Chapter 12 – More on Controls

# Objectives

* Using more of the VB controls
* Understanding the common dialog controls
* Creating a splash screen
* The life of a form
* Setting control properties

Notice that it took until chapter 11 before we revisited the GUI – that’s because VB makes user interface design a trivial issue. Furthermore, effective software architecture shouldn’t care about the actual interface – regardless if it is GUI or command line. That decision can be made well after the main functionality of the software is working. This chapter will look at some more of the built in VB.NET forms controls that you can add to your project.

This chapter will also examine the common dialog controls for things such as selecting files, picking fonts, colors and so on. We will also look at how to create a splash screen for your applications. Finally, this chapter will provide lists of the control properties, methods and events that we commonly need to interact with.

# Using More of the Visual Basic Controls

Let's begin by taking a look at more of the controls that are available for use in your VB applications. Some of these are quite simple in what they do, but they provide mechanisms that effectively lock down what a user can do.

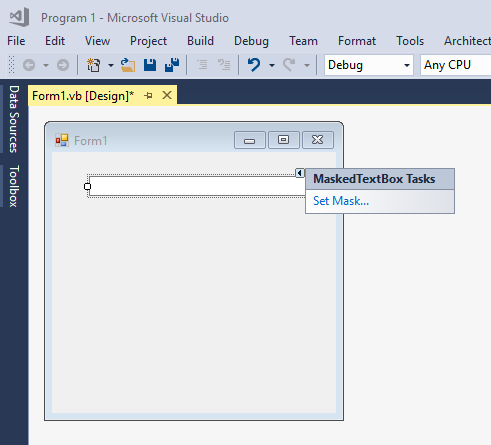
## 

## Masked TextBox Control

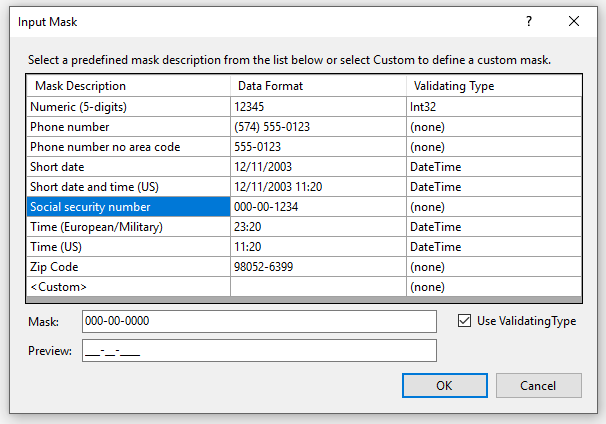
This control is extremely handy in creating a textbox that has an edit mask built in. The edit mask will constrain what your users can type in. Common uses for edit masks are to force telephone numbers, social security numbers, zip codes and dates to be entered a specific way.

Think about a social security number for a moment. If you simply use a regular textbox and ask your user to type in a social security number, they will probably wonder if you wanted hyphens (123-45-6789) or no hyphens (123456789) to be entered along with the numeric information. The Masked TextBox control will remove the ambiguity.

There are several built in masks for common data items that we would want to mask, but you can also add your own. Notice that when you drag and drop the control onto your form’s surface, that there is a small arrow that appears just above and toward the right end of the control. Any control that has this tasks arrow attached to it means that there are built in tasks that can be performed. In this case, if you click on the task arrow, the Masked TextBox task list appears. As you might expect, the Masked TextBox has a task which can set a mask:



Clicking on the Set Mask link, I am then shown the list of built in masks that I can pick from:



Notice that a social security number option appears and if I click on it, I see the mask value itself (000-00-0000) and a preview of what that mask will look like when the user encounters it (\_\_\_-\_\_-\_\_\_\_). This shows the user that the hyphens for the social security number are expected and that they are already in place where they belong. No more confusion as to what to enter/not enter. Let's take a look at the edit mask characters that are available for your use.

List of edit mask characters for Masked Textbox control:

|  |  |
| --- | --- |
| *Symbol* | *Meaning* |
| # | Numeric or space character option. + and – are allowed |
| . | Decimal point treated as a literal |
| , | Thousands separator |
| / | Date separator |
| \ | Escape a mask character turning it into a literal |
| : | Time separator |
| > | Converts all the following characters to uppercase |
| < | Converts all the following characters to lowercase |
| | | Disable the previous case, changing from here forward |
| A | An alphanumeric character is required |
| a | An alphanumeric character is optional |
| C | Character or space is optional |
| L | Letter required: restricted to A-Z and a-z |
| ? | Optional letter |
| & | Required character |
| $ | Currency symbol |
| 0 | Numeric character required |
| 9 | Numeric or space character optional |

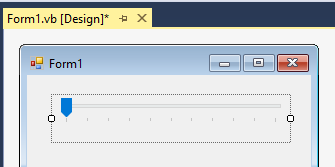
Here’s a list of some of the more useful Masked TextBox properties and methods:

|  |  |
| --- | --- |
| *Property/Method* | *Purpose* |
| AllowPromptAsInput | Allows prompt characters to be typed and moves to the next position in the mask |
| AsciiOnly | Restricts all input to 0-9A-Za-z |
| BeepOnError | If set to true, a beep occurs when invalid input is made – in addition the MaskInputRejected event is raised |
| HidePromptOnLeave | Determines if the user sees the prompt when focus is lost |
| MaskFull | Verifies whether the user entered all the required input |
| PromptChar | Specify a custom prompt character |
| Text | Retrieves the user’s input formatted according to the mask and the TextMaskFormat property |

## TrackBar Control

This control can be added to a project simply by dragging it from the toolbox. The TrackBar control is a slick way of providing similar functionality as the scrollbars, but having the ability to stop at certain detents. Probably the most familiar TrackBar control that users see is the Windows volume adjustment control.

Here’s a screenshot of a TrackBar placed on a form:

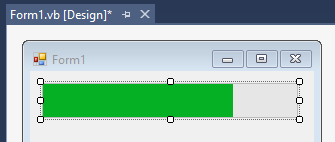


In addition to the typical properties that most controls have, the TrackBar control also shares some of the same properties as the scrollbar controls that we examined a few chapters back. The following list shows other TrackBar properties that are worthwhile to know about:

|  |  |
| --- | --- |
| *Property* | *Purpose* |
| LargeChange | How much a mouse click between the slider and an end will move the slider |
| Maximum | The upper tick value |
| Minimum | The lower tick value |
| Orientation | Should the TrackBar be oriented horizontally or vertically |
| SmallChange | The amount the slider should move when it is clicked on |
| TickFrequency | How many units between each tick mark |
| TickStyle | Describes if and where tick marks are displayed |
| Value | The current value of the slider |

## ProgressBar

The ProgressBar is used to show how some event is proceeding, such as the installation of software. This control is a standard available in the toolbox. Here is an example of the ProgressBar control placed on a form:



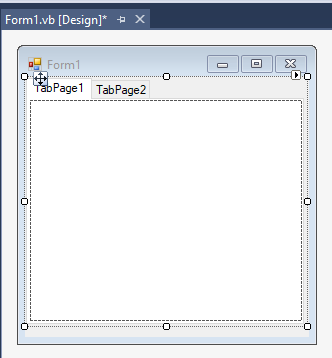
Like the TrackBar control, the ProgressBar control only has a few extra properties that we are interested in:

|  |  |
| --- | --- |
| *Property* | *Purpose* |
| Increment | Advances the progress bar by the value specified |
| Minimum | The smallest value on the left side of the progress bar |
| Maximum | The largest value on the right side of the progress bar |
| PerformStep | Advances the progress bar by a Step value |
| Step | The amount the progress bar advances by each PerformStep call (defaults to 10) |
| Value | The value of the progress bar to be filled in |

### 

## TabControl

The TabControl is a common way to group multiple pages within a single dialog box, like the “Options” settings on many software packages. Here’s a snapshot of the TabControl (with two pages placed in it):

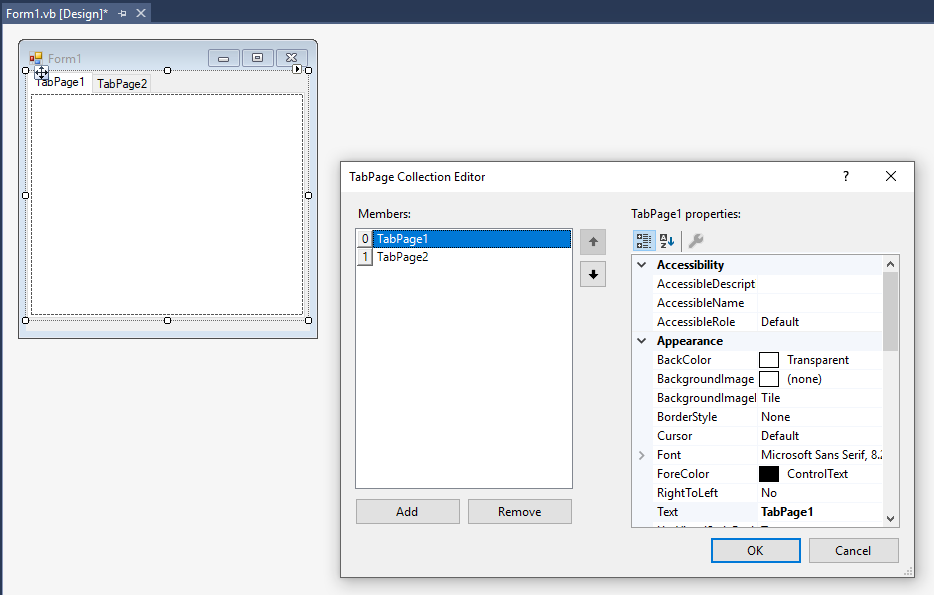


Notice that there is a task arrow on this control, which will quickly allow you to add or remove tabs. Let's take a look at the additional TabControl properties that are of interest to us:

|  |  |
| --- | --- |
| *Property* | *Purpose* |
| SelectedIndex | The index of the selected TabPage |
| SelectedTab | The selected TabPage |
| TabCount | The number of TabPages |
| TabPages | This is the collection that actually holds the tab pages |

Each tab page that you place in the TabControl is like an array index that represents a page. Each page can hold whatever other controls you like, including other TabControls. This is a very handy way to handle pages that need a way to organize a lot of controls that are placed on them, yet use limited real estate on the screen!

As aforementioned, we can add tab pages to a TabControl using the tasks arrow. We can also click on the … button in the TabControl's TabPages property. You will then see a screen that looks like the following:

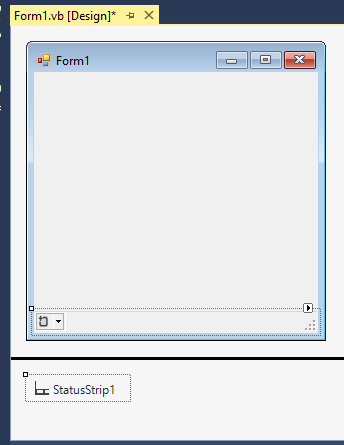


To add a page this way, simply click the Add button and fill in the properties of the page dialog box that is presented to you. The advantage of this technique is that you can modify each page’s properties quickly as you add the pages. This also means that you can make each page look distinct from the other pages if you choose.

## StatusStrip Control

Another control that's common to just about every user, yet one that you might not think of is the StatusStrip control. This is the familiar set of panels that you commonly see at the bottom of an application telling things like whether the Caps Lock is on, or if you are in insert versus overwrite mode, setting the zoom percent of the item being worked on and so forth.

The StatusStrip control is another one found in the toolbox. The next screenshot shows an example of the control added to a form:

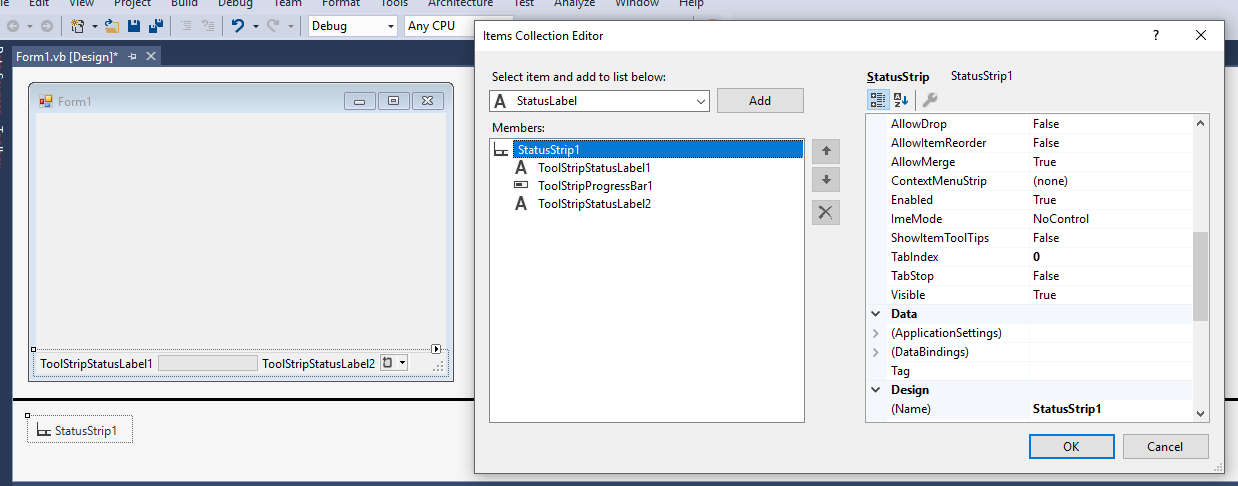


This control shows us the visual placement on the form, but also adds the control to the invisible tray section under the form. You can subdivide the StatusStrip up into multiple panels if you wish. There are four basic types of values which we can place on our StatusStrip (again this can be done through the Tasks arrow, or by clicking on the StatusStrip itself):

* a ToolStripStatusLabel, which displays text, an icon, or both (this is the most commonly used type)
* a ToolStripDropDownButton (like a ComboBox on the StatusStrip)
* a ToolStripSplitButton (allows resizing of portions of the StatusStrip)
* a ToolStripProgressBar (a ProgressBar on the StatusStrip)

In addition to adding the items by the two methods noted above, we can also use the StatusStrip’ property Items … button to edit the collection of items placed on the StatusStrip. The next diagram shows the editor, which looks somewhat similar to what appeared when we carried out the same operation on the TabControl above.

I have added a ToolStripStatusLabel, a ToolStripProgressBar and finally a second ToolStripStatusLabel:



Once you have placed the various pieces that you want on your StatusStrip, you can then click on each one and edit the necessary properties that you want to change. One that you will want to be aware of is the Spring property as it will use up any remaining space on the StatusStrip. I’ve done this to the second ToolStripStatusLabel. Here’s what I did at in Form\_Load:

'Chapter 12 - Program 1

Private Sub Form1\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

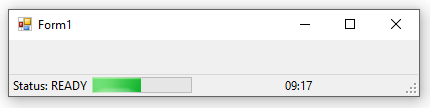
ToolStripStatusLabel1.Text = "Status: READY"

ToolStripProgressBar1.Value = 50

ToolStripStatusLabel2.Text = FormatDateTime(Now(), DateFormat.ShortTime)

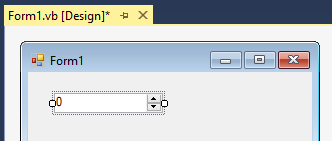
End Sub

This is the form upon execution:



## NumericUpDown Control

The NumericUpDown control simply allows a numeric value to be selected from a series of values by clicking up or down. A common use for this control is in allowing a user to select the number of copies of a document to print.

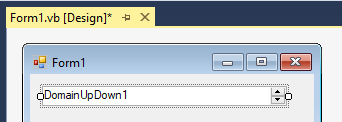


Here is a list of the additional properties that you need to work with the control:

|  |  |
| --- | --- |
| *Property* | *Purpose* |
| Accelerations | Sets how fast the user should be able to scroll up or down if the arrows are held down repeatedly |
| DecimalPlaces | Allows the specification of how many decimal places are to be shown in the control |
| Hexadecimal | Specifies that the values shown should be in hexadecimal |
| Increment | The step size of an up or down press |
| Maximum | The maximum value to display |
| Minimum | The minimum value to display |
| ReadOnly | User is not allowed to enter a value in the textbox on the control |
| ThousandsSeparator | Specifies whether the thousands place separator should be shown in the control |
| Value | The current value chosen |

## DomainUpDown Control

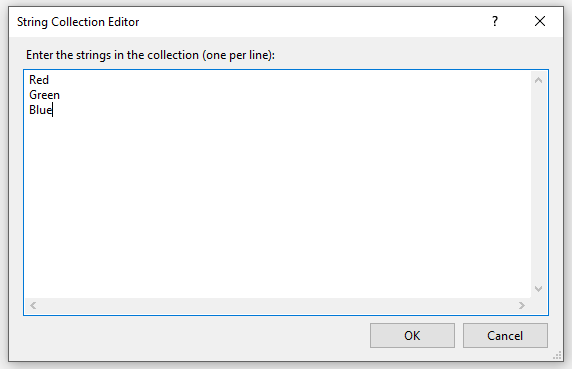
This control is identical to the NumericUpDown control, except that you can put whatever values you want into its Items collection. This control will allow you to use string items rather than purely numeric values:



Here is a list of the additional properties of the DomainUpDown control:

|  |  |
| --- | --- |
| *Property* | *Purpose* |
| Items | The collection of items that you want placed in the picker control |
| ReadOnly | User is not allowed to enter a new value in the textbox associated with the control |
| SelectedIndex | Location of the current selection |
| SelectedItem | The current value selected |
| Sorted | Indicates whether the collection should be sorted or not |

There is an Items collection editor to put in predefined values that make up a domain:

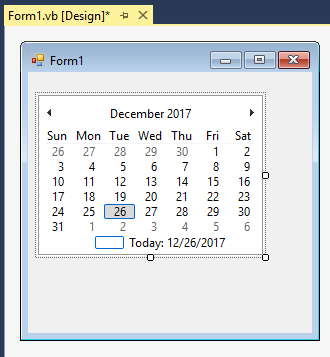


### 

## MonthCalendar Control

The MonthCalendar control provides with you a very easy way to display a calendar. In addition, if you want your user to be able to select dates, this control will also provide that functionality for you too.

There are many things that this control can do, so it is well worth your while to spend some time familiarizing yourself with the functionality of this control. I will just be scratching the surface of the control’s functionality here. Take the time to read the MSDN help on the MonthCalendar control.



This control really is a perfect example of what a control ought to be. Notice from the above screenshot that the control works as soon as it is dropped off on the form. If you were to do nothing more than create a form and add the MonthCalendar control to it, you would be impressed with the functionality that's available when you run the application.

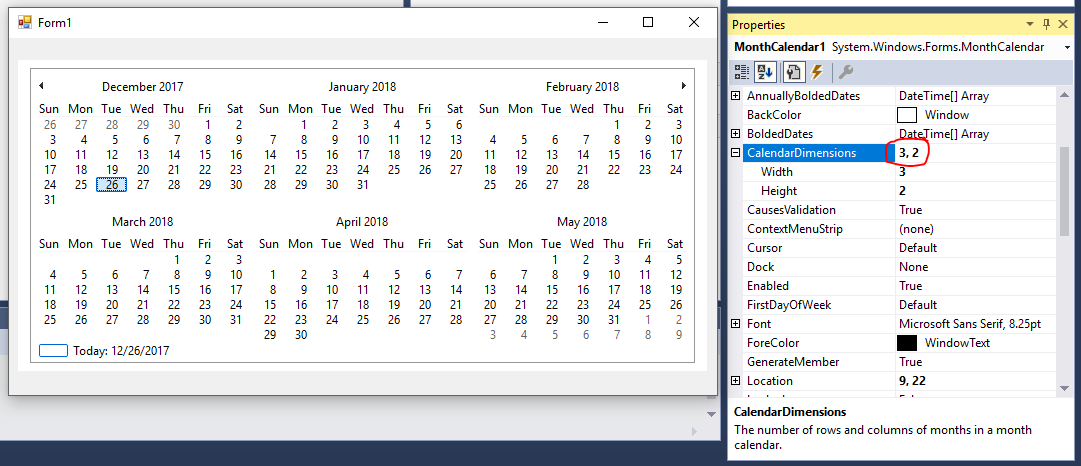
Immediately upon running your program, your user can click the top arrows of the control to move forward or backward an entire month at a time. The current day, according to the system clock, is highlighted on the control. The user can click on the month in the bar at the top of the control to select a different month from the control’s surface which changes to show the twelve months of the current picked year. If the user wants to select a different year, he/she can click on the year in the control's top bar to “zoom out” and select years by the decade and then “zoom in” to get to a particular year.

The only other activity to worry about is what happens when a user clicks on a date on the calendar. This will trigger the DateChanged and DateSelected events on the control that will, in turn, send the selected date through so that you may programmatically receive it and process it in whatever fashion you want.

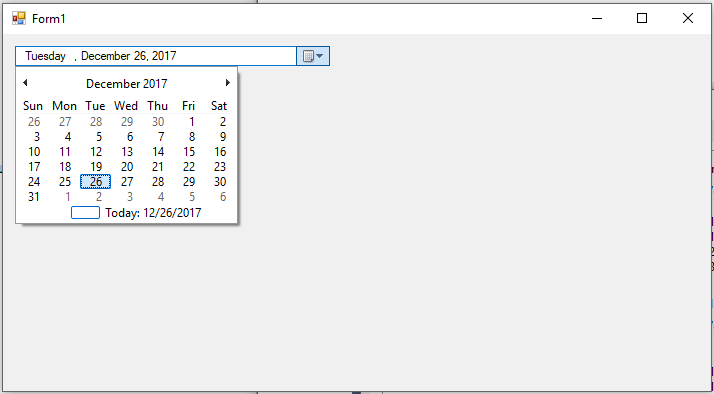
Here are some additional properties of the MonthCalendar control:

|  |  |
| --- | --- |
| *Property* | *Purpose* |
| AnnualBoldedDates | A collection of dates for holidays |
| BoldedDates | A collection of dates that should be bolded |
| CalendarDimensions | Lists the Width and Height of how many months should be displayed at one time on the form |
| FirstDayOfWeek | Allows you to specify which day the week starts on |
| MaxDate | The latest date that can be accessed |
| MaxSelectionCount | Sets the maximum number of dates that can be selected, if multiple selections are allowed |
| MinDate | The earliest date that can be accessed |
| MultiSelect | This property allows multiple dates to be selected |
| SelectionRange | Used to keep track of the dates picked |
| ShowWeekNumbers | Set this property to True to show the week numbers |
| TodaysDate | This has the current date value in it |

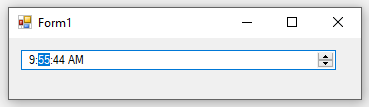
As the next figure shows, you can even set the MonthCalendar control up so that several months are displayed at once. I changed the CalendarDimensions property so that the width was 3 and the height was 2. Think about how easy it would be to create a scheduling system with this control! We’re able to see 6 months with one quick little change…



Before we leave this control, I need to mention the DateTimePicker control which is associated with the MonthCalendar Control. The DateTimePicker is essentially a dropdown ComboBox that interfaces with the MonthCalendar control to allow the section of a date and time, depending on how you format it. The screenshot below shows an example of the default DateTimePicker control. It simply looks like a ComboBox control, but when you click on the open arrow at the right end, a month calendar control is displayed below the ComboBox portion:



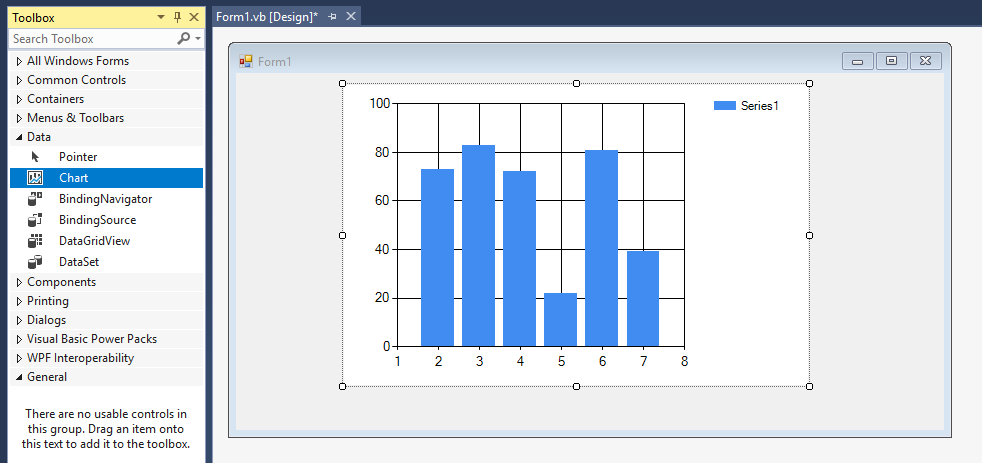
By choosing a different format and switching to an up/down button type control, I can get a completely different look for the DateTimePicker:



### 

## Chart Control

The Chart control is a very convenient way to graph items from within VB.NET. This control is not listed under the “All Windows Controls” section in the toolbox; instead you will have to expand the Data section to be able to find it. If you drag and drop the control over your form, you’ll see something like this, by default:



As its name implies, it is a graph/chart control. Virtually any type of graph you can think of can be created by the control, including 2D bar, 3D bar, pie, line and so on. Data can be sent to this control by entering values in one at a time, using an array or by populating a virtual DataGridView.

Here’s a sample application that shows how to create a data table, fill it with data and then assign it to the Chart control to graph the data. The form for this application is simply a maximized form with a Chart control added to it and stretched to the size of the application. I left the name of the control to its default of Chart1. I also added one Command button so that we could toggle back and forth between 2D and 3D Modes:

Public Class Form1

'Chapter 12 - Program 2

Private Sub Form1\_Load(sender As Object, e As EventArgs) Handles Me.Load

'Create a data table to hold the information -- think of it

'as a virtual spreadsheet

Dim myDataTable As New DataTable

'Here's the data points

With myDataTable

'What's the purpose of each column?

.Columns.Add("Year", GetType(Integer))

.Columns.Add("Cogs", GetType(Single))

.Columns.Add("Sprockets", GetType(Single))

'Here's the actual rows of data...

.Rows.Add(2017, 4000.0, 2000.0)

.Rows.Add(2018, 5000.0, 4000.0)

.Rows.Add(2019, 5000.0, 3500.0)

.Rows.Add(2020, 5300.0, 4000.0)

.Rows.Add(2021, 5500.0, 4200.0)

End With

'Clear out all of the data points on the chart

Chart1.Series.Clear()

'We've got 2 different sets we want to plot

'The first is Cogs sold by Year

'So create and set up the first data series

Chart1.Series.Add("Cogs")

With Chart1.Series(0)

Font = New Font("Arial", 10, FontStyle.Italic)

.BackGradientStyle =

DataVisualization.Charting.GradientStyle.TopBottom

.Color = Color.Blue

.BackSecondaryColor = Color.AliceBlue

.IsValueShownAsLabel = True

.ChartType = DataVisualization.Charting.SeriesChartType.Column

.LabelBackColor = Color.White

.LabelForeColor = Color.Blue

.Points.DataBind(myDataTable.DefaultView, "Year", "Cogs", Nothing)

End With

'Now we need to add the second data series

'which will be Sprockets sold by Year

Chart1.Series.Add("Sprockets")

With Chart1.Series(1)

Font = New Font("Arial", 10, FontStyle.Italic)

.BackGradientStyle =

DataVisualization.Charting.GradientStyle.TopBottom

.Color = Color.Brown

.BackSecondaryColor = Color.SandyBrown

.IsValueShownAsLabel = True

.ChartType = DataVisualization.Charting.SeriesChartType.Column

.LabelBackColor = Color.White

.LabelForeColor = Color.Blue

.Points.DataBind(myDataTable.DefaultView, "Year", "Sprockets",

Nothing)

End With

'Now set up the actual graphing area on the chart

With Chart1.ChartAreas(0)

.AxisX.Interval = 1

.AxisX.LabelStyle.Angle = -90

.AxisX.Title = "Year"

.AxisX.TitleFont = New Font("Arial", 12, FontStyle.Bold)

.AxisX.LabelStyle.Font = New Font("Arial", 10, FontStyle.Bold)

.AxisY.TitleFont = New Font("Arial", 12, FontStyle.Bold)

.AxisY.LabelStyle.Font = New Font("Arial", 10, FontStyle.Bold)

.AxisY.Title = "Units"

End With

End Sub

Private Sub cmdMakeChart3D\_Click(sender As Object, e As EventArgs)

Handles cmdMakeChart3D.Click

'Since we've already got the graph set up, we just need to change it to 3D

With Chart1.ChartAreas(0)

If .Area3DStyle.Enable3D = True Then

.Area3DStyle.Enable3D = False

Else

.Area3DStyle.Enable3D = True

.Area3DStyle.WallWidth = 2

.Area3DStyle.LightStyle =

DataVisualization.Charting.LightStyle.Realistic

.Area3DStyle.Rotation = 25

.Area3DStyle.Inclination = 15

.Area3DStyle.PointGapDepth = 50

.Area3DStyle.Perspective = 5

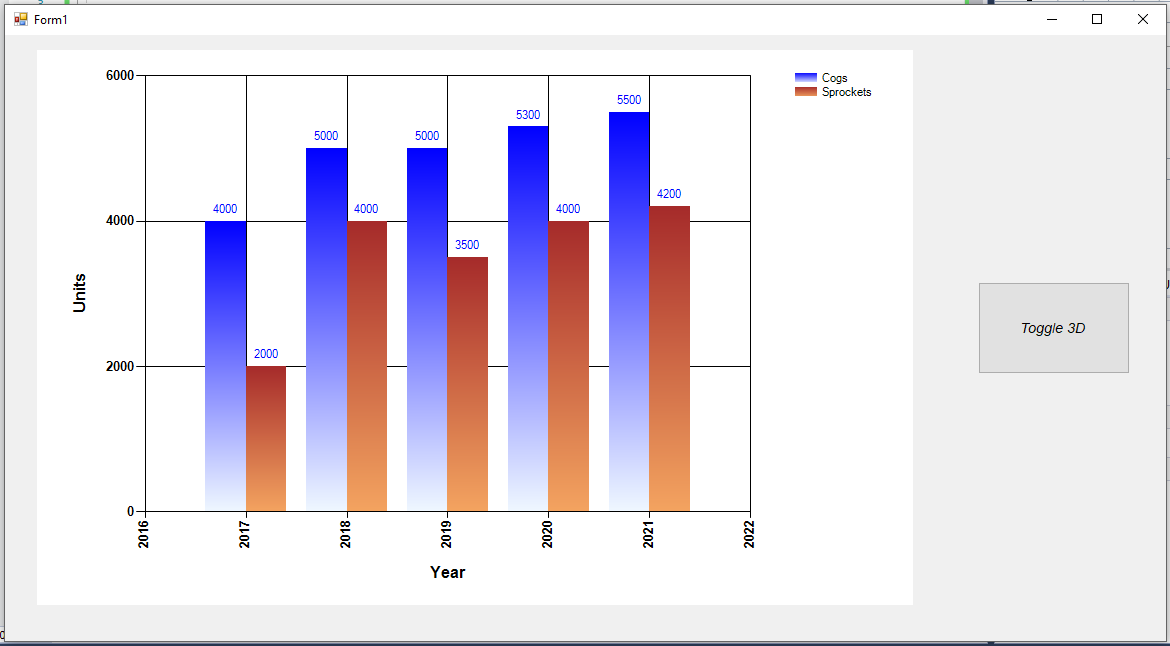
End If

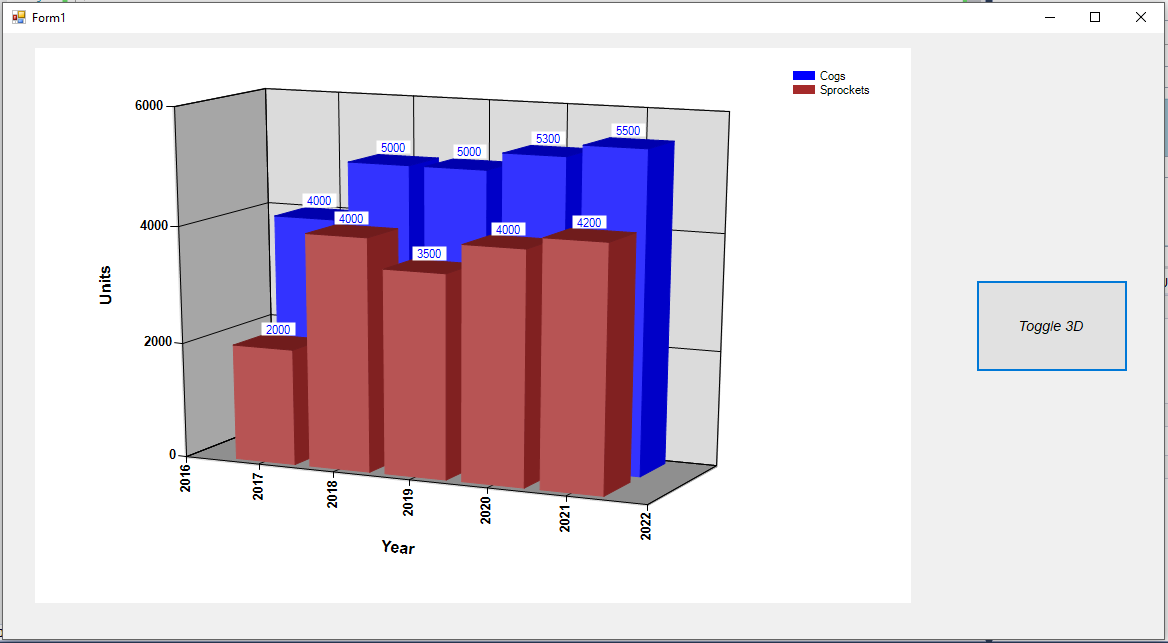
End With

End Sub

End Class

When you run the application and the form loads, your graph will be assembled and displayed. The next screenshots show the chart in both its 2D and 3D forms…





There literally are hundreds of properties, methods and events available on this control. You will need to spend some time researching what they can do for you. There is enough material on this control alone to fill a book. The information that’s in the MSDN library is good if you know what you want to mess with, but it doesn’t actually give you much in the way of example code, which is why I put this demo together. You will find that this is a super control to have access to should you ever need to do any kind of graphing or charting. Again, invest some time learning about it before you need to use it!

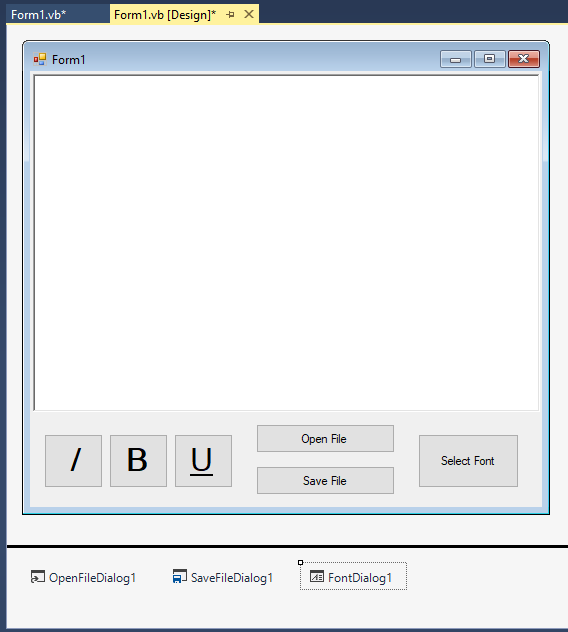
## RichTextBox Control

We already know that the TextBox control can be resized to allow for multiple lines of input and show scrollbars to move through the text. However, what if we want to allow the user to add fonts and other types of goodies to their text? Essentially, this would be like creating the Windows WordPad program.

Fortunately, Microsoft has made our lives quite easy by creating the RichTextBox control. This control is extremely versatile and doesn't require a lot of work in order to use it. The control even has the native ability to load and save Rich Text Format files (.RTF) files in addition to plain text files (.TXT) like Notepad creates.

Let's build an application around the RichTextBox control. We want to create a poorman's WordPad. We know that we can use the RichTextBox control to get most of the functionality. We will add three Command buttons to allow the user to italicize, bold and underline their work as they choose. In addition, let's add buttons that will allow the user to save their work, load their work and choose a different font.

This probably sounds like a lot of work, but once again the .NET framework comes through for us. Create a form as described above with the RichTextBox and 6 command buttons on it. You will also need to add a copy of the OpenFileDialog, SaveFileDialog and FontDialog to the form. The next figure shows what my form looks like:



Granted, we haven't talked much about the OpenFileDialog, SaveFileDialog or FontDialog yet, but you essentially just need to drag them over and you'll be able to follow the logic of the program code.

Here's the application's program code:

'Chapter 12 - Program 3

Public Class Form1

Private Sub cmdBold\_Click(sender As Object, e As EventArgs)

Handles cmdBold.Click

'This is called whenever someone clicks on the bold button

If Not RichTextBox1.SelectionFont Is Nothing Then

'If we have a font, we will need to create a pointer to it

'so that we can modify the bold property

Dim currentFont As Font = RichTextBox1.SelectionFont

Dim newFontStyle As FontStyle

'Now that we have the font pointer, check and see

'if it is bold.

If RichTextBox1.SelectionFont.Bold = True Then

'if it was bold, make it non-bold

newFontStyle = FontStyle.Regular

Else

'if it wasn't bold, make it bold

newFontStyle = FontStyle.Bold

End If

'Now update the current font

RichTextBox1.SelectionFont = New Font(

currentFont.FontFamily,

currentFont.Size,

newFontStyle)

End If

End Sub

Private Sub cmdItalics\_Click(sender As Object, e As EventArgs)

Handles cmdItalics.Click

'This gets called if someone clicks on the italics button

If Not RichTextBox1.SelectionFont Is Nothing Then

'Create a pointer to the font

Dim currentFont As Font = RichTextBox1.SelectionFont

Dim newFontStyle As FontStyle

'Check to see if the font is already italics or not and

'see it to the opposite setting

If RichTextBox1.SelectionFont.Italic = True Then

newFontStyle = FontStyle.Regular

Else

newFontStyle = FontStyle.Italic

End If

'Update the font

RichTextBox1.SelectionFont = New Font(

currentFont.FontFamily,

currentFont.Size,

newFontStyle)

End If

End Sub

Private Sub cmdUnderline\_Click(sender As Object, e As EventArgs)

Handles cmdUnderline.Click

'This is called if someone clicks the underline button

If Not RichTextBox1.SelectionFont Is Nothing Then

Dim currentFont As Font = RichTextBox1.SelectionFont

Dim newFontStyle As FontStyle

'Figure out if we are currently underlined or not and

'set the font up in the opposite manner

If RichTextBox1.SelectionFont.Underline = True Then

newFontStyle = FontStyle.Regular

Else

newFontStyle = FontStyle.Underline

End If

'Update the font

RichTextBox1.SelectionFont = New Font(

currentFont.FontFamily,

currentFont.Size,

newFontStyle)

End If

End Sub

Private Sub cmdOpenFile\_Click(sender As Object, e As EventArgs)

Handles cmdOpenFile.Click

'This is called if the user wants to open a file

'Show the OpenFileDialog box

OpenFileDialog1.ShowDialog()

'Take the filename from the dialog box and use the

'richtextbox's LoadFile method to actually load the file in

RichTextBox1.LoadFile(OpenFileDialog1.FileName)

'NOTE: There was no error trapping performed in the load. We

'assume the user supplied a good filename for a valid RTF file.

End Sub

Private Sub cmdSaveFile\_Click(sender As Object, e As EventArgs)

Handles cmdSaveFile.Click

'This is called if the user wants to save a file

'Show the SaveFileDialog box

SaveFileDialog1.ShowDialog()

'Take the filename from the dialog box and use the

'richtextbox's SaveFile method to actually save the file out

RichTextBox1.SaveFile(SaveFileDialog1.FileName)

'NOTE: There was no error trapping performed in the save. We

'assume that we can write to the location specified by the

'user.

End Sub

Private Sub cmdSelectFont\_Click(sender As Object, e As EventArgs)

Handles cmdSelectFont.Click

'This is called if the user wants to pick a new font

'Show the font selection box

FontDialog1.ShowDialog()

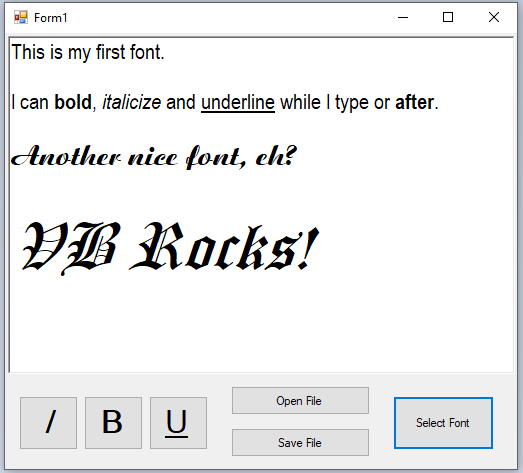
'Take the font that was picked and plug it into our richtextbox

RichTextBox1.SelectionFont = FontDialog1.Font

End Sub

End Class

Here's the application at work. Notice the different fonts in the RichTextBox – you can't do that in a normal multiline textbox. Everything works, including the load and save operations. The interface is a bit clunky, but it wouldn't take much work to make it better.



We'll finish the discussion of the RichTextBox control up by looking at the other properties and methods that you should know about:

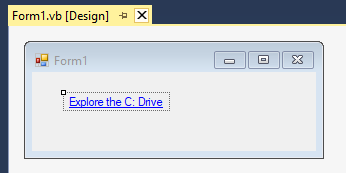
|  |  |
| --- | --- |
| *Property* | *Purpose* |
| Scrollbars | Indicates whether or not scrollbars should be shown |
| SelectionAlignment | Indicates whether text should be left, right or center justified |
| SelectionBullets | Handles bulleted lists in the selection |
| SelectionColor | Color of the text |
| SelectionFontName | Name of the font to use |
| SelectionFontSize | Size of the font to use |
| SelectionHangingIndents  SelectionIndents  SelectionRightIndents | Handles indents within the selection |

Additional RichTextBox control methods:

|  |  |
| --- | --- |
| *Methods* | *Purpose* |
| LoadFile | Loads a file into the RichTextBox |
| SaveFile | Saves the contents of the RichTextBox onto disk |
| Find | Used to search for text within the RichTextBox |

## LinkLabel Control

LinkLabels are a control that appear on forms as underlined labels. When the link label is clicked on, the resource that the label is connected to is displayed (usually a Web page) or executed (another program).



Here’s the code to make the LinkLabel work:

'Chapter 12 - Program 4

Public Class Form1

Private Sub LinkLabel1\_LinkClicked(sender As Object,

e As LinkLabelLinkClickedEventArgs)

Handles LinkLabel1.LinkClicked

'Mark link label as visited

Me.LinkLabel1.LinkVisited = True

'Start up an external program

Process.Start("C:\")

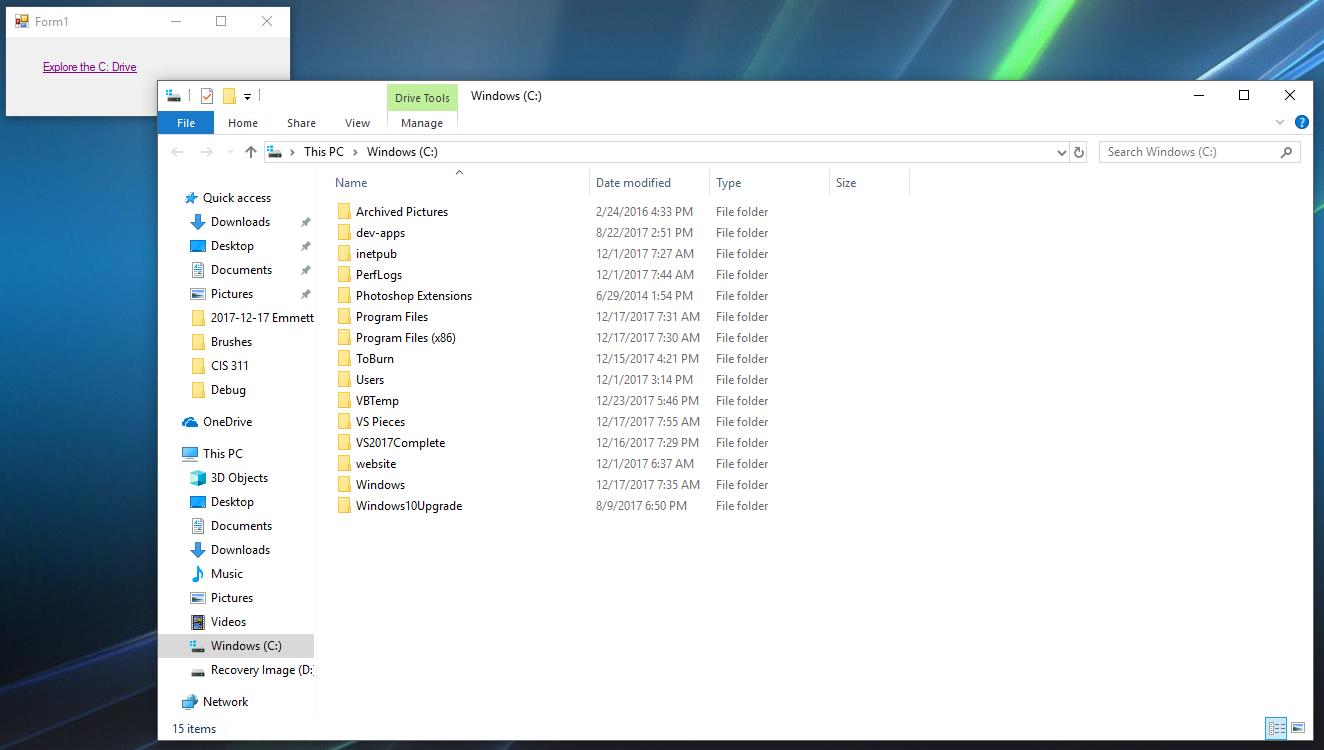
'Could have started up the Internet Explorer and pointed it

'at a site Process.Start("IExplore", "http://www.svsu.edu")

End Sub

End Class

The application at work. Note that the label changes color just like a hyperlink once it’s been visited:



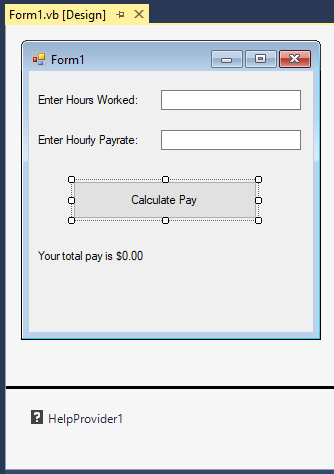
Additional LinkLabel control properties:

|  |  |
| --- | --- |
| *Property* | *Purpose* |
| ActiveLinkColor | Specifies the color of the active link when clicked |
| DisabledLinkColor | Specifies the color of a disabled link |
| LinkArea | Specifies which part of the text is part of the link |
| LinkBehavior | Specifies the behavior of the link, such as when mouseover occurs |
| LinkColor | Color of the link before visited |
| LinkVisited | Returns a Boolean indicating whether or not the link has been visited |
| Text | A link's text |
| VisitedColorLink | Color of the link after being visited |

## HelpProvider Control

The HelpProvider control is another non-visual control that can be used to provide help to your program’s users whenever they press the F1 key. The help can be local, e.g. in the program itself, in a data file on the local machine, or you could point the help to a series of online web pages.

Using the HelpProvider is easy once you drag it over to your application form. Simply decide which controls you are going to provide help for and then set the HelpProvider up to direct users to the appropriate help information if they press F1 on one of those controls which you have decided should provide help. Here’s a simple calculation application with a HelpProvider attached:

  
 I have actually bound the TextBoxes, Command button and last Label to the HelpProvider through code. I didn’t do anything other than drag the controls over and name them. Here’s the code behind what’s going on:

'Chapter 12 - Program 5

Public Class Form1

Private Sub Form1\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

'Bind all of the controls that we want to provide help on to the

'help they are supposed to show...

HelpProvider1.SetHelpString(txtHoursWorked,

"Enter the hours you worked this week." & vbCrLf &

"For example 40 or 29.5")

HelpProvider1.SetShowHelp(txtHoursWorked, True)

HelpProvider1.SetHelpString(txtHourlyPayrate,

"Enter your hourly payrate in dollars and cents." & vbCrLf

& "For example 10.00 or 12.25")

HelpProvider1.SetShowHelp(txtHourlyPayrate, True)

HelpProvider1.SetHelpString(lblTotalPay,

"This line will be updated once you enter your" & vbCrLf &

"hours, pay rate and press the calculate button.")

HelpProvider1.SetShowHelp(lblTotalPay, True)

HelpProvider1.SetHelpString(cmdCalculatePay,

"After entering your hours and pay rate" & vbCrLf &

"press this button to calculate your pay.")

HelpProvider1.SetShowHelp(cmdCalculatePay, True)

End Sub

Private Sub cmdCalculatePay\_Click(sender As Object, e As EventArgs)

Handles cmdCalculatePay.Click

lblTotalPay.Text = "Your Total Pay is : " &

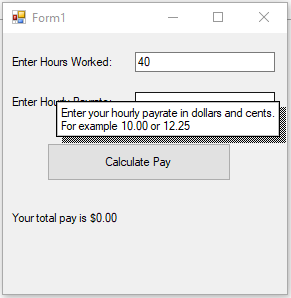
FormatCurrency(CSng(txtHoursWorked.Text) \*

CSng(txtHourlyPayrate.Text))

End Sub

End Class

So assuming I didn’t know what I was supposed to do with the second textbox, I can press F1 when I am sitting on it and some help text appears in the running application:



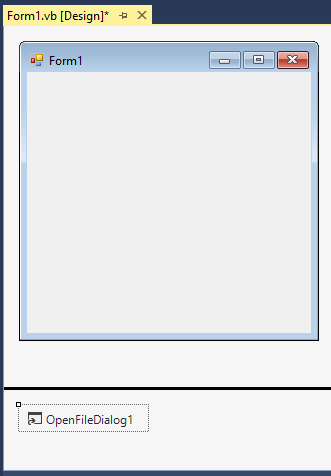
Here are the pertinent HelpProvider methods and properties:

|  |  |
| --- | --- |
| *Item* | *Purpose* |
| HelpNamespace | Sets the name of a help file to associate with the HelpProvider |
| SetHelpNavigator | Specifies the type of help |
| SetHelpString | Associates a help string with a control |
| SetShowHelp | Called for each control to identify that it has help available |

# Understanding the Common Dialog Controls

Have you ever noticed how just about every Windows application has the same dialog boxes pop up for commonly occurring situations such as opening files, saving files, printing, picking fonts and colors? There is a very good reason for this: Microsoft wrote dialogs (called Common Dialog Controls) specifically for these purposes and placed them in the operating system. Rather than having to create these entities yourself, you can take advantage of these tools since they are part of the Visual Studio package – we already put them to use in our RichTextBox demo program.

Visual Basic.NET simplifies life by making each dialog an individual control that's available straight from the toolbar. To use one of the controls, simply drag and drop the control on your form. When you add one of the common dialog controls to your form it’s not terribly exciting since it shows up in the "invisible" control section under the form as shown in the figure below:

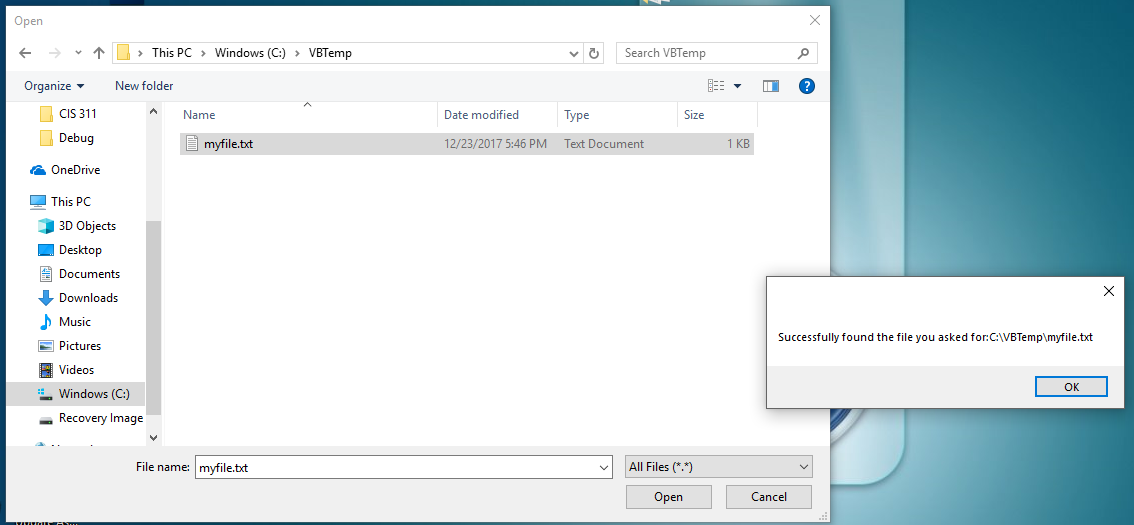


An OpenFileDialog control added to a form

The remainder of this section will discuss the various common dialog controls and how to work with them.

## OpenFileDialog Control

The OpenFileDialog control allows the user to select/enter the name of a file that he or she wants to open. The user can browse around on the filesystem to locate the file, or directly type the filename in from the keyboard. In addition, you can programmatically control what directory the user is placed in and what file extensions you want the user to be able to select. Here’s an example of the OpenFileDialog in action:



Here’s the code behind getting the OpenFileDialog box to do what I want it to do:

Imports System.ComponentModel

'Chapter 12 - Program 6

Public Class Form1

Private Sub Form1\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

'Set up the properties of the OpenFileDialog the way we want

'it to work.

'Only allow files that exist to be opened

OpenFileDialog1.CheckFileExists = True

OpenFileDialog1.CheckPathExists = True

'What types of files will we allow the user to look for?

OpenFileDialog1.Filter =

"Doc Files (\*.doc)|\*.doc|RTF Files (\*.rtf)|\*.rtf|All Files|\*.\*"

'Set up the default file extension and the initial directory

OpenFileDialog1.DefaultExt = ".doc"

OpenFileDialog1.InitialDirectory = "C:\"

'Finally, show the dialog box

If (OpenFileDialog1.ShowDialog <> DialogResult.OK) Then

'if the user cancelled the dialog box instead of clicking

'Open or pressing enter after typing a filename

MessageBox.Show("User cancelled Open File")

End If

End Sub

Private Sub OpenFileDialog1\_FileOk(sender As Object, e As CancelEventArgs)

Handles OpenFileDialog1.FileOk

'This is called whenever the user clicks on the "Open" button

'or presses the enter key after entering a filename

MessageBox.Show("Successfully found the file you asked for:" \_

& OpenFileDialog1.FileName)

End Sub

End Class

Additional OpenFileDialog control properties:

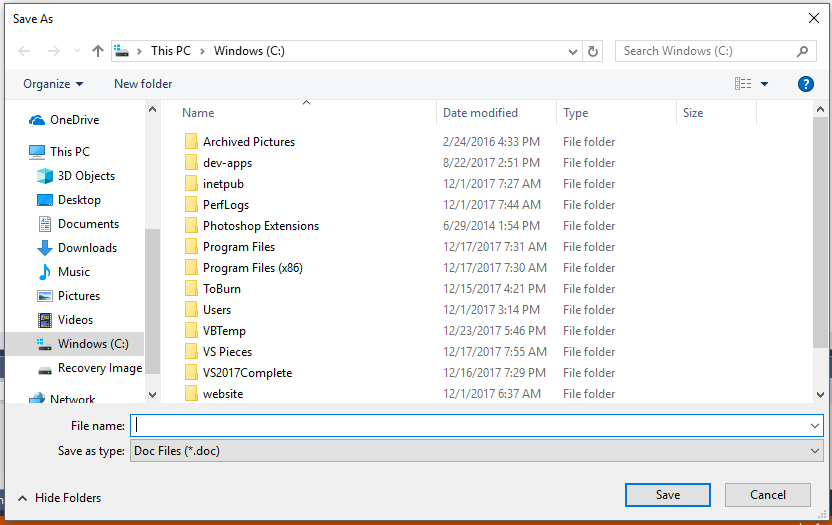
|  |  |
| --- | --- |
| *Property* | *Purpose* |
| AddExtension | Should the dialog box automatically add an extension to the file name if the user omits one? |
| CheckFileExists | Rule that says any file that is picked must be an existing file |
| CheckPathExists | Rule that says the path must be valid |
| DefaultExt | Gets or sets the default file name extension |
| DereferenceLinks | Gets or sets a value that states whether or not a shortcut returns the shortcut .lnk location or the actual file referenced by the shortcut |
| Filename | A string representing the name of the filename selected by the user |
| Filenames | All of the selected files in the dialog box if more than one was selected |
| Filter | Gets or sets the current file name filter string, determining which types of files to look for |
| InitialDirectory | Gets or sets the initial directory displayed |
| Multiselect | Indicates that more than one file can be picked at a time |
| ReadOnlyChecked | Gets or sets the read-only checkbox in the dialog box |
| RestoreDirectory | Gets or sets whether the dialog restores the current directory before closing |
| SafeFileName | Gets the name and extension of a file, without the path |
| SafeFileNames | Like SafeFileName but for multiple selections |
| ShowReadOnly | Indicates if a read-only checkbox should be shown |
| SupportMultiDottedExtensions | Will the dialog support files with multiple file name extensions? |
| ValidateNames | Indicates if the dialog will only accept valid Win32 filenames |

Additional OpenFileDialog control methods:

|  |  |
| --- | --- |
| *Methods* | *Purpose* |
| OpenFile | Opens the file selected by the user, with read-only permission, returning a Stream object the program can use to access the file |
| Reset | Return the dialog to its defaults |

## SaveFileDialog Control

The SaveFileDialog control is the opposite of the OpenFileDialog control: we use it when we want to allow the user to specify the name and location of a file to be saved. Programmatically, it is very similar to the OpenFileDialog control.



Here’s the related code for the above example:

'Chapter 12 - Program 7

Public Class Form1

Private Sub Form1\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

'Prompt if the user is attempting to overwrite a file

SaveFileDialog1.OverwritePrompt = True

'Setup the default file extension and the initial directory

SaveFileDialog1.DefaultExt = ".doc"

SaveFileDialog1.InitialDirectory = "C:\"

'Setup the types of files we would like the user to be able to

'save as

SaveFileDialog1.Filter =

"Doc Files (\*.doc)|\*.doc|RTF Files (\*.rtf)|\*.rtf|All Files|\*.\*"

'Show the dialog box

SaveFileDialog1.ShowDialog()

End Sub

End Class

Many of the methods and properties of the SaveFileDialog control are the same as the OpenFileDialog control. Here is a list of some of the other additional properties:

|  |  |
| --- | --- |
| *Properties* | *Purpose* |
| CreatePrompt | Prompt before creating a new file |
| Filename | The name of the file the user wants to save |
| OverwritePrompt | Warn before overwriting an existing file |

## ColorDialog Control

The ColorDialog control is the standard way in which we allow users to pick different colors. Just about everybody has seen this dialog box, which is shown in the next figure:



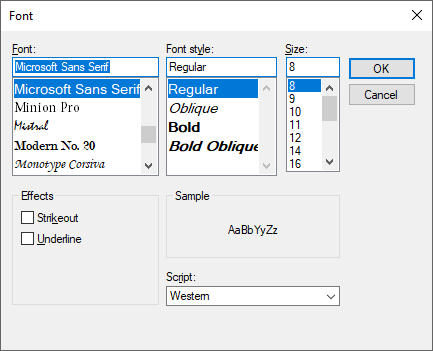
Here’s the simplest code you can write to show this dialog control:

ColorDialog1.ShowDialog()

Once the control has been closed, you can access the color that the user picked through the control's .Color property.

## FontDialog Control

The FontDialog control is used whenever you want your application's user to be able to select a different font:



Here’s the minimum amount of code you would need to show the dialog box:

FontDialog1.ShowDialog()

This snippet of code shows how to display the font dialog box and then set the font of a TextBox to whatever font the user selected through .Font property:

If (myFontBox.ShowDialog() = DialogResult.OK) Then

TextBox1.Font = myFontBox.Font

End If

## Printer Related Common Dialog Controls

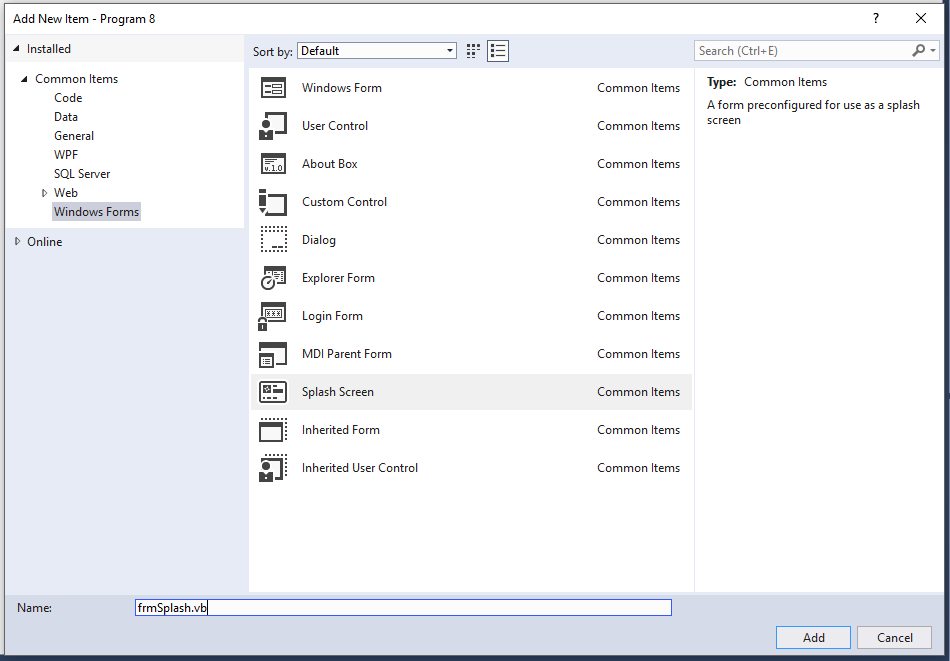
I realize that there are 3 printer related common dialog controls available from the VB.NET toolbox – we will discuss those when we get to the chapter on printing. Just forewarning: don’t get too excited about this! It’s not going to be as great as you think…

## 

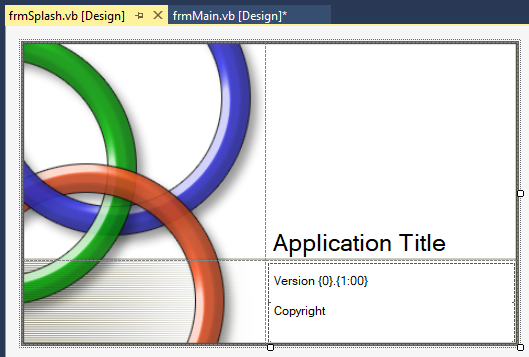
# Creating a Splash Screen

The last item that we will look at in this chapter is how to create a splash screen. Just about every Windows application has a splash screen that is displayed while the application is starting up. We can easily create this effect with Visual Basic too.

Create a project that has your main application form in it named frmMain. Now, use your friend the Project🡪Add Windows Form dialog to add a Splash Screen named frmSplash.vb:

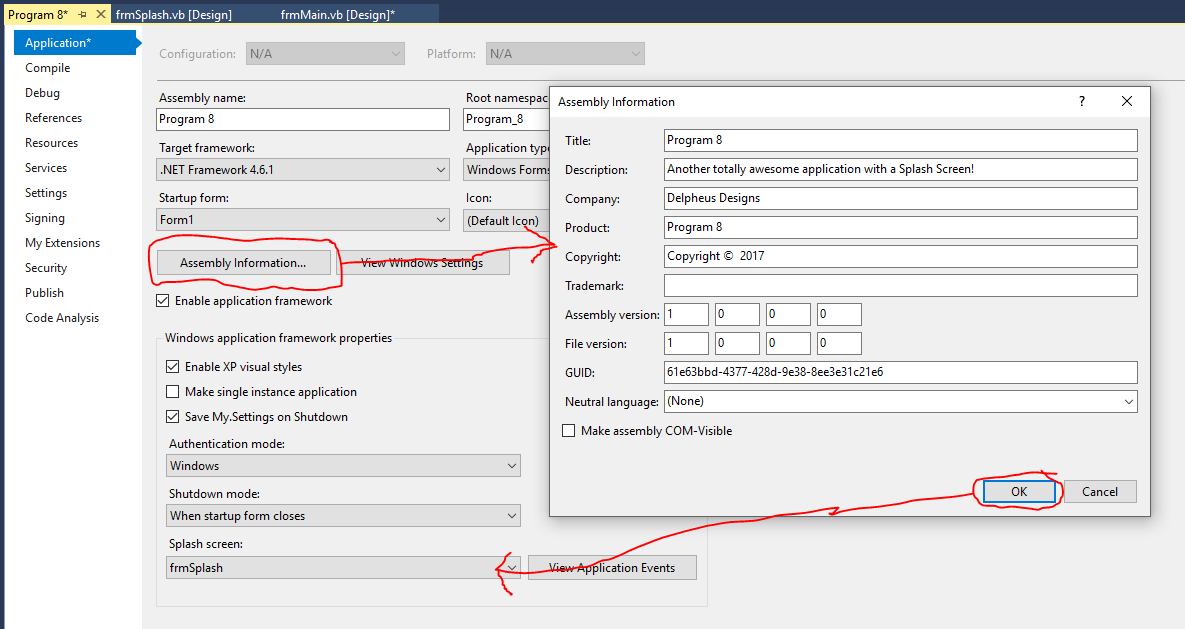


When we’re done, we get a pretty nifty predefined form…



Go ahead and get into your Project Property window, select the Application pane and press the Assembly Information button to update what you want to appear on the Splash Screen.

Also, while you are in the same general area, change the Project Property Splash Screen value from nothing to the frmSplash we just created:



Bask in your programming glory as you run your application, the splash screen appears and then your main application appears. This again is awesome stuff! We used to have to create the splash screen ourselves, and then add a timer on frmMain to display the splash screen for a certain amount of time and then unload the splash screen. Yuck! Do you want to see what that kind of code looked like? Well, you are going to anyway:

Module modMain

'Create instances of both the main and the splash forms

'The pre-.NET way of handling a splash screen...

Public myFrmMain As New frmMain()

Public myFrmSplash As New frmSplash()

Sub Main()

'Make sure both forms are loaded but not shown

myFrmSplash.Hide()

myFrmMain.Hide()

'Set up the properties on the timer located on the main form

myFrmMain.Timer1.Interval = 7000 'milliseconds

myFrmMain.Timer1.Enabled = True

'Show the splash screen (when frmMain's timer goes off,

'we will hide the shut off the timer, hide the splash

'screen and display the main application

myFrmSplash.ShowDialog()

End Sub

End Module

Here’s the code that would need to run behind the frmMain form:

Public Class frmMain

Inherits System.Windows.Forms.Form

Private Sub Timer1\_Tick(ByVal sender As Object, \_

ByVal e As System.EventArgs) \_

Handles Timer1.Tick

'Shut off the timer

Me.Enabled = False

'Hide the splash screen

modMain.myFrmSplash.Hide()

'Enable and show the main application screen

modMain.myFrmMain.Enabled = True

modMain.myFrmMain.ShowDialog()

End Sub

End Class

The idea behind this technique was to load both the main application form and the splash screen up, set up a timer to display the splash screen for 7 seconds and then hide the splash screen and show the main application form. If you are careful when you design your applications, you can load and unload forms as you need them. This will minimize the amount of memory required as well as the time it takes to process them.

# The Life of a Form

We’ve already learned a little bit about some of the methods that we can use when working with forms. We know that Load is called when a form is first brought into memory, and we’ve seen how to chain into Closing when the form is being exited. It’s worthwhile to spend a minute looking at how forms are actually instantiated and destroyed.

When a form is being created for the first time:

1. Control.HandleCreated – Occurs when a handle is created for the control/form
2. Control.BindingContextChanged – Occurs when the binding context property changes (this is how we hook a control up to an underlying data store)
3. Form.Load – Called once the form is loaded into memory but before being displayed to the user
4. Control.VisibleChanged – Occurs when the visibility of the control/form changes
5. Control.GotFocus – The control/form has focus placed upon it (all user input goes to it)
6. Form.Activated – Occurs when the form is activated in code or by the user
7. Form.Shown – Called whenever the form is first displayed

When a form is being shut down:

1. Form.Closing – Occurs when the form is closing
2. Form.FormClosing – Occurs before the form is closed
3. Form.Closed – Occurs when the form is closed
4. Form.FormClosed – Occurs after the form is closed
5. Form.Deactivate – The form is no longer the active form
6. Control.LostFocus – Focus is taken away from the control/form
7. Control.HandleDestroyed – The handle to the control/form is destroyed
8. Component.Disposed – The resources allocated are released back to the system

It’s important to understand when and how these events fire as these only occur one time. Other form events such as VisibleChanged and Enter will occur every time that the form appears. Without understanding the hierarchy, it is possible to overload the event handlers and essentially create code that doesn’t function properly. The same will be true with working with individual control events. Bottom line: make sure that you consult the MSDN documentation if you plan on trying to do something fairly exotic in a bunch of related event handlers.

# Setting Control Properties

As we have seen both in this chapter and in the past chapters, many of the properties that belong to controls are set as we drag and drop the controls onto a form. Setting the properties through either the Property Pages dialog boxes or the Properties window in the VB IDE is called “setting properties at design time.”

Design time properties are a very powerful mechanism that allow us to customize a control and immediately gauge if we are getting the look and feel that we are striving for. The other time that we can set properties is at runtime (called runtime properties), through code. We have already seen some examples of these runtime types of settings, especially with the common dialog controls that we looked at above.

It is extremely important, therefore, that you carefully read the help on any controls that you use in a Visual Basic project to understand which properties are accessible at design time and which are accessible at runtime. In many cases you will find that certain controls have some properties that can only be changed during one or the other.

Here’s an example of changing the caption of a Command button at runtime, after the user has clicked on the button:

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles

Button1.Click

'The .Text property of a command button can be changed both at

'design time and at runtime.

Button1.Text = "I Changed"

End Sub

It’s worthwhile to look at the basic properties, methods and events that controls can respond to. Again, specific controls may have more or less than what’s shown here, so be sure to consult the MSDN documentation for full details.

Control Properties:

|  |  |
| --- | --- |
| *Name* | *Description* |
| [AccessibilityObject](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.accessibilityobject.aspx) | Gets the [AccessibleObject](http://msdn.microsoft.com/en-us/library/system.windows.forms.accessibleobject.aspx) assigned to the control. |
| [AccessibleDefaultActionDescription](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.accessibledefaultactiondescription.aspx) | Gets or sets the default action description of the control for use by accessibility client applications. |
| [AccessibleDescription](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.accessibledescription.aspx) | Gets or sets the description of the control used by accessibility client applications. |
| [AccessibleName](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.accessiblename.aspx) | Gets or sets the name of the control used by accessibility client applications. |
| [AccessibleRole](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.accessiblerole.aspx) | Gets or sets the accessible role of the control |
| [AllowDrop](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.allowdrop.aspx) | Gets or sets a value indicating whether the control can accept data that the user drags onto it. |
| [Anchor](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.anchor.aspx) | Gets or sets the edges of the container to which a control is bound and determines how a control is resized with its parent. |
| [AutoScrollOffset](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.autoscrolloffset.aspx) | Gets or sets where this control is scrolled to in [ScrollControlIntoView](http://msdn.microsoft.com/en-us/library/system.windows.forms.scrollablecontrol.scrollcontrolintoview.aspx). |
| [AutoSize](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.autosize.aspx) | Infrastructure. This property is not relevant for this class. |
| [BackColor](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.backcolor.aspx) | Gets or sets the background color for the control. |
| [BackgroundImage](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.backgroundimage.aspx) | Gets or sets the background image displayed in the control. |
| [BackgroundImageLayout](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.backgroundimagelayout.aspx) | Gets or sets the background image layout as defined in the [ImageLayout](http://msdn.microsoft.com/en-us/library/system.windows.forms.imagelayout.aspx) enumeration. |
| [BindingContext](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.bindingcontext.aspx) | Gets or sets the [BindingContext](http://msdn.microsoft.com/en-us/library/system.windows.forms.bindingcontext.aspx) for the control. |
| [Bottom](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.bottom.aspx) | Gets the distance, in pixels, between the bottom edge of the control and the top edge of its container's client area. |
| [Bounds](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.bounds.aspx) | Gets or sets the size and location of the control including its nonclient elements, in pixels, relative to the parent control. |
| [CanEnableIme](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.canenableime.aspx) | Gets a value indicating whether the [ImeMode](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.imemode.aspx) property can be set to an active value, to enable IME support. |
| [CanFocus](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.canfocus.aspx) | Gets a value indicating whether the control can receive focus. |
| [CanRaiseEvents](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.canraiseevents.aspx) | Determines if events can be raised on the control. (Overrides [Component.CanRaiseEvents](http://msdn.microsoft.com/en-us/library/system.componentmodel.component.canraiseevents.aspx).) |
| [CanSelect](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.canselect.aspx) | Gets a value indicating whether the control can be selected. |
| [Capture](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.capture.aspx) | Gets or sets a value indicating whether the control has captured the mouse. |
| [CausesValidation](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.causesvalidation.aspx) | Gets or sets a value indicating whether the control causes validation to be performed on any controls that require validation when it receives focus. |
| [CheckForIllegalCrossThreadCalls](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.checkforillegalcrossthreadcalls.aspx) | Gets or sets a value indicating whether to catch calls on the wrong thread that access a control's [Handle](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.handle.aspx)property when an application is being debugged. |
| [ClientRectangle](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.clientrectangle.aspx) | Gets the rectangle that represents the client area of the control. |
| [ClientSize](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.clientsize.aspx) | Gets or sets the height and width of the client area of the control. |
| [CompanyName](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.companyname.aspx) | Gets the name of the company or creator of the application containing the control. |
| [Container](http://msdn.microsoft.com/en-us/library/system.componentmodel.component.container.aspx) | Gets the [IContainer](http://msdn.microsoft.com/en-us/library/system.componentmodel.icontainer.aspx) that contains the [Component](http://msdn.microsoft.com/en-us/library/system.componentmodel.component.aspx). (Inherited from [Component](http://msdn.microsoft.com/en-us/library/system.componentmodel.component.aspx).) |
| [ContainsFocus](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.containsfocus.aspx) | Gets a value indicating whether the control, or one of its child controls, currently has the input focus. |
| [ContextMenu](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.contextmenu.aspx) | Gets or sets the shortcut menu associated with the control. |
| [ContextMenuStrip](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.contextmenustrip.aspx) | Gets or sets the [ContextMenuStrip](http://msdn.microsoft.com/en-us/library/system.windows.forms.contextmenustrip.aspx) associated with this control. |
| [Controls](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.controls.aspx) | Gets the collection of controls contained within the control. |
| [Created](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.created.aspx) | Gets a value indicating whether the control has been created. |
| [CreateParams](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.createparams.aspx) | Gets the required creation parameters when the control handle is created. |
| [Cursor](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.cursor.aspx) | Gets or sets the cursor that is displayed when the mouse pointer is over the control. |
| [DataBindings](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.databindings.aspx) | Gets the data bindings for the control. |
| [DefaultBackColor](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.defaultbackcolor.aspx) | Gets the default background color of the control. |
| [DefaultCursor](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.defaultcursor.aspx) | Gets or sets the default cursor for the control. |
| [DefaultFont](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.defaultfont.aspx) | Gets the default font of the control. |
| [DefaultForeColor](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.defaultforecolor.aspx) | Gets the default foreground color of the control. |
| [DefaultImeMode](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.defaultimemode.aspx) | Gets the default Input Method Editor (IME) mode supported by the control. |
| [DefaultMargin](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.defaultmargin.aspx) | Gets the space, in pixels, that is specified by default between controls. |
| [DefaultMaximumSize](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.defaultmaximumsize.aspx) | Gets the length and height, in pixels, that is specified as the default maximum size of a control. |
| [DefaultMinimumSize](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.defaultminimumsize.aspx) | Gets the length and height, in pixels, that is specified as the default minimum size of a control. |
| [DefaultPadding](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.defaultpadding.aspx) | Gets the internal spacing, in pixels, of the contents of a control. |
| [DefaultSize](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.defaultsize.aspx) | Gets the default size of the control. |
| [DesignMode](http://msdn.microsoft.com/en-us/library/system.componentmodel.component.designmode.aspx) | Gets a value that indicates whether the [Component](http://msdn.microsoft.com/en-us/library/system.componentmodel.component.aspx) is currently in design mode. (Inherited from [Component](http://msdn.microsoft.com/en-us/library/system.componentmodel.component.aspx).) |
| [DisplayRectangle](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.displayrectangle.aspx) | Gets the rectangle that represents the display area of the control. |
| [Disposing](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.disposing.aspx) | Gets a value indicating whether the base Control class is in the process of disposing. |
| [Dock](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.dock.aspx) | Gets or sets which control borders are docked to its parent control and determines how a control is resized with its parent. |
| [DoubleBuffered](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.doublebuffered.aspx) | Gets or sets a value indicating whether this control should redraw its surface using a secondary buffer to reduce or prevent flicker. |
| [Enabled](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.enabled.aspx) | Gets or sets a value indicating whether the control can respond to user interaction. |
| [Events](http://msdn.microsoft.com/en-us/library/system.componentmodel.component.events.aspx) | Gets the list of event handlers that are attached to this [Component](http://msdn.microsoft.com/en-us/library/system.componentmodel.component.aspx). (Inherited from [Component](http://msdn.microsoft.com/en-us/library/system.componentmodel.component.aspx).) |
| [Focused](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.focused.aspx) | Gets a value indicating whether the control has input focus. |
| [Font](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.font.aspx) | Gets or sets the font of the text displayed by the control. |
| [FontHeight](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.fontheight.aspx) | Gets or sets the height of the font of the control. |
| [ForeColor](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.forecolor.aspx) | Gets or sets the foreground color of the control. |
| [Handle](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.handle.aspx) | Gets the window handle that the control is bound to. |
| [HasChildren](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.haschildren.aspx) | Gets a value indicating whether the control contains one or more child controls. |
| [Height](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.height.aspx) | Gets or sets the height of the control. |
| [ImeMode](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.imemode.aspx) | Gets or sets the Input Method Editor (IME) mode of the control. |
| [ImeModeBase](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.imemodebase.aspx) | Gets or sets the IME mode of a control. |
| [InvokeRequired](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.invokerequired.aspx) | Gets a value indicating whether the caller must call an invoke method when making method calls to the control because the caller is on a different thread than the one the control was created on. |
| [IsAccessible](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.isaccessible.aspx) | Gets or sets a value indicating whether the control is visible to accessibility applications. |
| [IsDisposed](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.isdisposed.aspx) | Gets a value indicating whether the control has been disposed of. |
| [IsHandleCreated](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.ishandlecreated.aspx) | Gets a value indicating whether the control has a handle associated with it. |
| [IsMirrored](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.ismirrored.aspx) | Gets a value indicating whether the control is mirrored. |
| [LayoutEngine](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.layoutengine.aspx) | Gets a cached instance of the control's layout engine. |
| [Left](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.left.aspx) | Gets or sets the distance, in pixels, between the left edge of the control and the left edge of its container's client area. |
| [Location](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.location.aspx) | Gets or sets the coordinates of the upper-left corner of the control relative to the upper-left corner of its container. |
| [Margin](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.margin.aspx) | Gets or sets the space between controls. |
| [MaximumSize](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.maximumsize.aspx) | Gets or sets the size that is the upper limit that [GetPreferredSize](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.getpreferredsize.aspx) can specify. |
| [MinimumSize](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.minimumsize.aspx) | Gets or sets the size that is the lower limit that [GetPreferredSize](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.getpreferredsize.aspx) can specify. |
| [ModifierKeys](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.modifierkeys.aspx) | Gets a value indicating which of the modifier keys (SHIFT, CTRL, and ALT) is in a pressed state. |
| [MouseButtons](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.mousebuttons.aspx) | Gets a value indicating which of the mouse buttons is in a pressed state. |
| [MousePosition](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.mouseposition.aspx) | Gets the position of the mouse cursor in screen coordinates. |
| [Name](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.name.aspx) | Gets or sets the name of the control. |
| [Padding](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.padding.aspx) | Gets or sets padding within the control. |
| [Parent](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.parent.aspx) | Gets or sets the parent container of the control. |
| [PreferredSize](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.preferredsize.aspx) | Gets the size of a rectangular area into which the control can fit. |
| [ProductName](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.productname.aspx) | Gets the product name of the assembly containing the control. |
| [ProductVersion](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.productversion.aspx) | Gets the version of the assembly containing the control. |
| [PropagatingImeMode](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.propagatingimemode.aspx) | Gets an object that represents a propagating IME mode. |
| [RecreatingHandle](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.recreatinghandle.aspx) | Gets a value indicating whether the control is currently re-creating its handle. |
| [Region](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.region.aspx) | Gets or sets the window region associated with the control. |
| [ResizeRedraw](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.resizeredraw.aspx) | Gets or sets a value indicating whether the control redraws itself when resized. |
| [Right](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.right.aspx) | Gets the distance, in pixels, between the right edge of the control and the left edge of its container's client area. |
| [RightToLeft](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.righttoleft.aspx) | Gets or sets a value indicating whether control's elements are aligned to support locales using right-to-left fonts. |
| [ScaleChildren](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.scalechildren.aspx) | Gets a value that determines the scaling of child controls. |
| [ShowFocusCues](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.showfocuscues.aspx) | Gets a value indicating whether the control should display focus rectangles. |
| [ShowKeyboardCues](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.showkeyboardcues.aspx) | Gets a value indicating whether the user interface is in the appropriate state to show or hide keyboard accelerators. |
| [Site](http://msdn.microsoft.com/en-us/library/24sf20kb.aspx) | Gets or sets the site of the control. (Overrides [Component.Site](http://msdn.microsoft.com/en-us/library/system.componentmodel.component.site.aspx).) |
| [Size](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.size.aspx) | Gets or sets the height and width of the control. |
| [TabIndex](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.tabindex.aspx) | Gets or sets the tab order of the control within its container. |
| [TabStop](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.tabstop.aspx) | Gets or sets a value indicating whether the user can give the focus to this control using the TAB key. |
| [Tag](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.tag.aspx) | Gets or sets the object that contains data about the control. |
| [Text](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.text.aspx) | Gets or sets the text associated with this control. |
| [Top](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.top.aspx) | Gets or sets the distance, in pixels, between the top edge of the control and the top edge of its container's client area. |
| [TopLevelControl](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.toplevelcontrol.aspx) | Gets the parent control that is not parented by another Windows Forms control. Typically, this is the outermost[Form](http://msdn.microsoft.com/en-us/library/system.windows.forms.form.aspx) that the control is contained in. |
| [UseWaitCursor](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.usewaitcursor.aspx) | Gets or sets a value indicating whether to use the wait cursor for the current control and all child controls. |
| [Visible](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.visible.aspx) | Gets or sets a value indicating whether the control and all its child controls are displayed. |
| [Width](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.width.aspx) | Gets or sets the width of the control. |
| [WindowTarget](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.windowtarget.aspx) | Infrastructure. This property is not relevant for this class. |

Control Methods:

|  |  |
| --- | --- |
| *Name* | *Description* |
| [AccessibilityNotifyClients(AccessibleEvents, Int32)](http://msdn.microsoft.com/en-us/library/yett96bx.aspx) | Notifies the accessibility client applications of the specified [AccessibleEvents](http://msdn.microsoft.com/en-us/library/system.windows.forms.accessibleevents.aspx) for the specified child control. |
| [AccessibilityNotifyClients(AccessibleEvents, Int32, Int32)](http://msdn.microsoft.com/en-us/library/7skkyz2e.aspx) | Notifies the accessibility client applications of the specified [AccessibleEvents](http://msdn.microsoft.com/en-us/library/system.windows.forms.accessibleevents.aspx) for the specified child control . |
| [BeginInvoke(Delegate)](http://msdn.microsoft.com/en-us/library/0b1bf3y3.aspx) | Executes the specified delegate asynchronously on the thread that the control's underlying handle was created on. |
| [BeginInvoke(Delegate, Object())](http://msdn.microsoft.com/en-us/library/a06c0dc2.aspx) | Executes the specified delegate asynchronously with the specified arguments, on the thread that the control's underlying handle was created on. |
| [BringToFront](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.bringtofront.aspx) | Brings the control to the front of the z-order. |
| [Contains](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.contains.aspx) | Retrieves a value indicating whether the specified control is a child of the control. |
| [CreateAccessibilityInstance](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.createaccessibilityinstance.aspx) | Creates a new accessibility object for the control. |
| [CreateControl](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.createcontrol.aspx) | Forces the creation of the visible control, including the creation of the handle and any visible child controls. |
| [CreateControlsInstance](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.createcontrolsinstance.aspx) | Creates a new instance of the control collection for the control. |
| [CreateGraphics](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.creategraphics.aspx) | Creates the [Graphics](http://msdn.microsoft.com/en-us/library/system.drawing.graphics.aspx) for the control. |
| [CreateHandle](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.createhandle.aspx) | Creates a handle for the control. |
| [CreateObjRef](http://msdn.microsoft.com/en-us/library/system.marshalbyrefobject.createobjref.aspx) | Creates an object that contains all the relevant information required to generate a proxy used to communicate with a remote object. (Inherited from [MarshalByRefObject](http://msdn.microsoft.com/en-us/library/system.marshalbyrefobject.aspx).) |
| [DefWndProc](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.defwndproc.aspx) | Sends the specified message to the default window procedure. |
| [DestroyHandle](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.destroyhandle.aspx) | Destroys the handle associated with the control. |
| [Dispose](http://msdn.microsoft.com/en-us/library/3cc9y48w.aspx) | Releases all resources used by the [Component](http://msdn.microsoft.com/en-us/library/system.componentmodel.component.aspx). (Inherited from [Component](http://msdn.microsoft.com/en-us/library/system.componentmodel.component.aspx).) |
| [Dispose(Boolean)](http://msdn.microsoft.com/en-us/library/a4zkb31d.aspx) | Releases the unmanaged resources used by the Control and its child controls and optionally releases the managed resources. (Overrides [Component.Dispose(Boolean)](http://msdn.microsoft.com/en-us/library/d9yzd5cx.aspx).) |
| [DoDragDrop](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.dodragdrop.aspx) | Begins a drag-and-drop operation. |
| [DrawToBitmap](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.drawtobitmap.aspx) | Supports rendering to the specified bitmap. |
| [EndInvoke](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.endinvoke.aspx) | Retrieves the return value of the asynchronous operation represented by the [IAsyncResult](http://msdn.microsoft.com/en-us/library/system.iasyncresult.aspx) passed. |
| [Equals(Object)](http://msdn.microsoft.com/en-us/library/bsc2ak47.aspx) | Determines whether the specified object is equal to the current object. (Inherited from [Object](http://msdn.microsoft.com/en-us/library/system.object.aspx).) |
| [Finalize](http://msdn.microsoft.com/en-us/library/az5741fh.aspx) | Releases unmanaged resources and performs other cleanup operations before the [Component](http://msdn.microsoft.com/en-us/library/system.componentmodel.component.aspx) is reclaimed by garbage collection. (Inherited from [Component](http://msdn.microsoft.com/en-us/library/system.componentmodel.component.aspx).) |
| [FindForm](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.findform.aspx) | Retrieves the form that the control is on. |
| [Focus](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.focus.aspx) | Sets input focus to the control. |
| [FromChildHandle](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.fromchildhandle.aspx) | Retrieves the control that contains the specified handle. |
| [FromHandle](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.fromhandle.aspx) | Returns the control that is currently associated with the specified handle. |
| [GetAccessibilityObjectById](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.getaccessibilityobjectbyid.aspx) | Retrieves the specified [AccessibleObject](http://msdn.microsoft.com/en-us/library/system.windows.forms.accessibleobject.aspx). |
| [GetAutoSizeMode](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.getautosizemode.aspx) | Retrieves a value indicating how a control will behave when its [AutoSize](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.autosize.aspx) property is enabled. |
| [GetChildAtPoint(Point)](http://msdn.microsoft.com/en-us/library/a6zktd23.aspx) | Retrieves the child control that is located at the specified coordinates. |
| [GetChildAtPoint(Point, GetChildAtPointSkip)](http://msdn.microsoft.com/en-us/library/ms158404.aspx) | Retrieves the child control that is located at the specified coordinates, specifying whether to ignore child controls of a certain type. |
| [GetContainerControl](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.getcontainercontrol.aspx) | Returns the next [ContainerControl](http://msdn.microsoft.com/en-us/library/system.windows.forms.containercontrol.aspx) up the control's chain of parent controls. |
| [GetHashCode](http://msdn.microsoft.com/en-us/library/system.object.gethashcode.aspx) | Serves as a hash function for a particular type. (Inherited from [Object](http://msdn.microsoft.com/en-us/library/system.object.aspx).) |
| [GetLifetimeService](http://msdn.microsoft.com/en-us/library/system.marshalbyrefobject.getlifetimeservice.aspx) | Retrieves the current lifetime service object that controls the lifetime policy for this instance. (Inherited from [MarshalByRefObject](http://msdn.microsoft.com/en-us/library/system.marshalbyrefobject.aspx).) |
| [GetNextControl](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.getnextcontrol.aspx) | Retrieves the next control forward or back in the tab order of child controls. |
| [GetPreferredSize](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.getpreferredsize.aspx) | Retrieves the size of a rectangular area into which a control can be fitted. |
| [GetScaledBounds](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.getscaledbounds.aspx) | Retrieves the bounds within which the control is scaled. |
| [GetService](http://msdn.microsoft.com/en-us/library/system.componentmodel.component.getservice.aspx) | Returns an object that represents a service provided by the [Component](http://msdn.microsoft.com/en-us/library/system.componentmodel.component.aspx) or by its [Container](http://msdn.microsoft.com/en-us/library/system.componentmodel.container.aspx). (Inherited from [Component](http://msdn.microsoft.com/en-us/library/system.componentmodel.component.aspx).) |
| [GetStyle](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.getstyle.aspx) | Retrieves the value of the specified control style bit for the control. |
| [GetTopLevel](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.gettoplevel.aspx) | Determines if the control is a top-level control. |
| [GetType](http://msdn.microsoft.com/en-us/library/system.object.gettype.aspx) | Gets the [Type](http://msdn.microsoft.com/en-us/library/system.type.aspx) of the current instance. (Inherited from [Object](http://msdn.microsoft.com/en-us/library/system.object.aspx).) |
| [Hide](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.hide.aspx) | Conceals the control from the user. |
| [InitializeLifetimeService](http://msdn.microsoft.com/en-us/library/system.marshalbyrefobject.initializelifetimeservice.aspx) | Obtains a lifetime service object to control the lifetime policy for this instance. (Inherited from[MarshalByRefObject](http://msdn.microsoft.com/en-us/library/system.marshalbyrefobject.aspx).) |
| [InitLayout](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.initlayout.aspx) | Called after the control has been added to another container. |
| [Invalidate](http://msdn.microsoft.com/en-us/library/598t492a.aspx) | Invalidates the entire surface of the control and causes the control to be redrawn. |
| [Invalidate(Boolean)](http://msdn.microsoft.com/en-us/library/xz8ytzt0.aspx) | Invalidates a specific region of the control and causes a paint message to be sent to the control. Optionally, invalidates the child controls assigned to the control. |
| [Invalidate(Rectangle)](http://msdn.microsoft.com/en-us/library/8dtk06x2.aspx) | Invalidates the specified region of the control (adds it to the control's update region, which is the area that will be repainted at the next paint operation), and causes a paint message to be sent to the control. |
| [Invalidate(Region)](http://msdn.microsoft.com/en-us/library/wtzka3b5.aspx) | Invalidates the specified region of the control (adds it to the control's update region, which is the area that will be repainted at the next paint operation), and causes a paint message to be sent to the control. |
| [Invalidate(Rectangle, Boolean)](http://msdn.microsoft.com/en-us/library/8bk29t0f.aspx) | Invalidates the specified region of the control (adds it to the control's update region, which is the area that will be repainted at the next paint operation), and causes a paint message to be sent to the control. Optionally, invalidates the child controls assigned to the control. |
| [Invalidate(Region, Boolean)](http://msdn.microsoft.com/en-us/library/w6w9csht.aspx) | Invalidates the specified region of the control (adds it to the control's update region, which is the area that will be repainted at the next paint operation), and causes a paint message to be sent to the control. Optionally, invalidates the child controls assigned to the control. |
| [Invoke(Delegate)](http://msdn.microsoft.com/en-us/library/zyzhdc6b.aspx) | Executes the specified delegate on the thread that owns the control's underlying window handle. |
| [Invoke(Delegate, Object())](http://msdn.microsoft.com/en-us/library/a1hetckb.aspx) | Executes the specified delegate, on the thread that owns the control's underlying window handle, with the specified list of arguments. |
| [InvokeGotFocus](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.invokegotfocus.aspx) | Raises the [GotFocus](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.gotfocus.aspx) event for the specified control. |
| [InvokeLostFocus](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.invokelostfocus.aspx) | Raises the [LostFocus](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.lostfocus.aspx) event for the specified control. |
| [InvokeOnClick](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.invokeonclick.aspx) | Raises the [Click](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.click.aspx) event for the specified control. |
| [InvokePaint](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.invokepaint.aspx) | Raises the [Paint](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.paint.aspx) event for the specified control. |
| [InvokePaintBackground](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.invokepaintbackground.aspx) | Raises the PaintBackground event for the specified control. |
| [IsInputChar](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.isinputchar.aspx) | Determines if a character is an input character that the control recognizes. |
| [IsInputKey](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.isinputkey.aspx) | Determines whether the specified key is a regular input key or a special key that requires preprocessing. |
| [IsKeyLocked](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.iskeylocked.aspx) | Determines whether the CAPS LOCK, NUM LOCK, or SCROLL LOCK key is in effect. |
| [IsMnemonic](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.ismnemonic.aspx) | Determines if the specified character is the mnemonic character assigned to the control in the specified string. |
| [MemberwiseClone](http://msdn.microsoft.com/en-us/library/system.object.memberwiseclone.aspx) | Creates a shallow copy of the current [Object](http://msdn.microsoft.com/en-us/library/system.object.aspx). (Inherited from [Object](http://msdn.microsoft.com/en-us/library/system.object.aspx).) |
| [MemberwiseClone(Boolean)](http://msdn.microsoft.com/en-us/library/ms131262.aspx) | Creates a shallow copy of the current [MarshalByRefObject](http://msdn.microsoft.com/en-us/library/system.marshalbyrefobject.aspx) object. (Inherited from[MarshalByRefObject](http://msdn.microsoft.com/en-us/library/system.marshalbyrefobject.aspx).) |
| [NotifyInvalidate](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.notifyinvalidate.aspx) | Infrastructure. Raises the [Invalidated](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.invalidated.aspx) event with a specified region of the control to invalidate. |
| [OnAutoSizeChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onautosizechanged.aspx) | Raises the [AutoSizeChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.autosizechanged.aspx) event. |
| [OnBackColorChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onbackcolorchanged.aspx) | Raises the [BackColorChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.backcolorchanged.aspx) event. |
| [OnBackgroundImageChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onbackgroundimagechanged.aspx) | Raises the [BackgroundImageChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.backgroundimagechanged.aspx) event. |
| [OnBackgroundImageLayoutChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onbackgroundimagelayoutchanged.aspx) | Raises the [BackgroundImageLayoutChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.backgroundimagelayoutchanged.aspx) event. |
| [OnBindingContextChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onbindingcontextchanged.aspx) | Raises the [BindingContextChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.bindingcontextchanged.aspx) event. |
| [OnCausesValidationChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.oncausesvalidationchanged.aspx) | Raises the [CausesValidationChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.causesvalidationchanged.aspx) event. |
| [OnChangeUICues](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onchangeuicues.aspx) | Raises the [ChangeUICues](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.changeuicues.aspx) event. |
| [OnClick](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onclick.aspx) | Raises the [Click](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.click.aspx) event. |
| [OnClientSizeChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onclientsizechanged.aspx) | Raises the [ClientSizeChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.clientsizechanged.aspx) event. |
| [OnContextMenuChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.oncontextmenuchanged.aspx) | Raises the [ContextMenuChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.contextmenuchanged.aspx) event. |
| [OnContextMenuStripChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.oncontextmenustripchanged.aspx) | Raises the [ContextMenuStripChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.contextmenustripchanged.aspx) event. |
| [OnControlAdded](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.oncontroladded.aspx) | Raises the [ControlAdded](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.controladded.aspx) event. |
| [OnControlRemoved](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.oncontrolremoved.aspx) | Raises the [ControlRemoved](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.controlremoved.aspx) event. |
| [OnCreateControl](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.oncreatecontrol.aspx) | Raises the [CreateControl](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.createcontrol.aspx) method. |
| [OnCursorChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.oncursorchanged.aspx) | Raises the [CursorChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.cursorchanged.aspx) event. |
| [OnDockChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.ondockchanged.aspx) | Raises the [DockChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.dockchanged.aspx) event. |
| [OnDoubleClick](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.ondoubleclick.aspx) | Raises the [DoubleClick](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.doubleclick.aspx) event. |
| [OnDragDrop](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.ondragdrop.aspx) | Raises the [DragDrop](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.dragdrop.aspx) event. |
| [OnDragEnter](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.ondragenter.aspx) | Raises the [DragEnter](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.dragenter.aspx) event. |
| [OnDragLeave](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.ondragleave.aspx) | Raises the [DragLeave](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.dragleave.aspx) event. |
| [OnDragOver](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.ondragover.aspx) | Raises the [DragOver](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.dragover.aspx) event. |
| [OnEnabledChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onenabledchanged.aspx) | Raises the [EnabledChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.enabledchanged.aspx) event. |
| [OnEnter](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onenter.aspx) | Raises the [Enter](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.enter.aspx) event. |
| [OnFontChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onfontchanged.aspx) | Raises the [FontChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.fontchanged.aspx) event. |
| [OnForeColorChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onforecolorchanged.aspx) | Raises the [ForeColorChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.forecolorchanged.aspx) event. |
| [OnGiveFeedback](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.ongivefeedback.aspx) | Raises the [GiveFeedback](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.givefeedback.aspx) event. |
| [OnGotFocus](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.ongotfocus.aspx) | Raises the [GotFocus](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.gotfocus.aspx) event. |
| [OnHandleCreated](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onhandlecreated.aspx) | Raises the [HandleCreated](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.handlecreated.aspx) event. |
| [OnHandleDestroyed](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onhandledestroyed.aspx) | Raises the [HandleDestroyed](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.handledestroyed.aspx) event. |
| [OnHelpRequested](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onhelprequested.aspx) | Raises the [HelpRequested](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.helprequested.aspx) event. |
| [OnImeModeChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onimemodechanged.aspx) | Raises the [ImeModeChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.imemodechanged.aspx) event. |
| [OnInvalidated](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.oninvalidated.aspx) | Raises the [Invalidated](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.invalidated.aspx) event. |
| [OnKeyDown](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onkeydown.aspx) | Raises the [KeyDown](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.keydown.aspx) event. |
| [OnKeyPress](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onkeypress.aspx) | Raises the [KeyPress](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.keypress.aspx) event. |
| [OnKeyUp](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onkeyup.aspx) | Raises the [KeyUp](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.keyup.aspx) event. |
| [OnLayout](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onlayout.aspx) | Raises the [Layout](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.layout.aspx) event. |
| [OnLeave](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onleave.aspx) | Raises the [Leave](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.leave.aspx) event. |
| [OnLocationChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onlocationchanged.aspx) | Raises the [LocationChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.locationchanged.aspx) event. |
| [OnLostFocus](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onlostfocus.aspx) | Raises the [LostFocus](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.lostfocus.aspx) event. |
| [OnMarginChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onmarginchanged.aspx) | Raises the [MarginChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.marginchanged.aspx) event. |
| [OnMouseCaptureChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onmousecapturechanged.aspx) | Raises the [MouseCaptureChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.mousecapturechanged.aspx) event. |
| [OnMouseClick](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onmouseclick.aspx) | Raises the [MouseClick](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.mouseclick.aspx) event. |
| [OnMouseDoubleClick](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onmousedoubleclick.aspx) | Raises the [MouseDoubleClick](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.mousedoubleclick.aspx) event. |
| [OnMouseDown](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onmousedown.aspx) | Raises the [MouseDown](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.mousedown.aspx) event. |
| [OnMouseEnter](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onmouseenter.aspx) | Raises the [MouseEnter](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.mouseenter.aspx) event. |
| [OnMouseHover](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onmousehover.aspx) | Raises the [MouseHover](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.mousehover.aspx) event. |
| [OnMouseLeave](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onmouseleave.aspx) | Raises the [MouseLeave](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.mouseleave.aspx) event. |
| [OnMouseMove](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onmousemove.aspx) | Raises the [MouseMove](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.mousemove.aspx) event. |
| [OnMouseUp](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onmouseup.aspx) | Raises the [MouseUp](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.mouseup.aspx) event. |
| [OnMouseWheel](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onmousewheel.aspx) | Raises the [MouseWheel](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.mousewheel.aspx) event. |
| [OnMove](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onmove.aspx) | Raises the [Move](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.move.aspx) event. |
| [OnNotifyMessage](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onnotifymessage.aspx) | Notifies the control of Windows messages. |
| [OnPaddingChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onpaddingchanged.aspx) | Raises the [PaddingChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.paddingchanged.aspx) event. |
| [OnPaint](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onpaint.aspx) | Raises the [Paint](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.paint.aspx) event. |
| [OnPaintBackground](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onpaintbackground.aspx) | Paints the background of the control. |
| [OnParentBackColorChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onparentbackcolorchanged.aspx) | Raises the [BackColorChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.backcolorchanged.aspx) event when the [BackColor](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.backcolor.aspx) property value of the control's container changes. |
| [OnParentBackgroundImageChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onparentbackgroundimagechanged.aspx) | Raises the [BackgroundImageChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.backgroundimagechanged.aspx) event when the [BackgroundImage](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.backgroundimage.aspx) property value of the control's container changes. |
| [OnParentBindingContextChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onparentbindingcontextchanged.aspx) | Raises the [BindingContextChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.bindingcontextchanged.aspx) event when the [BindingContext](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.bindingcontext.aspx) property value of the control's container changes. |
| [OnParentChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onparentchanged.aspx) | Raises the [ParentChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.parentchanged.aspx) event. |
| [OnParentCursorChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onparentcursorchanged.aspx) | Raises the [CursorChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.cursorchanged.aspx) event. |
| [OnParentEnabledChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onparentenabledchanged.aspx) | Raises the [EnabledChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.enabledchanged.aspx) event when the [Enabled](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.enabled.aspx) property value of the control's container changes. |
| [OnParentFontChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onparentfontchanged.aspx) | Raises the [FontChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.fontchanged.aspx) event when the [Font](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.font.aspx) property value of the control's container changes. |
| [OnParentForeColorChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onparentforecolorchanged.aspx) | Raises the [ForeColorChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.forecolorchanged.aspx) event when the [ForeColor](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.forecolor.aspx) property value of the control's container changes. |
| [OnParentRightToLeftChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onparentrighttoleftchanged.aspx) | Raises the [RightToLeftChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.righttoleftchanged.aspx) event when the [RightToLeft](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.righttoleft.aspx) property value of the control's container changes. |
| [OnParentVisibleChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onparentvisiblechanged.aspx) | Raises the [VisibleChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.visiblechanged.aspx) event when the [Visible](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.visible.aspx) property value of the control's container changes. |
| [OnPreviewKeyDown](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onpreviewkeydown.aspx) | Raises the [PreviewKeyDown](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.previewkeydown.aspx) event. |
| [OnPrint](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onprint.aspx) | Raises the [Paint](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.paint.aspx) event. |
| [OnQueryContinueDrag](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onquerycontinuedrag.aspx) | Raises the [QueryContinueDrag](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.querycontinuedrag.aspx) event. |
| [OnRegionChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onregionchanged.aspx) | Raises the [RegionChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.regionchanged.aspx) event. |
| [OnResize](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onresize.aspx) | Raises the [Resize](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.resize.aspx) event. |
| [OnRightToLeftChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onrighttoleftchanged.aspx) | Raises the [RightToLeftChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.righttoleftchanged.aspx) event. |
| [OnSizeChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onsizechanged.aspx) | Raises the [SizeChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.sizechanged.aspx) event. |
| [OnStyleChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onstylechanged.aspx) | Raises the [StyleChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.stylechanged.aspx) event. |
| [OnSystemColorsChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onsystemcolorschanged.aspx) | Raises the [SystemColorsChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.systemcolorschanged.aspx) event. |
| [OnTabIndexChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.ontabindexchanged.aspx) | Raises the [TabIndexChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.tabindexchanged.aspx) event. |
| [OnTabStopChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.ontabstopchanged.aspx) | Raises the [TabStopChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.tabstopchanged.aspx) event. |
| [OnTextChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.ontextchanged.aspx) | Raises the [TextChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.textchanged.aspx) event. |
| [OnValidated](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onvalidated.aspx) | Raises the [Validated](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.validated.aspx) event. |
| [OnValidating](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onvalidating.aspx) | Raises the [Validating](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.validating.aspx) event. |
| [OnVisibleChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.onvisiblechanged.aspx) | Raises the [VisibleChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.visiblechanged.aspx) event. |
| [PerformLayout](http://msdn.microsoft.com/en-us/library/z8aw0sd8.aspx) | Forces the control to apply layout logic to all its child controls. |
| [PerformLayout(Control, String)](http://msdn.microsoft.com/en-us/library/xt7y95c1.aspx) | Forces the control to apply layout logic to all its child controls. |
| [PointToClient](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.pointtoclient.aspx) | Computes the location of the specified screen point into client coordinates. |
| [PointToScreen](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.pointtoscreen.aspx) | Computes the location of the specified client point into screen coordinates. |
| [PreProcessControlMessage](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.preprocesscontrolmessage.aspx) | Preprocesses keyboard or input messages within the message loop before they are dispatched. |
| [PreProcessMessage](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.preprocessmessage.aspx) | Preprocesses keyboard or input messages within the message loop before they are dispatched. |
| [ProcessCmdKey](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.processcmdkey.aspx) | Processes a command key. |
| [ProcessDialogChar](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.processdialogchar.aspx) | Processes a dialog character. |
| [ProcessDialogKey](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.processdialogkey.aspx) | Processes a dialog key. |
| [ProcessKeyEventArgs](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.processkeyeventargs.aspx) | Processes a key message and generates the appropriate control events. |
| [ProcessKeyMessage](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.processkeymessage.aspx) | Processes a keyboard message. |
| [ProcessKeyPreview](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.processkeypreview.aspx) | Previews a keyboard message. |
| [ProcessMnemonic](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.processmnemonic.aspx) | Processes a mnemonic character. |
| [RaiseDragEvent](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.raisedragevent.aspx) | Infrastructure. Raises the appropriate drag event. |
| [RaiseKeyEvent](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.raisekeyevent.aspx) | Infrastructure. Raises the appropriate key event. |
| [RaiseMouseEvent](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.raisemouseevent.aspx) | Infrastructure. Raises the appropriate mouse event. |
| [RaisePaintEvent](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.raisepaintevent.aspx) | Infrastructure. Raises the appropriate paint event. |
| [RecreateHandle](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.recreatehandle.aspx) | Forces the re-creation of the handle for the control. |
| [RectangleToClient](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.rectangletoclient.aspx) | Computes the size and location of the specified screen rectangle in client coordinates. |
| [RectangleToScreen](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.rectangletoscreen.aspx) | Computes the size and location of the specified client rectangle in screen coordinates. |
| [ReflectMessage](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.reflectmessage.aspx) | Reflects the specified message to the control that is bound to the specified handle. |
| [Refresh](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.refresh.aspx) | Forces the control to invalidate its client area and immediately redraw itself and any child controls. |
| [ResetBackColor](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.resetbackcolor.aspx) | Resets the [BackColor](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.backcolor.aspx) property to its default value. |
| [ResetBindings](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.resetbindings.aspx) | Causes a control bound to the [BindingSource](http://msdn.microsoft.com/en-us/library/system.windows.forms.bindingsource.aspx) to reread all the items in the list and refresh their displayed values. |
| [ResetCursor](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.resetcursor.aspx) | Resets the [Cursor](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.cursor.aspx) property to its default value. |
| [ResetFont](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.resetfont.aspx) | Resets the [Font](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.font.aspx) property to its default value. |
| [ResetForeColor](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.resetforecolor.aspx) | Resets the [ForeColor](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.forecolor.aspx) property to its default value. |
| [ResetImeMode](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.resetimemode.aspx) | Resets the [ImeMode](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.imemode.aspx) property to its default value. |
| [ResetMouseEventArgs](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.resetmouseeventargs.aspx) | Infrastructure. Resets the control to handle the [MouseLeave](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.mouseleave.aspx) event. |
| [ResetRightToLeft](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.resetrighttoleft.aspx) | Resets the [RightToLeft](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.righttoleft.aspx) property to its default value. |
| [ResetText](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.resettext.aspx) | Resets the [Text](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.text.aspx) property to its default value. |
| [ResumeLayout](http://msdn.microsoft.com/en-us/library/y53zat12.aspx) | Resumes usual layout logic. |
| [ResumeLayout(Boolean)](http://msdn.microsoft.com/en-us/library/w8k76wfs.aspx) | Resumes usual layout logic, optionally forcing an immediate layout of pending layout requests. |
| [RtlTranslateAlignment(ContentAlignment)](http://msdn.microsoft.com/en-us/library/y2f831he.aspx) | Converts the specified [ContentAlignment](http://msdn.microsoft.com/en-us/library/system.drawing.contentalignment.aspx) to the appropriate [ContentAlignment](http://msdn.microsoft.com/en-us/library/system.drawing.contentalignment.aspx) to support right-to-left text. |
| [RtlTranslateAlignment(HorizontalAlignment)](http://msdn.microsoft.com/en-us/library/1fe717f0.aspx) | Converts the specified [HorizontalAlignment](http://msdn.microsoft.com/en-us/library/system.windows.forms.horizontalalignment.aspx) to the appropriate [HorizontalAlignment](http://msdn.microsoft.com/en-us/library/system.windows.forms.horizontalalignment.aspx) to support right-to-left text. |
| [RtlTranslateAlignment(LeftRightAlignment)](http://msdn.microsoft.com/en-us/library/bw337a96.aspx) | Converts the specified [LeftRightAlignment](http://msdn.microsoft.com/en-us/library/system.windows.forms.leftrightalignment.aspx) to the appropriate [LeftRightAlignment](http://msdn.microsoft.com/en-us/library/system.windows.forms.leftrightalignment.aspx) to support right-to-left text. |
| [RtlTranslateContent](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.rtltranslatecontent.aspx) | Converts the specified [ContentAlignment](http://msdn.microsoft.com/en-us/library/system.drawing.contentalignment.aspx) to the appropriate [ContentAlignment](http://msdn.microsoft.com/en-us/library/system.drawing.contentalignment.aspx) to support right-to-left text. |
| [RtlTranslateHorizontal](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.rtltranslatehorizontal.aspx) | Converts the specified [HorizontalAlignment](http://msdn.microsoft.com/en-us/library/system.windows.forms.horizontalalignment.aspx) to the appropriate [HorizontalAlignment](http://msdn.microsoft.com/en-us/library/system.windows.forms.horizontalalignment.aspx) to support right-to-left text. |
| [RtlTranslateLeftRight](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.rtltranslateleftright.aspx) | Converts the specified [LeftRightAlignment](http://msdn.microsoft.com/en-us/library/system.windows.forms.leftrightalignment.aspx) to the appropriate [LeftRightAlignment](http://msdn.microsoft.com/en-us/library/system.windows.forms.leftrightalignment.aspx) to support right-to-left text. |
| [Scale(SizeF)](http://msdn.microsoft.com/en-us/library/ms158413.aspx) | Scales the control and all child controls by the specified scaling factor. |
| [ScaleControl](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.scalecontrol.aspx) | Scales a control's location, size, padding and margin. |
| [ScaleCore](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.scalecore.aspx) | Infrastructure. This method is not relevant for this class. |
| [Select](http://msdn.microsoft.com/en-us/library/7wt11hea.aspx) | Activates the control. |
| [Select(Boolean, Boolean)](http://msdn.microsoft.com/en-us/library/hb97bya5.aspx) | Activates a child control. Optionally specifies the direction in the tab order to select the control from. |
| [SelectNextControl](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.selectnextcontrol.aspx) | Activates the next control. |
| [SendToBack](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.sendtoback.aspx) | Sends the control to the back of the z-order. |
| [SetAutoSizeMode](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.setautosizemode.aspx) | Sets a value indicating how a control will behave when its [AutoSize](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.autosize.aspx) property is enabled. |
| [SetBounds(Int32, Int32, Int32, Int32)](http://msdn.microsoft.com/en-us/library/z0tayb1b.aspx) | Sets the bounds of the control to the specified location and size. |
| [SetBounds(Int32, Int32, Int32, Int32, BoundsSpecified)](http://msdn.microsoft.com/en-us/library/ad5ah3k5.aspx) | Sets the specified bounds of the control to the specified location and size. |
| [SetBoundsCore](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.setboundscore.aspx) | Performs the work of setting the specified bounds of this control. |
| [SetClientSizeCore](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.setclientsizecore.aspx) | Sets the size of the client area of the control. |
| [SetStyle](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.setstyle.aspx) | Sets a specified [ControlStyles](http://msdn.microsoft.com/en-us/library/system.windows.forms.controlstyles.aspx) flag to either true or false. |
| [SetTopLevel](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.settoplevel.aspx) | Sets the control as the top-level control. |
| [SetVisibleCore](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.setvisiblecore.aspx) | Sets the control to the specified visible state. |
| [Show](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.show.aspx) | Displays the control to the user. |
| [SizeFromClientSize](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.sizefromclientsize.aspx) | Determines the size of the entire control from the height and width of its client area. |
| [SuspendLayout](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.suspendlayout.aspx) | Temporarily suspends the layout logic for the control. |
| [ToString](http://msdn.microsoft.com/en-us/library/z819d1t5.aspx) | Returns a [String](http://msdn.microsoft.com/en-us/library/system.string.aspx) containing the name of the [Component](http://msdn.microsoft.com/en-us/library/system.componentmodel.component.aspx), if any. This method should not be overridden. (Inherited from [Component](http://msdn.microsoft.com/en-us/library/system.componentmodel.component.aspx).) |
| [Update](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.update.aspx) | Causes the control to redraw the invalidated regions within its client area. |
| [UpdateBounds](http://msdn.microsoft.com/en-us/library/zxe4zss7.aspx) | Updates the bounds of the control with the current size and location. |
| [UpdateBounds(Int32, Int32, Int32, Int32)](http://msdn.microsoft.com/en-us/library/t83c26f8.aspx) | Updates the bounds of the control with the specified size and location. |
| [UpdateBounds(Int32, Int32, Int32, Int32, Int32, Int32)](http://msdn.microsoft.com/en-us/library/37zzd7tc.aspx) | Updates the bounds of the control with the specified size, location, and client size. |
| [UpdateStyles](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.updatestyles.aspx) | Forces the assigned styles to be reapplied to the control. |
| [UpdateZOrder](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.updatezorder.aspx) | Updates the control in its parent's z-order. |
| [WndProc](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.wndproc.aspx) | Processes Windows messages. |

## Control Events:

|  |  |
| --- | --- |
| *Name* | *Description* |
| [AutoSizeChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.autosizechanged.aspx) | Infrastructure. This event is not relevant for this class. |
| [BackColorChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.backcolorchanged.aspx) | Occurs when the value of the [BackColor](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.backcolor.aspx) property changes. |
| [BackgroundImageChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.backgroundimagechanged.aspx) | Occurs when the value of the [BackgroundImage](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.backgroundimage.aspx) property changes. |
| [BackgroundImageLayoutChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.backgroundimagelayoutchanged.aspx) | Occurs when the [BackgroundImageLayout](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.backgroundimagelayout.aspx) property changes. |
| [BindingContextChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.bindingcontextchanged.aspx) | Occurs when the value of the [BindingContext](http://msdn.microsoft.com/en-us/library/system.windows.forms.bindingcontext.aspx) property changes. |
| [CausesValidationChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.causesvalidationchanged.aspx) | Occurs when the value of the [CausesValidation](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.causesvalidation.aspx) property changes. |
| [ChangeUICues](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.changeuicues.aspx) | Occurs when the focus or keyboard user interface (UI) cues change. |
| [Click](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.click.aspx) | Occurs when the control is clicked. |
| [ClientSizeChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.clientsizechanged.aspx) | Occurs when the value of the [ClientSize](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.clientsize.aspx) property changes. |
| [ContextMenuChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.contextmenuchanged.aspx) | Occurs when the value of the [ContextMenu](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.contextmenu.aspx) property changes. |
| [ContextMenuStripChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.contextmenustripchanged.aspx) | Occurs when the value of the [ContextMenuStrip](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.contextmenustrip.aspx) property changes. |
| [ControlAdded](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.controladded.aspx) | Occurs when a new control is added to the [Control.ControlCollection](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.controlcollection.aspx). |
| [ControlRemoved](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.controlremoved.aspx) | Occurs when a control is removed from the [Control.ControlCollection](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.controlcollection.aspx). |
| [CursorChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.cursorchanged.aspx) | Occurs when the value of the [Cursor](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.cursor.aspx) property changes. |
| [Disposed](http://msdn.microsoft.com/en-us/library/system.componentmodel.component.disposed.aspx) | Occurs when the component is disposed by a call to the [Dispose](http://msdn.microsoft.com/en-us/library/system.componentmodel.component.dispose.aspx) method. (Inherited from [Component](http://msdn.microsoft.com/en-us/library/system.componentmodel.component.aspx).) |
| [DockChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.dockchanged.aspx) | Occurs when the value of the [Dock](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.dock.aspx) property changes. |
| [DoubleClick](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.doubleclick.aspx) | Occurs when the control is double-clicked. |
| [DragDrop](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.dragdrop.aspx) | Occurs when a drag-and-drop operation is completed. |
| [DragEnter](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.dragenter.aspx) | Occurs when an object is dragged into the control's bounds. |
| [DragLeave](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.dragleave.aspx) | Occurs when an object is dragged out of the control's bounds. |
| [DragOver](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.dragover.aspx) | Occurs when an object is dragged over the control's bounds. |
| [EnabledChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.enabledchanged.aspx) | Occurs when the [Enabled](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.enabled.aspx) property value has changed. |
| [Enter](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.enter.aspx) | Occurs when the control is entered. |
| [FontChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.fontchanged.aspx) | Occurs when the [Font](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.font.aspx) property value changes. |
| [ForeColorChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.forecolorchanged.aspx) | Occurs when the [ForeColor](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.forecolor.aspx) property value changes. |
| [GiveFeedback](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.givefeedback.aspx) | Occurs during a drag operation. |
| [GotFocus](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.gotfocus.aspx) | Occurs when the control receives focus. |
| [HandleCreated](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.handlecreated.aspx) | Occurs when a handle is created for the control. |
| [HandleDestroyed](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.handledestroyed.aspx) | Occurs when the control's handle is in the process of being destroyed. |
| [HelpRequested](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.helprequested.aspx) | Occurs when the user requests help for a control. |
| [ImeModeChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.imemodechanged.aspx) | Occurs when the [ImeMode](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.imemode.aspx) property has changed. |
| [Invalidated](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.invalidated.aspx) | Occurs when a control's display requires redrawing. |
| [KeyDown](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.keydown.aspx) | Occurs when a key is pressed while the control has focus. |
| [KeyPress](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.keypress.aspx) | Occurs when a key is pressed while the control has focus. |
| [KeyUp](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.keyup.aspx) | Occurs when a key is released while the control has focus. |
| [Layout](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.layout.aspx) | Occurs when a control should reposition its child controls. |
| [Leave](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.leave.aspx) | Occurs when the input focus leaves the control. |
| [LocationChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.locationchanged.aspx) | Occurs when the [Location](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.location.aspx) property value has changed. |
| [LostFocus](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.lostfocus.aspx) | Occurs when the control loses focus. |
| [MarginChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.marginchanged.aspx) | Occurs when the control's margin changes. |
| [MouseCaptureChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.mousecapturechanged.aspx) | Occurs when the control loses mouse capture. |
| [MouseClick](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.mouseclick.aspx) | Occurs when the control is clicked by the mouse. |
| [MouseDoubleClick](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.mousedoubleclick.aspx) | Occurs when the control is double clicked by the mouse. |
| [MouseDown](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.mousedown.aspx) | Occurs when the mouse pointer is over the control and a mouse button is pressed. |
| [MouseEnter](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.mouseenter.aspx) | Occurs when the mouse pointer enters the control. |
| [MouseHover](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.mousehover.aspx) | Occurs when the mouse pointer rests on the control. |
| [MouseLeave](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.mouseleave.aspx) | Occurs when the mouse pointer leaves the control. |
| [MouseMove](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.mousemove.aspx) | Occurs when the mouse pointer is moved over the control. |
| [MouseUp](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.mouseup.aspx) | Occurs when the mouse pointer is over the control and a mouse button is released. |
| [MouseWheel](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.mousewheel.aspx) | Occurs when the mouse wheel moves while the control has focus. |
| [Move](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.move.aspx) | Occurs when the control is moved. |
| [PaddingChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.paddingchanged.aspx) | Occurs when the control's padding changes. |
| [Paint](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.paint.aspx) | Occurs when the control is redrawn. |
| [ParentChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.parentchanged.aspx) | Occurs when the [Parent](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.parent.aspx) property value changes. |
| [PreviewKeyDown](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.previewkeydown.aspx) | Occurs before the [KeyDown](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.keydown.aspx) event when a key is pressed while focus is on this control. |
| [QueryAccessibilityHelp](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.queryaccessibilityhelp.aspx) | Occurs when [AccessibleObject](http://msdn.microsoft.com/en-us/library/system.windows.forms.accessibleobject.aspx) is providing help to accessibility applications. |
| [QueryContinueDrag](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.querycontinuedrag.aspx) | Occurs during a drag-and-drop operation and enables the drag source to determine whether the drag-and-drop operation should be canceled. |
| [RegionChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.regionchanged.aspx) | Occurs when the value of the [Region](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.region.aspx) property changes. |
| [Resize](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.resize.aspx) | Occurs when the control is resized. |
| [RightToLeftChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.righttoleftchanged.aspx) | Occurs when the [RightToLeft](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.righttoleft.aspx) property value changes. |
| [SizeChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.sizechanged.aspx) | Occurs when the [Size](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.size.aspx) property value changes. |
| [StyleChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.stylechanged.aspx) | Occurs when the control style changes. |
| [SystemColorsChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.systemcolorschanged.aspx) | Occurs when the system colors change. |
| [TabIndexChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.tabindexchanged.aspx) | Occurs when the [TabIndex](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.tabindex.aspx) property value changes. |
| [TabStopChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.tabstopchanged.aspx) | Occurs when the [TabStop](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.tabstop.aspx) property value changes. |
| [TextChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.textchanged.aspx) | Occurs when the [Text](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.text.aspx) property value changes. |
| [Validated](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.validated.aspx) | Occurs when the control is finished validating. |
| [Validating](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.validating.aspx) | Occurs when the control is validating. |
| [VisibleChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.visiblechanged.aspx) | Occurs when the [Visible](http://msdn.microsoft.com/en-us/library/system.windows.forms.control.visible.aspx) property value changes. |