

LIST and COMBO BOXES

- The list box control and the combo box control are useful user input controls. These controls both offer the user a choice of a list of items.

LIST AND COMBO BOX PROPERTIES

- **Name** - Listbox object names should begin with **lst**.
- **Name** - Combobox object names should begin with **cbo**.
- **Items** - Used to display the String Collection Editor dialog box where a set of strings is typed.
- **SelectedItem** - Is the selected List or Combo Box item. This property is available only at run-time.
- **SelectedIndex** - Is the index of the selected list item. This property is available only at run-time.
- **Sorted** - Is set to True to display the list items in alphabetical order.
- **Text** - Text of currently selected item in list box.

Working with Selected Items

- The easiest method to retrieve a single, selected item is to look at the control's **Text** property. This will give you the text of the currently selected item in the list. If no items are currently selected then the Text property will contain an empty string. Similarly, you can use the **SelectedItem** property to retrieve the same value, but you have to cast the result to a string before assigning to a variable.

```
String lastName = lstSurname.text;  
String lastName = (String)lstSurname.SelectedItem;
```

Indexed Items

- The items in a list box or combo box have an index value, with 0 being the index of the first item, 1 the index of the next, and so on. In your program you can get or set the currently selected item by reading or assigning the **SelectedIndex** property. If nothing in the control is selected, the property will be a -1.

```
//gets index of currently selected item, or -1 if nothing is selected  
  
int index = lstSurname.SelectedIndex;  
if (index != -1)    //safety check  
{  
    //get the text selected using the index  
    String lastName = (String)lstSurname.items[index];  
    //show the text we retrieved  
    MessageBox.Show(lastName);  
}
```

Add Items

- The `listbox.Items.Add(item);` method is used to add an item to a list at run time.
- The `combobox.Items.Add(item);` method is used to add an item to a combo box at run time.

Remove Items

- The `listbox.Items.Remove(item);` method is used to delete a specified item from the list.
- The `listbox.Items.RemoveAt(index);` method is used to delete a specified item using their index value.

Clear Items

- The `listbox.Items.Clear();` method is used to delete the contents of the list box
- The `listbox.ClearSelected();` will deselect the item during runtime