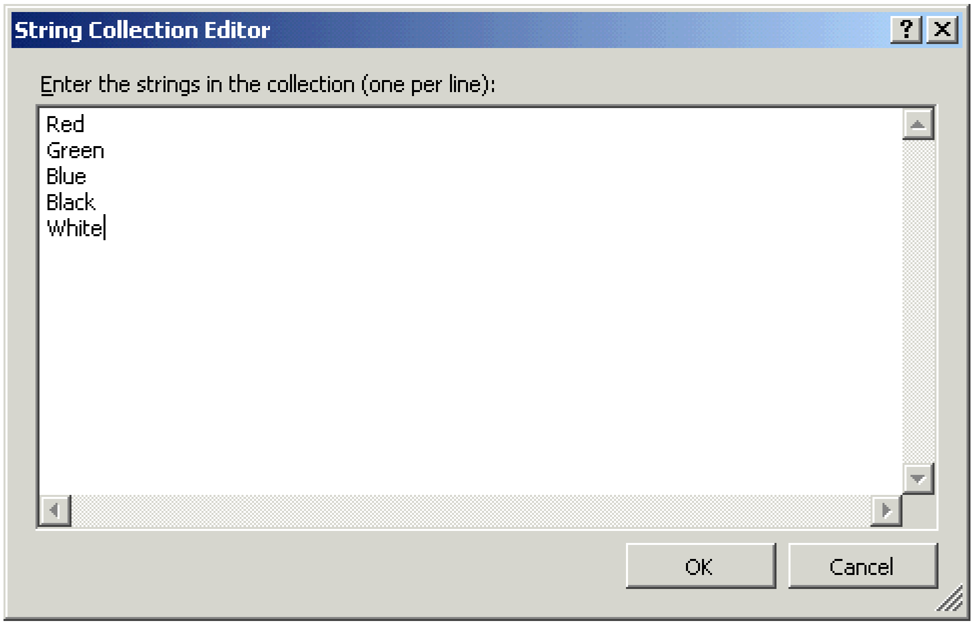
THE COMBOBOX CONTROL

The **ComboBox** control combines **TextBox** features with a drop-down list, a component that contains a list from which a value can be selected. A combo box usually appears as a text box with a down arrow to its right. By default, the user can enter text into the text box or click the down arrow to display a list of predefined items. If the user chooses an element from this list, that element is displayed in the text box. If the list contains more elements than can be displayed in the drop-down list, a scrollbar appears. The maximum number of items that a drop-down list can display at one time is set by the property **MaxDropDownItems**.

The name of a **ComboBox** control should begin with the three-letter prefix **cbo**.

To add elements to the drop-down list of a combo box, you can use the **Items** property and enter each of the values. For example, if I wanted to create a combo box that provided the user with a list of colour options, I would click the **Items** property and then enter the values in the following **String Collection Editor** window:



Enter the strings in the collection (one per line):

```
Red
Green
Blue
Black
White
```

Alternatively, you can add elements to the **Items** collection programmatically by using the **Add** method, as follows:

```
cboGold.Items.Add("Red")
cboGold.Items.Add("Green")
cboGold.Items.Add("Blue")
cboGold.Items.Add("Black")
cboGold.Items.Add("White")
```

The **DropDownStyle** property controls the appearance and functionality of the combo box. The available properties are as follows:

Simple: This type of combo box does not display a drop-down arrow. Instead, a scrollbar appears next to the control, allowing the user to select a choice from theh list. The user can also type in a selection.

DropDown: This type of combo box displays a drop-down list when the down arrow is clicked. The user can type a new item in the combo box.

DropDownList: This type of combo box displays a drop-down list but does not allow the user to type in the combo box.

When a combo box is clicked by the user, a **SelectedIndexChanged** event occurs. In order to determine which element the user selected, you will need to use either the **SelectedIndex** or **SelectedItem** property.

The **SelectedIndex** property returns the index of the selected item, or -1 if there is no selected item. So, using the above example, if the user selected the colour **Red** from the combo box, the value 0 would be returned since it's the first element in my collection. If the user selected the colour **Green**, 1 would be returned.

The **SelectedItem** property returns a reference to the selected item. So, staying with the same example, if the user selected the colour **Black**, the value **"Black"** would be returned.