

**Mr. Giansante**

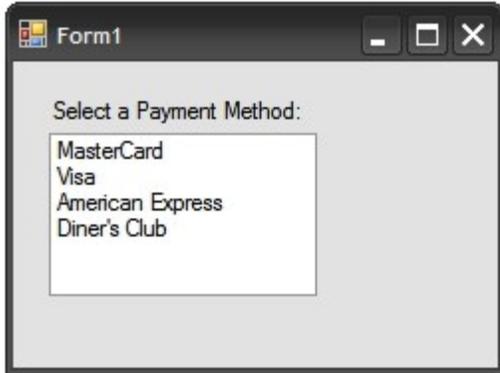


# **Visual Basic**

**ComboBox and ListBox  
Controls**

**August 2016**

# ListBox Control



## Step 1 - Adding the Items

Items can be added as illustrated below. Items appear in the order that you insert them (unless you have the Sorted property set to True).

In Form | Load ...

```
ListBox1.Items.Add("MasterCard")
ListBox1.Items.Add("Visa")
ListBox1.Items.Add("American Express")
ListBox1.Items.Add("Diner's Club")

ListBox1.SelectedIndex = 2
```

The **SelectedIndex** property determines which item is showing when the program starts. Note that the items are numbered, in the order that they appear, starting with "0" (zero). So, in the above example, with the SelectedIndex property being set to "2", "American Express" would be the default item.

## Step 2 - Determining which Item the user selected

When the program is running, you can respond to the user selecting an option as follows:

In ListBox1 | Click ...

```
If ListBox1.SelectedItem = "MasterCard" Then
    MessageBox.Show("MasterCard")
End If
```

Another way to accomplish this is to use the **SelectedIndex** property.

```
If ListBox1.SelectedIndex = 0 Then
    MessageBox.Show("MasterCard")
End If
```

## The "Sorted" Property

If you want the items in the ListBox to be in alphabetical (or alphanumerical) order, set the **Sorted** property to True. Otherwise, the items will appear in the order that they were added to the ListBox.

## How Many Items?

To determine how many items are in the ListBox, use the following:

```
ListBox1.Items.Count
```

## Adding Items in a Specific Location

Use the **Insert** method.

```
ListBox1.Items.Insert(2, "Discovery Card")
```

The number specified will be the index of the new item. So, in this case, with an index value of "2", the item would be placed third in the ListBox (remember that the first item has an index of "0").

## Erasing All Items in a ListBox

The **Clear** method is used to remove all items from a ListBox.

```
ListBox1.Items.Clear
```

## Removing Specific Items from the ListBox

The **Remove** and **RemoveAt** methods are used to remove items from a ListBox.

```
ListBox1.Items.Remove("MasterCard")
... Or ...
ListBox1.Items.RemoveAt(1)
```

# ComboBox Control

The ComboBox Control functions almost identically to the ListView Control. The main difference is how the choices are presented to the user.

## DropDownStyle Property

You can control the behaviour of the ComboBox by using the **DropDownStyle** property. It can have the following three values:

### Simple

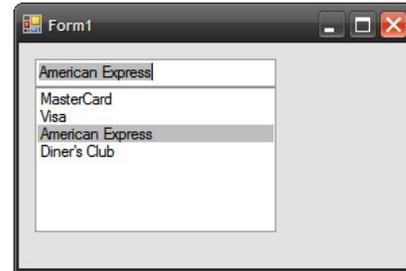
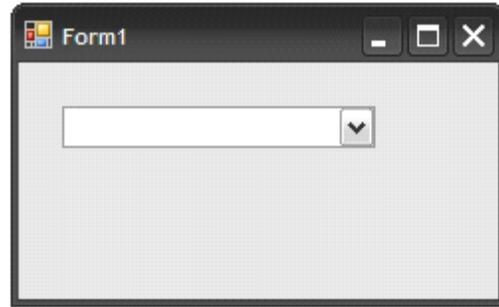
Functions similar to a regular ListView control but the selected item appears at the top.

### DropDown

The user can type in TextBox part.

### DropDownList

The user cannot type in TextBox part.



## Step 1 - Adding the Items

Items can be added as illustrated below. Items appear in the order that you insert them (unless you have the Sorted property set to True).

In Form | Load ...

```
ComboBox1.Items.Add("1968")
ComboBox1.Items.Add("1969")
ComboBox1.Items.Add("1970")
ComboBox1.Items.Add("1971")
ComboBox1.Items.Add("1972")

ComboBox1.SelectedIndex = 2
```

The **SelectedIndex** property determines which item is showing when the program starts. Note that the items are numbered, in the order that they appear, starting with "0" (zero). So, in the above example, with the SelectedIndex property being set to "2", "1970" would be the default item.

## Step 2 - Determining which Item the user selected

When the program is running, you can respond to the user selecting an option as follows:

In ComboBox1 | SelectedIndexChanged ...

```
Label1.Text = ComboBox1.Items(ComboBox1.SelectedIndex)
```

# ComboBox Exercises

---

\_\_\_\_\_

Name

\_\_\_\_\_

Date

The code to add items to a ComboBox control (named Combo1) should be placed in

\_\_\_\_\_ | \_\_\_\_\_

If you want the items in the ComboBox to be listed in alphabetical order, you should set the

\_\_\_\_\_ property to **True**.

If you don't want the user to be able to type in the TextBox part of the ComboBox, you should set

the \_\_\_\_\_ property to \_\_\_\_\_.

Write the code required to place the words "Red", "Blue", "Green" and "Yellow" in a ComboBox Control (named ComboBox1).

Assuming you have the ComboBox Control described above, write the code required to make "Green" the value that is displayed when the program starts (ie. the default value).

Assuming you have the ComboBox described above, write the code that will place a message in a Label Control (named Label1) that reads "You choose Red", "You choose Blue", etc.

**Note:** This can be accomplished with only one line of code.

Write the code to place all the available fonts in a ComboBox Control (named Combo1)

In conjunction with the above example, write the code that will change the font of a TextBox Control (named TextBox1) to the font the user selects in the ComboBox. **Hint:** It should only be one line.