

Reviewing the Stylebits Parameters

The four parameters of stylebits are AddBit, RemoveBit, AddExtendedBit, RemoveExtendedBit. For a review of these four parameters, and an introduction to Stylebits in general, please view [Stylebits - Windows](#).

A combobox is a combination listbox and textbox. Comboboxes are designed to save space by only showing the listbox when needed. When the arrow button is clicked, the listbox drops down. The user can then mouse click an item from the loaded array. The selected item is then printed in the textbox portion of the combobox. In the standard combobox, it is also possible for the user to type an entry in the textbox that is not already a member of the array. Unlike other controls, comboboxes do not respond well to changes with border stylebits. For this reason, the ExtendedBits aren't all that useful with comboboxes. The stylebits `_CBS_LOWERCASE` and `_CBS_UPPERCASE` work quite well with comboboxes. Even though `_CBS_SORT` is a documented Window's constant, I have never been able to sort a combobox array with the stylebit alone. Fortunately alphabetizing is easily accomplished with Liberty BASIC's native `SORT` and `RELOAD` commands. The stylebits that are most effective with comboboxes are `_CBS_DROPDOWN`, `_CBS_DISABLENOSCROLL`, `_CBS_DROPDOWNLIST`, and `_CBS_AUTOHSCROLL`.

The Listbox and the Textbox

`_CBS_DROPDOWN` as an AddBits is the standard combobox setting. This allows the dropdown listbox. Including `_CBS_DROPDOWN` in RemoveBits will keep the listbox visible at all times, rather than dropping down when needed. The textbox allows user entry. The listbox automatically scrolls to an item which satisfies the entry being typed. Try typing in "July", then backspace the "y" and "l" and continue with "ne" to see this scrolling in action.

- **`_CBS_DISABLENOSCROLL`** draws a disabled scroll bar rather than no scroll bar when the dropdown listbox displays all array items. This gives a more polished look to the listbox. `_CBS_DISABLEDNOSCROLL` looks especially nice when coupled as an AddBits with the RemoveBits `_CBS_DROPDOWN`.
- **`_CBS_DROPDOWNLIST`** used as an AddBits will cause the textbox to act as statictext rather than an edit control. This will prevent users from typing into the textbox component of the combobox. This is especially useful if you want your user to select only what's available in the combobox array. Each keypress displays a member of the array with that initial character. Repeating the character brings up the next array item with the same initial letter in alphabetical order. You have probably seen this when prompted for your state abbreviation. Repeatedly pressing the 'M' brings up "Maine, Maryland, Massachusetts, etc.
- **`_CBS_AUTOHSCROLL`** is another standard combobox default. The textbox scrolls when the user types in characters beyond the visible width of the combobox. By removing `_CBS_AUTOHSCROLL`, you can limit the number of characters typed by the user. This is, of

course, dependent upon the width of the combobox and the size and weight of the font.



Probably the best way to see combobox stylebits effects are to see them in action. In this demo, six comboboxes are each loaded with the same array. Try mouse clicking and keyboard entry with all of them.

```

WindowWidth = 400
WindowHeight = 500
UpperLeftX = Int((DisplayWidth - WindowWidth)/2)
UpperLeftY = Int((DisplayHeight - WindowHeight)/3)

Dim month$(12)
For i = 1 to 12
    Read m$
    month$(i) = m$
Next i

Nomainwin

'Set up a default button
Button #main.default, "", selMonth, UL, -100, -100
'Remove default button from tabstop cycle
Stylebits #main.default, 0, _WS_TABSTOP, 0, 0
'The Standard Combobox
Statictext #main.st1, "Standard", 30, 14, 100, 24
Combobox #main.cb1, month$(), selMonth, 30, 40, 100, 124
'Invisible Combobox; Not very practical.
Statictext #main.st2, "Hidden", 150, 14, 100, 24
Combobox #main.cb2, month$(), selMonth, 150, 40, 100, 124
Stylebits #main.cb2, 0, _WS_VISIBLE, 0, 0
'Listbox stays visible at all times; Textbox accepts user input.
Statictext #main.st3, "Visible Listbox", 270, 14, 100, 24
Combobox #main.cb3, month$(), selMonth, 270, 40, 100, 124
Stylebits #main.cb3, 0, _CBS_DROPDOWN, 0, 0
'Disabled vertical scroll bar when all items are visible
Statictext #main.st4,
"Disabled Unneeded Scrollbar", 30, 130, 100, 54
Combobox #main.cb4, month$(), selMonth, 30, 200, 100, 260
Stylebits #main.cb4, _CBS_DISABLENOSCROLL, _CBS_DROPDOWN, 0, 0
'Textbox doesn't accept user input, but instead selects Combobox item

```

```
'beginning with pressed character; Mouse clicking Textbox drops Listbo
x.
    Statictext #main.st5, "Keypress Initial", 150, 200, 100, 48
    Combobox #main.cb5, month$(), selMonth, 150, 250, 100, 124
    Stylebits #main.cb5, _CBS_DROPDOWNLIST, 0, 0, 0
'Limits user input to number of visible characters
    Statictext #main.st6,
"Limit Number of Characters", 270, 200, 100, 48
    Combobox #main.cb6, month$(), selMonth, 270, 250, 80, 124
    Stylebits #main.cb6, 0, _CBS_AUTOHSCROLL, 0, 0

    Open "Stylebits and the Combobox" for Dialog as #main
    #main "Trapclose [endDemo]"
    #main "Font Times_New_Roman 12 Bold"
    #main.cb2 "![Invisible]"
    #main.cb1 "Setfocus"

    Wait

[endDemo]
    Close #main
    End

Sub selMonth handle$
    If handle$ = "#main.default" Then
        #main.cb1 "Contents? m1$"
        #main.cb2 "Contents? m2$"
        #main.cb3 "Contents? m3$"
        #main.cb4 "Contents? m4$"
        #main.cb5 "Contents? m5$"
        #main.cb6 "Contents? m6$"
        month$ = "The Textboxes Contents" + Chr$(13) + _
            "Combobox 1: ";m1$ + Chr$(13) + _
            "Combobox 2: ";m2$ + Chr$(13) + _
            "Combobox 3: ";m3$ + Chr$(13) + _
            "Combobox 4: ";m4$ + Chr$(13) + _
            "Combobox 5: ";m5$ + Chr$(13) + _
            "Combobox 6: ";m6$ + Chr$(13)
    Else
        #handle$ "Selectionindex? sel"
        month$ = month$(sel)
    End If
    Notice "You chose ";month$
    #main.cb5 "Reload"
End Sub
```

```
Data "January", "February", "March", "April", "May", "June",  
"July"  
Data "August", "September", "October", "November", "December"
```

Dialog Windows, Default Buttons and the TabStop Cycle

The advantage of placing comboboxes on a dialog window is the use of the default button. A dialog window can have a default button that is activated when the user hits the ENTER key even if that button is not the control with focus. The default button must have the .default extension. In the above demo, the default button is used to read the contents of the textbox portion of each combobox. If you tab from one control to another, you will find that every sixth tab press appears to do nothing. The second combobox is not the culprit. Because it's hidden, that combobox is not receiving focus. It is the default button that lies off the window, but still able to receive focus. The stylebits `_WS_TABSTOP` as a `RemoveBits` for the default button removes the button from the tab cycle. The user can now tab through the five comboboxes without skipping a beat.

A List of Stylebits

You can get a list of all [dwStyles](#) and [dwExStyles](#) available with the Stylebits command at the [MSDN Library - Combo Box Styles](#).