will need to add it to the table you are building (depending on where you are in the process).

Step 2: Determine Feature types, and note common functions

In step 2, you will group your categories together into Feature Types. For instance, the Fonts, Insert, and Formatting categories will all end up in the same feature type. These types will ultimately become your tabs in your Ribbon UI.

During this step, you should also note those features that are very commonly used and should be quickly available to the user.

Table 2 shows the sample application with the next two columns filled in.

Feature Name	Category	Feature Type	Common?	
Insert Picture	Insert	Document		
Bold text	Style	Document		
Italicize text	Style	Document		
Underline text	Style	Document		
Font name	Fonts	Document		
Font size	Fonts	Document		
Spell Check	Grammar Tool	Document		
Grammar check	Grammar Tool	Document		
Save to file system	File operation	File	**	
Publish to blog	File operation	File	*	
Manage Blog Categories	Blog Categories	Blog		
Apply Blog Category	Blog Categories	Blog		
Manage common tags	Tagging	Blog		
Apply common tag	Tagging	Blog		
Insert custom tag	Tagging	Blog		
Font color	Fonts	Document		
Open from local machine	File operation	File	**	
Open previously published document	File operation	File	*	
Set attachment	Blog Doc Settings	Blog		
Insert table	Insert	Document		
Copy	Clipboard	Document	**	
Paste	Clipboard	Document	**	
Cut	Clipboard	Document		

Table 2: UI Design Table Step 2

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

I have identified three feature types for this sample application: Blog, Document, and File. I also identified six functions as being commonly used with four of these functions being very commonly used (denoted by double asterisks). I will use this information for the next step (and the final column in the table).

Step 3: Determine where each item will appear in the Ribbon UI

At this point, it should be easy to determine where things go in the interface. You should have a Tab for each feature type and a Chunk for each category. If you have too many items within a Chunk, you can always break down the items in that category into multiple categories. There really is no maximum or minimum number of elements that you will want to put into a tab as the space will expand to the right or down depending on the type of layout controls you are using. Note that all the items in the File feature type appear to be commonly used functions. I will place these items in the Start Menu instead of their own Tab/Chunk structure. Finally, I have identified four items in the sample application as very commonly used functions; I will place buttons for them on the Quick Access Toolbar where they will always be available to the end user. I will put this information into the last column in the table and proceed to the next step.

See Appendix A for the completed table for the sample application.

Step 4: Create Your Interface

Tabs/Chunks

1. From within Visual Studio 2005, create a new Windows Forms project (or use an existing project with either a new Form or an existing one).
2. Drop a standard ImageList onto your form and add standard icons for Copy, Paste, Cut, Bold, Italic, Underline, or any icons you need for your buttons/menus.
3. Next, drag a radRibbonBar onto your form from the toolbox (it will automatically dock to the top of the application window). Point the radRibbonBar's ImageList property to the ImageList you created in step 2.
4. Click Add New Tab, type the heading you want to use, and then click Enter (repeat for each Tab). For the sample application, you will create a Blog Tab and a Document Tab.
5. Click the Blog Tab in the sample application and the Chunk area will open.
6. Create a new Chunk for each category, so for the sample application's Blog Tab there will be three Chunks: Categories, Tagging, and Settings.
7. Click the first Chunk and change its Text property to Categories. (You might also want to change the (name) property to something like CategoriesChunk or ChunkCategories.) Repeat this for the other two Chunks using their respective names. The UI for the form in Visual Studio 2005 should look like this:

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

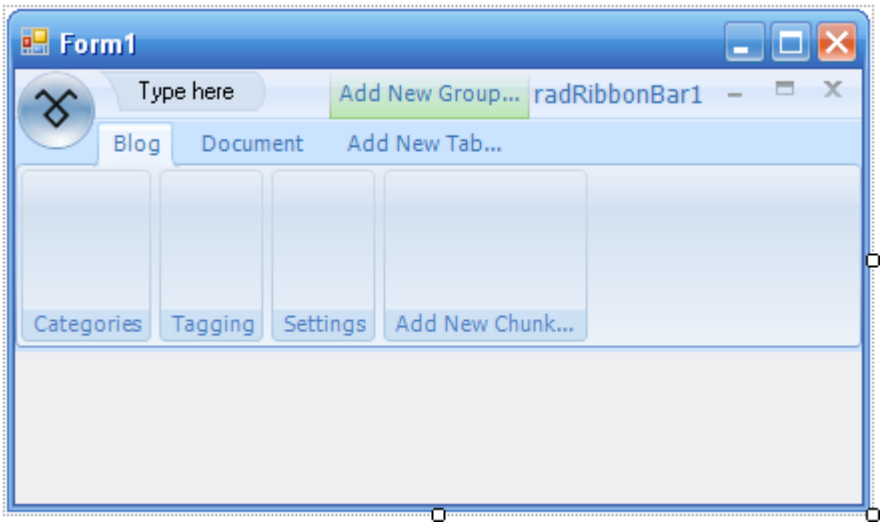

Figure 1: Sample App UI Design in Visual Studio 2005 after step 7

8. Next, add controls to each Chunk. Follow these steps for the Categories Tab for the sample application:
    a. Add a vertical button group by right-clicking the Chunk, selecting the **Add an Item** submenu, and then choosing the **Vertical Button Group** item.
    b. Add the RadButtonElement by right-clicking the vertical button group, selecting the **Edit Items** submenu, and then choosing the RadButtonElement item from the Add menu.
    c. Change the new RadButtonElement's properties as follows:
        i. (name) → ManageCategoriesButton
        ii. (optional) ImageIndex → [the image in your ImageList that applies to this item]
        iii. Text → Manage Categories
        iv. TextAlignment →MiddleRight
        v. TextImageRelation → ImageBeforeText
    d. Click OK.
    e. Add a RadDropDownButton beneath Manage Categories by right-clicking the Manage Categories button, selecting the Insert Adjacent submenu, and then choosing the RadDropDownButton item.
    f. Change the new RadDropDownButton's properties as follows:
        i. (name) → CategoriesDDButton
        ii. (optional) ImageIndex → [the image in your ImageList that applies to this item]
        iii. Text → Categories
        iv. TextAlignment →MiddleRight
        v. TextImageRelation → ImageBeforeText
        vi. NOTE: You could add items to this RadDropDownButton by using the Items property, but this type of feature normally would obtain the list of categories programmatically from a blog server.
        vii. If you get your items out of order you can simply drag one element until it is above or below another element and that item will push the other element over or down to make room for the element you drag/dropped.
    g. At this point your UI will look like this:

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

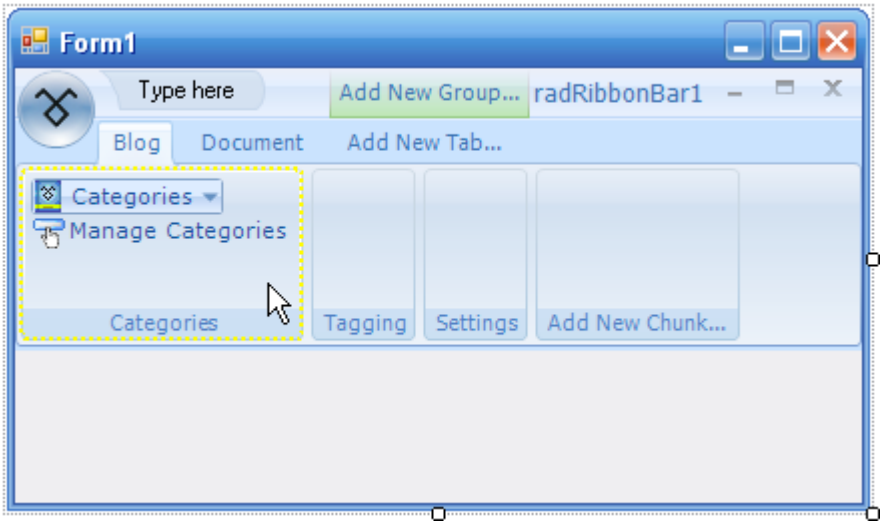

Figure 2: Sample App UI Design in Visual Studio 2005 after step 8

9. Repeat steps 5 through 8 for the other Chunks in the Blog Tab and also repeat for all the Chunks in the Document Tab.

Start Menu/Quick Launch Toolbar

1. Next, you will add items to the Start Menu. Click the Ribbon control and set the StartButton image in the properties grid.
2. Click the StartButton/Start Menu image and add items as you need them. Follow these steps to create the Start Menu for the sample application:
    a. Click Start Menu, select Add New and then select new RadMenuItem in the drop down.
    b. Click the newly created menu option and set its properties in the Properties pane as follows:
        i. (name) → OpenStartMenu
        ii. Text → Open
        iii. Optional: add an image and set the TextImageRelation and TextAlignment properties or set the DisplayStyle property to Text.
    c. Click the Start Menu, hover over the Open menu item, hover over the “Add new” on the right and select the new RadMenuItem from the submenu
    d. Click the new MenuItem you just created and set its properties as follows:
        i. (name) → OpenFileStartMenu
        ii. Text → File
        iii. TextAlignment →MiddleRight
        iv. TextImageRelation → ImageBeforeText
    e. Click the Start Menu, rest on the Open menu item, rest on Add New on the right (below the File item), and then select the new RadMenuItem from the submenu.
    f. Click the new MenuItem you just created and set its properties as follows:
        i. (name) → OpenFromBlogStartMenu
        ii. Text → From Blog
        iii. DisplayStyle → Text
    g. Click the Start Menu and add a new RadMenuItem by resting on the Add New menu that appears just below the Open menu, and then selecting new RadMenuItem from the submenu.
    h. Click the newly created menu option and set its properties in the Properties window as follows:
        i. (name) → SaveStartMenu
        ii. Text → Save
        iii. Optional: add an image and set the TextImageRelation and TextAlignment properties or set the DisplayStyle property to Text.

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

i. Click the Start Menu and add a new RadMenuItem by resting on the Add New menu that appears just below the Save menu, and then selecting new RadMenuItem from the submenu.
j. Click the newly created menu option and set its properties in the Properties window as follows:
    i. (name) → PublishStartMenu
    ii. Text → Publish
    iii. Optional: add an image and set the TextImageRelation and TextAlignment properties or set the DisplayStyle property to Text.
k. Add a shortcut key to the Open File menu by selecting OpenFileStartMenu from the drop-down of the Properties grid, clicking the ellipses in the CommandBinding property, and then setting the shortcut key in the dialog box that appears.
l. Set the OpenFileStartMenu item's Hint property to Ctrl+O. This will give the end user a visual cue as to what the shortcut keys are.
m. Additional steps you might want to take are to assign more shortcut keys to each item in the Start Menu and add text to the Hint property for each item you have added a shortcut key assignment. *You can also use this same method to assign shortcut keys to any of the other controls you have created in previous steps (including Tabs).*

3. Create Quick Launch icons for the commonly used features of the application.
    a. Click in the Quick Launch area where there is text that says "Type here." Type the text: "Open File", and then press Enter. Repeat this for Save File, Copy, and Paste.
    b. Change those areas of text to simple icons.
        i. Click the Open File item and change its properties as follows:
            1. (name) → OpenFileQuickLaunchMenu
            2. ToolTipText → Ctrl+O
            3. ImageIndex → [Pick the icon that you used for creating the OpenFileStartMenu]
            4. DisplayStyle → Image
            5. Note: if the Contextual Tab gets in your way simply Add a new Tab which will move the Contextual Tab to the right (you can always delete the Tab you created later)
    c. Repeat these steps for the other QuickLaunch items.
    d. Your app should look similar to this (in Visual Studio 2005):

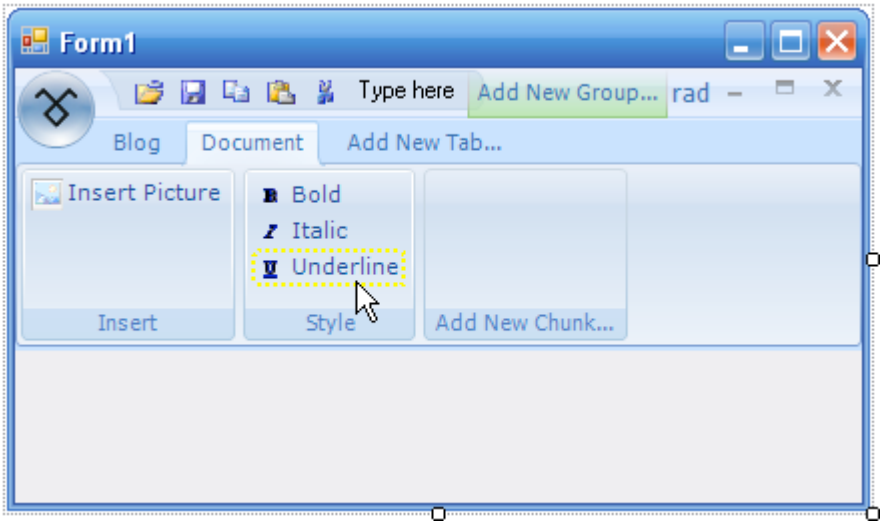

Figure 4: Sample App UI Design after step 3 (Document Tab)

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

Step 5: Show the Interface to an End User

Now that you have created the interface, you should compile it and show it to an end user (or someone else) because what you think is a good organization of your features may not make any sense at all to a user. This really is the most important step of all. You want your UI to make perfect sense to a user because they will ultimately use your application. If something does not make sense to the user, you may want to tweak the placement of your controls to better organize the functions so that all controls are in a place that your end users would logically expect them to be. Ask the person giving you feedback to help you identify where your controls could be better placed.

Dynamic Elements

One of the other things to consider is that you may have Tabs or Chunks that are based on some application state. Building the table previously mentioned will help you to organize the UI and will help you place these dynamic elements of your RadRibbonBar. The easiest solution for making your Ribbon elements to be more dynamic would be to create the complete UI including optional items and simply set the Visibility property to False for those items that rely on some application state. When you need the optional element to appear you can simply set the Visibility property to Visible in your code. The **t**elerik r.a.d.controls for WinForms also includes functionality for dynamically adding and removing elements from the RadRibbonBar's UI also has Contextual Tabs making a number of additional complex scenarios possible.

Summary

Many applications mimic the latest Microsoft Office Suite. With the advent of Office 2007, the new Ribbon user interface will be a control that many users will come to expect. Because of this, you need to be familiar with this new interface.

The most important thing to realize when designing a Ribbon user interface element is that organizing your interface properly is essential for making your application easy for the end user to use. By working through the steps this article has provided, you can use the Telerik RadRibbonBar to build a modern UI in your applications - and one that makes sense to users, not just one that "looks right."

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

Appendix A—The Sample Applications Feature Matrix

Feature Name	Category	Feature Type	Common?	UI Location
Insert Picture	Insert	Document		***Document*** Tab's ***Insert*** Chunk
Bold text	Style	Document		***Document*** Tab's ***Style*** Chunk
Italicize text	Style	Document		***Document*** Tab's ***Style*** Chunk
Underline text	Style	Document		***Document*** Tab's ***Style*** Chunk
Font name	Fonts	Document		***Document*** Tab's ***Style*** Chunk
Font size	Fonts	Document		***Document*** Tab's ***Style*** Chunk
Spell Check	Grammar Tool	Document		***Document*** Tab's ***Style*** Chunk
Grammar check	Grammar Tool	Document		***Document*** Tab's ***Style*** Chunk
Save to file system	File operation	File	**	***Start Menu*** and ***Quick Launch***
Publish to blog	File operation	File	*	***Start Menu***
Manage Blog Categories	Blog Categories	Blog		***Blog*** Tab's ***Categories*** Chunk
Apply Blog Category	Blog Categories	Blog		***Blog*** Tab's ***Categories*** Chunk
Manage common tags	Tagging	Blog		***Blog*** Tab's ***Tagging*** Chunk
Apply common tag	Tagging	Blog		***Blog*** Tab's ***Tagging*** Chunk
Insert custom tag	Tagging	Blog		***Blog*** Tab's ***Tagging*** Chunk
Font color	Fonts	Document		***Document*** Tab's ***Font*** Chunk
Open from local machine	File operation	File	**	***Start Menu*** and ***Quick Launch***
Open previously published doc	File operation	File	*	***Quick Launch***
Set attachment	Blog Doc Settings	Blog		***Blog*** Tab's ***Settings*** Chunk
Insert table	Insert	Document		***Document*** Tab's ***Insert*** Chunk
Copy	Clipboard	Document	**	***Document*** Tab's ***Clipboard*** Chunk and ***Quick Launch***
Paste	Clipboard	Document	**	***Document*** Tab's ***Clipboard*** Chunk and ***Quick Launch***
Cut	Clipboard	Document		***Document*** Tab's ***Clipboard*** Chunk