**ComboBox**

A ComboBox object combines a text box and a list box. The ComboBox control has the properties:

* Name
* Items
* Text
* Sorted
* SelectedItem
* SelectedIndex

A ComboBox object name should begin with the prefix cbo. The items in the list box of the Combo box have an index value beginning with 0. Because a user can type or select a value in a combo box, both TextChanged and SelectedIndexChanged events need to be considered. The ComboBox control class also contains:

* Items.Add()
* Items.Removed()
* Items.Clear()

Methods for controlling the contents of the list box part of the combo box are at run time.

An object that has focus receives the user input. An ampersand (&) may be used in an object’s Test property to define an access key. The tab order for the objects in an application can be set with the TabIndex property. The Enabled property can be set to False so that the user cannot select an object. The Visible property can be used to hide and object.

A combo box displays a text box for typing an entry and an arrow that can be clicked to choose an item from a list. A scroll bar automatically appears in the list when necessary. For example, the Tuition Calculator application can be modified to include a combo box for the user to enter or select the number of credit hours.

Tuition Calculator

Course Level

Calculate

Undergraduate

Graduate

Thesis and Dissertation

Credit Hours

**9**

A ComboBox has the properties:

* **Name** identifies a control for the programmer. It is good programming style to begin ComboBox object names with cbo
* **Items** contains the …button that is clicked to display the String Collection Editor dialog box where a set of strings is typed for the list box of the combo box.
* **Text** is the text displayed in the text box of the combobox.
* **Sorted** is set to True to display the list items in alphabetical order. This property is available only in the Design window.
* **SelectedItem** is the currently selected list 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.

The items in the list box of the combo box have an index value, with 0 being the index of the first item, 1 the index of the next, and so on. An index value of -1 means that none of the list items are selected. The Text property contains the item typed by the user in the text box.

The entry in a combo box changes when the user either types a value or selects a value from the list. Typing a value raised a TextChanged event, and selecting a value raised a SelectedIndexChanged event. Two different event procedures can be written for these events or a single event procedure can be written to handle both events. An If…Then…Else statement can be used to determine if a value ws entered or an item selected as in the statement below:

If Me.cboCreditHours.SelectedIndex >=0 Then

MessageBox.Show(Me.cboCreditHours.SelectedItem)

Else

MessageBox.Show(Me.cboCreditHours.SelectedItem)

End If

The ComboBox control class also contains methods for adding and clearing list items at run time. The Items.Add() method is used to add an item to a list at run time and takes the form:

cboControl.Items.Add(item)

cboControl is the name of the combo box object and Item is a value or a string. The item is added to the end of the list, or in the proper position if the list is sorted. The following statement demonstrates Items.Add():

Me.cboCreditHours.Items.Add(“18”)

The Items.Remove() method is used to delete a specified item from the list and takes the form:

cboControl.Items.Remove(item)

cboControl is the name of the combo box object and Item is a value or a string.

The Items.Clear() method is used to delete the contents of the list in the combobox takes the form:

cboControl.Items.Clear()

cboConlrol is the name of the combo box object.

**ComboBox Assignment**

1. **Open The Tuition Calculator Project**
2. **Modify the Interface**
   1. Refer to the form below to add, position and size a label and combo box object. Use the table to set properties.

Tuition Calculator

Course Level

Calculate

Undergraduate

Graduate

Thesis and Dissertation

Credit Hours

**9**

|  |  |  |
| --- | --- | --- |
| **Object** | **Name** | **Text** |
| Label1 | lblCreditHours | Credit Hours: |
| ComboBox1 | cboCreditHours | *empty* |

* 1. Use the ComboBox Items property to add the list items 3,6,9,12, and 15 to the combo box.

1. **Modify the Application Code**
   1. Display the code windo
   2. Modify the btnCalculate\_Click event procedure:

Const decUnergraduatePerHour As Decimal = 75

Const decGraduatePerHour As Decimal =145

Const decThesisPerHour As Decimal =160

Dim intCreditHours As Integer

DimdecTuition As Decimal

If Me..cboCreditHours.SelectedIndex .=0 Then

intCreditHours = Val(Me.cboCreditHours.SelectedItem)

Else

intCreditHours = Val(Me.cboCreditHours.Text)

End If

Select Case

1. **Run the Application**