**Using the Combo Box Control**

A combo box control combines the features of a text box and a list box. This control allows the user to select an item either by typing text into the combo box, or by selecting it from the list.

Combo boxes present a list of choices to the user. If the number of items exceeds what can be displayed in the combo box, scroll bars will automatically appear on the control. The user can then scroll up and down or left to right through the list.

## When to Use a Combo Box Instead of a List Box

Generally, a combo box is appropriate when there is a list of *suggested* choices, and a list box is appropriate when you want to limit input to what is on the list. A combo box contains an edit field, so choices not on the list can be typed in this field.

In addition, combo boxes save space on a form. Because the full list is not displayed until the user clicks the down arrow (except for Style 1, which is always dropped down), a combo box can easily fit in a small space where a list box would not fit.

## Combo Box Styles

There are three combo box styles. Each style can be set at design time and uses values, or equivalent Visual Basic constants, to set the style of the combo box.

### Drop-down Combo Box

With the default setting (Style = 0 – Dropdown Combo), a combo box is a drop-down combo box. The user can either enter text directly (as in a text box) or click the detached arrow at the right of the combo box to open a list of choices. Selecting one of the choices inserts it into the text portion at the top of the combo box. The user also can open the list by pressing ALT+ DOWN ARROW when the control has the focus.

### Simple Combo Box

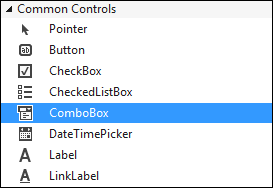
Setting the Style property of a combo box to 1 – Simple Combo specifies a simple combo box in which the list is displayed at all times. To display all entries in the list, you must draw the list box large enough to display the entries. A vertical scroll bar is automatically inserted when there are more entries than can be displayed. The user can still enter text directly or select from the list. As with a drop-down combo box, a simple combo box also allows users to enter choices not on the list.

### Drop-down List Box

A drop-down list box (Style = 2 – Dropdown List) is like a regular list box — it displays a list of items from which a user must choose. Unlike list boxes, however, the list is not displayed until you click the arrow to the right of the box. The key difference between this and a drop-down combo box is that the user can't type into the box, he can only select an item from the list. Use this type of list box when space is at a premium.

# Add a Combo Box to a VB form

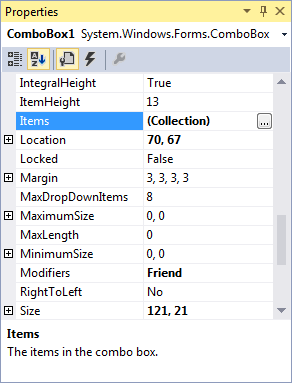
Create a new project for this section. Add a button to your new form. Then, locate the Combo Box on the Visual Basic toolbar. It looks like this:



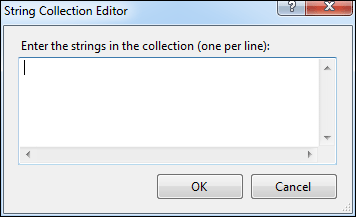
Double click the icon to add a Combo Box to your form. Or click once with the left hand mouse button, and then draw one on the form.

A combo box is a way to limit the choices your user will have. When a black down-pointing arrow is clicked, a drop down list of items appears. The user can then select one of these options. So let's set it up to do that.

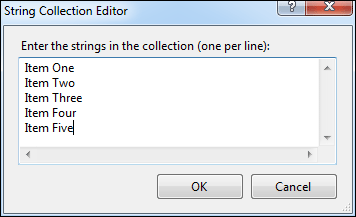
* Click on your Combo Box to select it. Then locate the Item property from the Properties Box:



* Click the grey button, as above. The one with the three dots in it. When you do, you'll get the following box popping up:

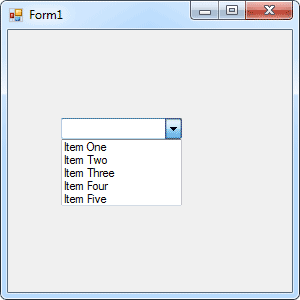


* To use the String Collection Editor, type an item and press Return (it's just like a normal textbox. Each item will be one item in your drop-down box.)
* Enter five items, as in the image below:



* Then click the OK button at the bottom.

The Editor will close, and it will look like nothing has happened. However, run your program and test out your new Combo Box. You should have something like this:



You now need to know how to get values from the list. Once you know how to get a value from the list, you can put the value into a variable and test it with some Conditional logic.

Getting a value from a Combo Box is fairly straightforward, because it acts just like a Textbox. A Textbox has a Text property, and so does a Combo Box. To get a value from a Textbox, you would code like this

**MyVariable = Textbox1.Text**

Whatever is in the Textbox will be transferred to the variable called MyVariable. The process is exactly the same for a Combo Box. The code is like this:

**MyVariable = Combobox1.Text**

Now we are transferring whatever is selected from the Combo Box to the variable called**MyVariable**.

Let's try it. Double click the button you added to your form. This will open the code window. Then enter the following code for the button:

**Dim MyVariable as String**

**MyVariable = Combobox1.Text**

**MessageBox.Show( MyVariable )**

Run your program. When the program is running, select an item from your Combo Box. Then click your button. Whatever was in the Combo Box window should have ended up in the Message Box.

And that's all there is to getting a value from a Combo Box - just access its Text Property and pass it to a variable.

Finally, the Combo Box has a **DropDownStyle** property. Locate this property and you'll notice its value has a drop down box. The box contains three different Combo Box styles to choose from. Experiment with all three and see how they differ.