

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](#).

Line Wrapping

How can I get text to wrap in a textbox? This question is frequently asked on the Liberty BASIC Forum. Line wrapping is easily achieved with a few stylebits.

```
Nomainwin
WindowWidth=200
WindowHeight=160

text1$ =
"STYLEBITS allows you to change the style of a Liberty BASIC window or
control."
text2$ = "Add to or edit this text."

Textbox #Main.txtbx, 0, 0, 190, 68
Stylebits #Main.txtbx, _WS_
VSCROLL OR _ES_MULTILINE, _ES_AUTOHSCROLL, 0, 0
Statictext #Main, text2$, 5, 80, 190, 30

Open "LineWrap for LB" for Window_nf as #Main
Print #Main, "Trapclose EndDemo"
Print #Main, "Font Times_New_Roman 12 Bold"
Print #Main.txtbx, text1$
Wait

Sub EndDemo handle$
Close #Main
End
End Sub
```

Go ahead and type in your own text. The text will wrap automatically as you type. It is possible to begin a new line in this word wrapping textbox, but you must use CTRL-ENTER rather than just ENTER. If you don't like scroll bars, then use a resizable textbox. Substitute the above stylebits command with

```
Stylebits #main.txtbx, _WS_THICKFRAME OR _ES_MULTILINE, _ES_AUTOHSCROLL
```

, 0, 0

and let the user tug and pull the textbox all over the window. Be careful with this one. If your user shrinks the textbox and drags it behind another control, the textbox may become irretrievable. If you choose this effect, you may want to add some resize handling using the Locate and Refresh commands.

Limiting Text Entry

You can limit text length by omitting the vertical scroll bar. Without the ability to scroll, the textbox will only accept what can be visually seen. By carefully controlling the font size and the height of the textbox, user input can be contained.

```
WindowWidth=200
WindowHeight=160
Nomainwin

text1$ = "Doc and Sleepy and Grumpy and Sneezy and"
text2$ = "Can you name the other 3?"

Textbox #main.txtbx, 0, 0, 190, 68
Stylebit
s #main.txtbx, _ES_MULTILINE, _ES_AUTOHSCROLL OR _ES_AUTOVSCROLL, 0, 0
Statictext #main, text2$, 5, 80, 190, 30

Open "LineWrap for LB" for Window_nf as #main
Print #main, "Trapclose EndDemo"
Print #main, "Font Times_New_Roman 12 Bold"
Print #main.txtbx, text1$

Wait

Sub EndDemo handle$
  Close #main
End
End Sub
```

By removing the multiline stylebit you can control the length of a single line of text input.

```
Stylebits #main.txtbx, 0, _ES_AUTOHSCROLL OR _ES_AUTOVSCROLL, 0, 0
```

The textbox is nice, but I really need a text editor. What stylebits will add word wrapping to the text editor?
Unfortunately, the text editor is a [widget](#) and widgets do not respond well to stylebits. If your program

requires a word wrapping text editor, consider downloading Alyce Watson's [Text Editor by API](#). This API modified textbox works just like a real text editor, it's compatible with Liberty BASIC v3.x as well as v4.x, and, best of all, like so many of [Alyce's Liberty BASIC contributions](#), it's **free**.

Scrolling vs Wrapping

It isn't always desirable to wrap text. If you anticipate text will extend beyond the limits of your textbox but you still want to keep the text on one line, consider adding a horizontal scroll bar.

```
Stylebits #main.txtbx, _WS_HSCROLL, 0, 0, 0
```

Textbox Borders

Textbox borders respond to window style (`_WS_`) stylebits, but may require unique combinations. While

```
Stylebits #main.gb, 0, _WS_BORDER, 0, 0
```

will remove the border of a graphic box, the same effect with a textbox can only be achieved with the removal of an extended style:

```
Stylebits #main.txtbx, 0, _WS_BORDER, 0, _WS_EX_CLIENTEDGE
```

Thanks to

[Mike Bradbury \(See Reply #3\)](#) for posting this solution for removing the textbox border.

How about adding a horizontal scroll bar and removing the border? Here's a different look.

```
Stylebits #main.txtbx, _WS_HSCROLL, _WS_BORDER, 0, _WS_EX_CLIENTEDGE
```

Rather than removing a border, stylebits might be used to draw a more defined border, giving a distinct 3 - D effect. `_WS_EX_DLGMODALFRAME` draws a raised border

```
Stylebits #main.txtbx, 0, 0, _WS_EX_DLGMODALFRAME, 0
```

while `_WS_EX_STATICEDGE` draws a "step into" sunken border

```
Stylebits #main.txtbx, 0, 0, _WS_EX_STATICEDGE, 0
```

Restricting User Input

Edit style (`_ES_`) stylebits can be used to restrict or modify user input.

Some examples include

- Numbers Only (Unfortunately, commas and periods are not accepted. Forget about using this style if decimals are involved.)

```
Stylebits #main.txtbx, _ES_NUMBER, 0, 0, 0
```

- Password (The asterisk is the default. This character can be changed by using the `SetPasswordChar` function. Note that `_ES_MULTILINE` must be included as a `RemoveBit` parameter.)

```
Stylebits #main.txtbx, _ES_PASSWORD, _ES_MULTILINE, 0, 0
```

- Uppercase / Lowercase (Converts all user input to uppercase / lowercase.)

```
Stylebits #main.txtbx, _ES_UPPERCASE, 0, 0, 0
```

```
Stylebits #main.txtbx, _ES_LOWERCASE, 0, 0, 0
```

- Left / Center / Right Justify (Can be used in either single line textboxes or multiline textboxes.)

```
Stylebits #main.txtbx, _ES_LEFT, 0, 0, 0
```

```
Stylebits #main.txtbx, _ES_CENTER, _ES_MULTILINE, 0, 0
```

```
Stylebits #main.txtbx, _ES_RIGHT, 0, 0, 0
```

- Read Only (Will not accept user input. Issuing a `Disable` command later in the program will 'gray out' any text in the textbox, but `Enable` will not reverse the `_ES_READONLY` style.)

```
Stylebits #main.txtbx, _ES_READONLY, 0, 0, 0
```

Demo: Borders, Line Wrapping and Textbox Entry

```
' There is no error catching here to prevent a  
' crash if more than one demo window is  
' opened at a time. Please be sure to close  
' the demo window before opening a new one.
```

```
WindowWidth=400
WindowHeight=400

UpperLeftX = Int((DisplayWidth - WindowWidth)/2)
UpperLeftY = Int((DisplayHeight - WindowHeight)/2) - 14

Nomainwin
text$ = "No stylebits have been added to this textbox."

Textbox #main.txtbx1, 22, 40, 300, 32

    Statictext #main, "_Border Styles_", 38, 100, 130, 20
    Button #main.style1, "Add H Scroll Bar"
, [style1], UL, 30, 130, 130, 32
    Button #main.style2, "Borderless", [style2], UL, 30, 170, 130, 32
    Button #main.style3, "Blend in Window"
, [style3], UL, 30, 210, 130, 32
    Button #main.style4, "Raised Border"
, [style4], UL, 30, 250, 130, 32
    Button #main.style5, "SunkenBorder"
, [style5], UL, 30, 290, 130, 32
    Button #main.style6, "Resizable Border"
, [style6], UL, 30, 330, 130, 32

    Statictext #main, "_Text Entry_", 218, 100, 130, 20
    Button #main.entry1, "Line Wrap 1"
, [entry1], UL, 200, 130, 130, 32
    Button #main.entry2, "Line Wrap 2"
, [entry2], UL, 200, 170, 130, 32
    Button #main.entry3, "Numbers Only"
, [entry3], UL, 200, 210, 130, 32
    Button #main.entry4, "Read Only", [entry4], UL, 200, 250, 130, 32
    Button #main.entry5, "Upper Case", [entry5], UL, 200, 290, 130, 32
    Button #main.entry6, "Stay Selected"
, [entry6], UL, 200, 330, 130, 32

Open "Stylebits Demo" for Window_nf as #main

#main, "Trapclose [endDemo]"
#main, "Font Times_New_Roman 12 Bold"
#main.txtbx1, text$
Wait

[endDemo]
Close #main
```

End

[style1] 'Horizontal Scroll

textBx\$ = "Stylebits #w.tb, _WS_HSCROLL, 0, 0, 0"

Stylebits #w.tb, _WS_HSCROLL, 0, 0, 0

Textbox #w.tb, 22, 40, 200, 48

staticTxt\$ = "Stylebits #w.tb, _WS_HSCROLL, 0, 0, 0" + _
Chr\$(13);Chr\$(13);

"This textbox has no line wrapping, but user can scroll to end." + _

" Be sure to allow extra height to accomodate the Horizontal Scroll B
ar."

Statictext #w, staticTxt\$, 30, 140, 300, 154

Open "Stylebits: Horizontal Scroll" for Window as #w

#w, "Font Times_New_Roman 12 Bold"

#w, "Trapclose [closeDemoWin]"

#w.tb, textBx\$

Wait

[style2] 'Borderless

textBx\$ = "Stylebits #w.tb, 0, _WS_BORDER, 0, _WS_EX_CLIENTEDGE"

Stylebits #w.tb, 0, _WS_BORDER, 0, _WS_EX_CLIENTEDGE

Textbox #w.tb, 22, 40, 300, 32

staticTxt\$ = "0, _WS_BORDER, 0, _WS_EX_CLIENTEDGE" + _
Chr\$(13);Chr\$(13);

"Both _WS_BORDER and _WS_EX_CLIENTEDGE bits " + _
"must be removed."

Statictext #w, staticTxt\$, 30, 140, 300, 154

Open "Stylebits: Borderless" for Window as #w

#w, "Font Times_New_Roman 12 Bold"

#w, "Trapclose [closeDemoWin]"

#w.tb, textBx\$

Wait

[style3] 'Blend in Window

TextboxColor\$ = "Buttonface"

textBx\$ =

"Stylebits #w.tb, _WS_HSCROLL, _WS_BORDER, 0, _WS_EX_CLIENTEDGE"

Stylebits #w.tb, _WS_HSCROLL, _WS_BORDER, 0, _WS_EX_CLIENTEDGE

Textbox #w.tb, 22, 40, 300, 48

staticTxt\$ = "_WS_HSCROLL, _WS_BORDER, 0, _WS_EX_CLIENTEDGE" + _

```
Chr$(13);Chr$(13);
"By changing the TextboxColor$ to Buttonface, and removing " + _
"the borders, the textbox appears to blend into the window. Add the H
orizontal " + _
"Scroll Bar with the _WS_HSCROLL stylebit."
Statictext #w, staticTxt$, 30, 140, 300, 154

Open "Stylebits: Blend in Window" for Window as #w
#w, "Font Times_New_Roman 12 Bold"
#w, "Trapclose [closeDemoWin]"
#w.tb, textBx$
TextboxColor$ = "White" 'Reset to white for next display
Wait

[style4] 'Raised Border
textBx$ = "Stylebits #w, 0, 0, _WS_EX_DLGMODALFRAME, 0"

Stylebits #w.tb, 0, 0, _WS_EX_DLGMODALFRAME, 0
Textbox #w.tb, 22, 40, 300, 32
staticTxt$ = "0, 0, _WS_EX_DLGMODALFRAME, 0" + _
Chr$(13);Chr$(13);
"The textbox has a raised double border. This " + _
"raised double border cannot be resized by the user."
Statictext #w, staticTxt$, 30, 140, 300, 154

Open "Stylebits: Raised Border" for Window as #w
#w, "Font Times_New_Roman 12 Bold"
#w, "Trapclose [closeDemoWin]"
#w.tb, textBx$
Wait

[style5] 'Sunken Border
textBx$ = "Stylebits #w, 0, 0, _WS_EX_STATICEDGE, 0"

Stylebits #w.tb, 0, 0, _WS_EX_STATICEDGE, 0
Textbox #w.tb, 22, 40, 300, 32
staticTxt$ = "0, 0, _WS_EX_STATICEDGE, 0" + _
Chr$(13);Chr$(13);
"The textbox has a sunken double border. This " + _
"gives a crisp 3 - D appearance of increased depth."
Statictext #w, staticTxt$, 30, 140, 300, 154

Open "Stylebits: Sunken Border" for Window as #w
#w, "Font Times_New_Roman 12 Bold"
#w, "Trapclose [closeDemoWin]"
```

```
#w.tb, textBx$
Wait

[style6] 'Resizable Border
textBx$ = "Stylebits #w, _WS_THICKFRAME, 0, 0, 0"

Stylebits #w.tb, _WS_THICKFRAME, 0, 0, 0
Textbox #w.tb, 22, 40, 300, 32
staticTxt$ = "_WS_THICKFRAME, 0, 0, 0" + _
    Chr$(13);Chr$(13);
"The textbox has a raised double border. This " + _

"raised double border can be resized by the user. The native LB " + _
    "commands LOCATE and REFRESH can be used to resize after " + _
    "the window is opened."
Statictext #w, staticTxt$, 30, 140, 300, 154

Open "Stylebits: Raised Border" for Window as #w
#w, "Font Times_New_Roman 12 Bold"
#w, "Trapclose [closeDemoWin]"
#w.tb, textBx$
Wait

[entry1] 'Line Wrap w/ Vertical Scroll Bar
textBx$ =
"Stylebits #w.tb, _WS_VSCROLL OR _ES_MULTILINE, _ES_AUTOHSCROLL, 0, 0"
Stylebits #w.tb, _WS_VSCROLL OR _ES_MULTILINE, _ES_AUTOHSCROLL, 0,
0
Textbox #w.tb, 22, 40, 300, 50
staticTxt$ =
"_WS_VSCROLL OR _ES_MULTILINE, _ES_AUTOHSCROLL, 0, 0" + _
    Chr$(13);Chr$(13);
"To cause a carriage return, you must press CTRL-RETURN. Try it."
Statictext #w, staticTxt$, 30, 140, 300, 154

Open
"Stylebits: Line Wrap w/ Vertical Scroll Bar" for Window as #w
#w, "Font Times_New_Roman 12 Bold"
#w, "Trapclose [closeDemoWin]"
#w.tb, textBx$
Wait

[entry2] 'Line Wrap w/ Resizable Frame
textBx$ =
"Stylebits #w.tb, _THICKFRAME OR _ES_MULTILINE, _ES_AUTOHSCROLL, 0, 0"
Stylebits #w.tb, _WS_THICKFRAME OR _ES_MULTILINE, _ES_AUTOHSCROLL,
```



```
0, 0
    Textbox #w.tb, 22, 40, 280, 50
    staticTxt$ =
"_WS_THICKFRAME OR _ES_MULTINE, _ES_AUTOHSCROLL, 0, 0" + _
    Chr$(13);Chr$(13);
"Th
is textbox can be resized to show as little or as much text as " + _
    "desired. To cause a carriage return, you must press CTRL-
RETURN. Try it."
    Statictext #w, staticTxt$, 30, 140, 300, 154

    Open "Stylebits: Line Wrap w/ Resizable Frame" for Window as #w
    #w, "Font Times_New_Roman 12 Bold"
    #w, "Trapclose [closeDemoWin]"
    #w.tb, textBx$
    Wait

[entry3] 'Numbers Only
    textBx$ = "Stylebits #w, _ES_NUMBER, 0, 0, 0"

    Stylebits #w.tb, _ES_NUMBER, 0, 0, 0
    Textbox #w.tb, 22, 40, 300, 32
    staticTxt$ = "_ES_NUMBER, 0, 0, 0" + _
        Chr$(13);Chr$(13);
"Although any characters may be printed to this textbox " + _

"within the program, the textbox will only accept numerical user input
. " + _
    Chr$(13);
>Delete what's in the textbox and try to enter numerical and " + _
    "non - numerical characters."
    Statictext #w, staticTxt$, 30, 140, 300, 154

    Open "Stylebits: Numbers Only" for Window as #w
    #w, "Font Times_New_Roman 12 Bold"
    #w, "Trapclose [closeDemoWin]"
    #w.tb, textBx$
    Wait

[entry4] 'Read Only
    textBx$ = "Stylebits #w, _ES_READONLY, 0, 0, 0"

    Stylebits #w.tb, _ES_READONLY, 0, 0, 0
    Textbox #w.tb, 22, 40, 300, 32
    staticTxt$ = "_ES_READONLY, 0, 0, 0" + _
        Chr$(13);Chr$(13);
```

"Although any characters may be printed to this textbox " + _

"within the program, the textbox will not accept user input. Unlike the native " + _

+ _

```
"out. Try to change the text."
Statictext #w, staticTxt$, 30, 140, 300, 154
```

```
Open "Stylebits: Read Only" for Window as #w
#w, "Font Times_New_Roman 12 Bold"
#w, "Trapclose [closeDemoWin]"
#w.tb, textBx$
Wait
```

[entry5] 'Uppercase

```
textBx$ = "Stylebits #w, _ES_UPPERCASE, 0, 0, 0"
```

```
Stylebits #w.tb, _ES_UPPERCASE, 0, 0, 0
Textbox #w.tb, 22, 40, 300, 32
staticTxt$ = "_ES_UPPERCASE, 0, 0, 0" + _
Chr$(13);Chr$(13);
```

"Any character typed by the user will be printed in the " + _

"uppercase form.";Chr\$(13);Chr\$(13);

"_ES_LOWERCASE works in the same way."

```
Statictext #w, staticTxt$, 30, 140, 300, 154
```

```
Open "Stylebits: Uppercase" for Window as #w
#w, "Font Times_New_Roman 12 Bold"
#w, "Trapclose [closeDemoWin]"
#w.tb, textBx$
Wait
```

[entry6] 'Stay Selected

```
textBx$ = "Stylebits #w, _ES_NOHIDESEL 0, 0, 0"
```

```
Stylebits #w.tb1, _ES_NOHIDESEL, 0, 0, 0
Textbox #w.tb1, 22, 40, 300, 32
Textbox #w.tb2, 22, 80, 300, 32
staticTxt$ = "_ES_NOHIDESEL, 0, 0, 0" + _
Chr$(13);Chr$(13);
```

"The highlighted text in the upper text box remains " + _

```
"selected even wh
en focus shifts to another control.";Chr$(13);Chr$(13) + _

"The user can override the selection unless the _ES_READONLY " + _
    "stylebit is also applied."
Statictext #w, staticTxt$, 30, 140, 300, 154

Open "Stylebits: Stay Selected" for Window as #w
#w, "Font Times_New_Roman 12 Bold"
#w, "Trapclose [closeDemoWin]"
#w.tb1, textBx$
#w.tb1, "!Selectall"
Wait

[closeDemoWin]
Close #w
Wait
```

Other Stylebits with Textboxes

The best way to see how textboxes can be altered with Stylebits is to experiment. You may well find that other combinations will create additional display and editing styles.

A List of Stylebits

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