

Custom Cursors With LoadCursorFromFile

Janet Terra

[Custom Cursors With LoadCursorFromFile](#) | [Changing the Cursor](#) | [Finding Custom Cursors](#) | [Setting the Cursor in the Control Panel](#) | [SetCursor vs SetSystemCursor](#) | [Static Cursors and Animated Cursors](#) | [Finding System Cursors](#)

Changing the Cursor

Mitchell Kotler wrote two articles for the Liberty BASIC Newsletters about loading Windows cursors using the 16 bit user.dll, "LoadCursor" API. The demo was written for Liberty BASIC v2. As an update to Liberty BASIC v4.x and the 32 bit user32.dll, Gordon Sweet submitted [Changing the Cursor - Two Examples](#) for the Liberty BASIC Newsletter #121. These programs temporarily change the cursor to any of 25 predefined Windows images. It is possible to temporarily change the cursor to a custom designed cursor using the user32.dll, "LoadCursorFromFile" API.

Finding Custom Cursors

This article originally appeared in the Liberty BASIC Newsletters #135 and was based upon information valid for Windows XP. The available cursors and cursor names differ in Windows 7. In either version, the cursor files are found in the C:\WINDOWS\Cursors\ directory. You may need to adjust the variable path\$ if your Windows path differs. This demo loads four cursors found in the Windows\Cursors\ folder. In addition, four custom cursors, can be loaded and used. These public domain cursors were obtained from [RealWorld Graphics](#).

- [BlueTranspCircle.cur](#)

- [Details](#)
- [Download](#)
- 13 KB

- [GreenTranspCircle.cur](#)

- [Details](#)
- [Download](#)
- 13 KB

- [LumaBlue.ani](#)
 - [Details](#)
 - [Download](#)
 - 8 KB

- [LumaGreen.ani](#)
 - [Details](#)
 - [Download](#)
 - 8 KB

Custom cursors can be saved in the same folder as the .bas file. It is not necessary to save them in the WINDOWS>Cursors> folder.

Setting the Cursor in the Control Panel

My Computer> Control Properties> Mouse> Pointers brings up a dialog that will allow you to change the Mouse Pointer scheme. You can search the web to find numerous libraries of cursors in just about every theme imaginable. **This demo DOES NOT alter or add to your computer mouse settings.**

SetCursor vs SetSystemCursor

The code in this demo temporarily changes the cursor while it is over the graphicbox only. Each time the mouse moves, the call to SetCursor must be made. The cursor reverts to the original when not moving over the graphicbox. It is possible to permanently change the look of the cursor with the API call, SetSystemCursor. Be careful, though, as this call can make irreversible and unpleasant changes. **This demo DOES NOT call SetSystemCursor.**

Static Cursors and Animated Cursors

Static cursors end with the extension **.cur**. An animated cursor usually ends with the extension **.ani**. Both are loaded with the user32.dll "LoadCursorFromFile" API call. This demo displays both static and animated cursors. Use the Menu to sample the different icons.

```
'Demo by Janet Terra
'Originally published Liberty BASIC Newsletter #135
'Updated July, 2011, for LBPE
'Requires BlueTranspCircle.cur, GreenTranspCircle.cur, LumaBlue.ani, L
umaGreen.ani
'or substitute with your own cursor files

nomainwin
dim StyleCursor(8)
path$ = "C:\WINDOWS\Cursors\"
'Cursors following path$ found in Windows Directory
' Sample Cursors common to both Windows XP and Windows 7
StyleCursor$(1) = path$;"arrow_rl.cur"
StyleCursor$(2) = path$;"pen_il.cur"
StyleCursor$(3) = path$;"lnodrop.cur"
StyleCursor$(4) = path$;"help_im.cur"

StyleCursor$(5) = "BlueTranspCircle.cur"
StyleCursor$(6) = "GreenTranspCircle.cur"
StyleCursor$(7) = "LumaBlue.ani"
StyleCursor$(8) = "LumaGreen.ani"

WindowWidth = 308
WindowHeight = 368
UpperLeftX = int((DisplayWidth - WindowWidth)/2)
UpperLeftY = int((DisplayHeight - WindowHeight)/2)

msg$ = " Move Mouse Over "
menu #main, "Cursors", "Windows - Black Arrow", _
[cursor1], "Windows - Blue Pen", _
[cursor2], "Windows - No Drop", [cursor3], _
"Windows - Arrow with Help", [cursor4], |, _
"Custom - Blue Circle", [cursor5], "Custom - Green Circle", _
[cursor6], |, "Custom Animated - Luma Blue", [cursor7], _
"Custom Animated - Luma Green", [cursor8], |, "Normal", _
[cursor0], |, "E&xit", [endDemo]
```

```
    graphicbox #main.g, 0, 0, 300, 100
    statictext #main.s, "", 0, 110, 300, 100
    stylebits #main.s, _SS_CENTERIMAGE, 0, 0, 0
    textbox #main.t, 0, 220, 300, 100
    stylebits #main.t, _ES_READONLY or _ES_MULTILINE,
    _ES_AUTOHSCROLL, 0, 0
    open "Load Cursor From File" for window_nf as #main
    #main "trapclose [endDemo]"
    hMain = hwnd(#main)
    hGB = hwnd(#main.g)
    call SetClass hGB

'Text Messages to User
    #main, "font Microsoft_Sans_Serif 12 bold"
    #main.g "font Microsoft_Sans_Serif 12 bold"
    #main.g "down; fill yellow; color darkgreen; bgcolor yellow"
    #main.g "\\\\";msg$;"Graphic Window"
    #main.g "flush"
    #main.s msg$;"Regular Window"
    #main.t chr$(13);chr$(10);chr$(13);chr$(10);msg$;"Textbox"

'Trapping When mouseMove"
    #main.g "when mouseMove [showCursor]"
    #main.g "setfocus"

[cursor0]
    cursorHandle = _OCR_NORMAL
wait

[cursor1]
    cursorFile$ = StyleCursor$(1)
    cursorHandle = LoadCursorFromFile(cursorFile$)
wait

[cursor2]
    cursorFile$ = StyleCursor$(2)
    cursorHandle = LoadCursorFromFile(cursorFile$)
wait

[cursor3]
    cursorFile$ = StyleCursor$(3)
    cursorHandle = LoadCursorFromFile(cursorFile$)
wait

[cursor4]
```

```
        cursorFile$ = StyleCursor$(4)
        cursorHandle = LoadCursorFromFile(cursorFile$)
wait

[cursor5]
        cursorFile$ = StyleCursor$(5)
        cursorHandle = LoadCursorFromFile(cursorFile$)
wait

[cursor6]
        cursorFile$ = StyleCursor$(6)
        cursorHandle = LoadCursorFromFile(cursorFile$)
wait

[cursor7]
        cursorFile$ = StyleCursor$(7)
        cursorHandle = LoadCursorFromFile(cursorFile$)
wait

[cursor8]
        cursorFile$ = StyleCursor$(8)
        cursorHandle = LoadCursorFromFile(cursorFile$)
wait

[showCursor]
        Call SetCursor cursorHandle
wait

[endDemo]
        close #main
end

'Call just once in the beginning of the program
sub SetClass hWin
    index = _GCL_HCURSOR or 0
    value = 0
    calldll #user32, "SetClassLongA", _
        hWin as ulong, _
        index as long, _
        value as long, _
        result as long
end sub

'Set Cursor to Handle of Loaded Cursor
sub SetCursor hCursor
    calldll #user32, "SetCursor", _
```

```
        hCursor as ulong, _
        result as long
    end sub

'Load Cursor from File
function LoadCursorFromFile(file$)
    callDll #user32, "LoadCursorFromFileA", _
    file$ as ptr, _
    LoadCursorFromFile as ulong
end function
```

Finding System Cursors

Not all Windows cursors are common to both Windows XP and Windows 7. This next demo uses [the Files command](#) to list the available cursors. Hover the mouse over the graphicbox to preview that cursor. Note that some .ani cursor files appear to be non-functioning.

```
' Use the Files command to find cursors available
' on the system

    dim info$(10, 10)
    path$ = "C:\WINDOWS\Cursors\"
    files path$, info$()
    nFiles = val(info$(0, 0))

' Fill an array with the available cursors
    dim sysCursors$(nFiles)
    for i = 1 to nFiles
        sysCursors$(i) = info$(i, 0)
    next i

' The Preview
    nomainwin
    WindowWidth = 400
    WindowHeight = 450
    graphicbox #demo.g, 0, 0, 400, 200
    listbox #demo.cur, sysCursors$(), [selCursor], 25, 220, 340, 180
    open "Preview System Cursors" for window as #demo
    #demo "trapclose [endDemo]"
    #demo.g "down; fill cyan; backcolor cyan"
    #demo.g "font verdana 14 bold; place 20 40"
    #demo.g "\\Hover Mouse to Preview Cursor"
```

```
#demo.g "flush"
#demo "font verdana 12 bold"
#demo.cur "singleclickselect"

' Necessary Cursor Information
path$ = "C:\WINDOWS\Cursors\"
hDemo = hWnd(#demo)
hGB = hWnd(#demo.g)
Call SetClass hGB

' Trapping When mouseMove
#demo.g "When mouseMove [showCursor]"
#demo.g "Setfocus"
wait

[selCursor]
#demo.cur "selectionindex? selCur"
cursorHandle = LoadCursorFromFile(path$;sysCursors$(selCur))
wait

[showCursor]
Call SetCursor cursorHandle
wait

[endDemo]
close #demo
end

sub SetClass hWin
    index = _GCL_HCURSOR or 0
    value = 0
    callDll #user32, "SetClassLongA", _
        hWin as ulong, _
        index as long, _
        value as long, _
        result as long
end sub

sub SetCursor hCursor
    callDll #user32, "SetCursor", _
        hCursor as ulong, _
        result as long
end sub

function LoadCursorFromFile(file$)
```

```
    calldll #user32, "LoadCursorFromFileA", _  
    file$ as Ptr, _  
    LoadCursorFromFile as ulong  
end function
```

[Custom Cursors With LoadCursorFromFile](#) | [Changing the Cursor](#) | [Finding Custom Cursors](#) | [Setting the Cursor in the Control Panel](#) | [SetCursor vs SetSystemCursor](#) | [Static Cursors and Animated Cursors](#) | [Finding System Cursors](#)