Designing Your Program with GUI Compatible Colors

The Default GUI Colors

Table of Contents

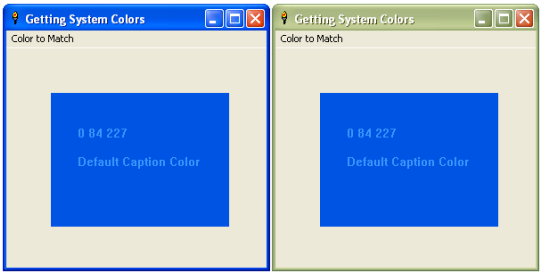
The default GUI colors are a blue shade and a gray shade. The blue shade is the default caption color and the gray shade is the default background color. Liberty BASIC allows the programmer to retrieve the default background color by assigning the color `buttonface`.

```
Print #g "Color Buttonface"
```

There is no native Liberty BASIC color for the caption color.

Clashing With the Desktop Color Theme

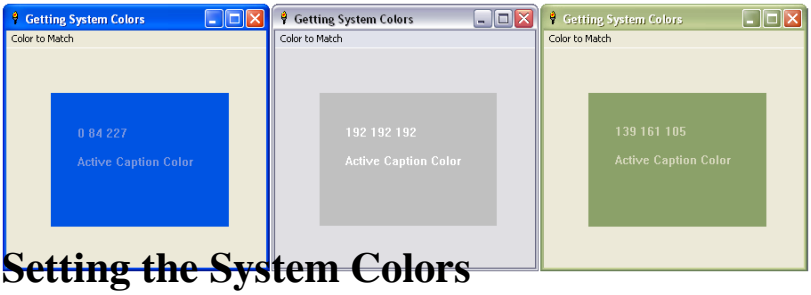
The desktop color themes can easily be customized by the computer's owner. When programs are shared, there is no way of predicting what those desktop colors themes may be. What looks very nice on one theme may look absolutely appalling on another. Compare these two color choices with two different theme colors.



Getting the System Color

Matching and coordinating the program colors to the desktop theme colors will give a more professional look to your application. Use the API call `GetSysColor` to identify these colors.

```
Function GetSysColor(nIndex)
    CallDLL #user32, "GetSysColor", _
        nIndex as Long, _
        GetSysColor as Ulong
End Function
```



Setting the System Colors

The API call `SetSysColor` can be used to change the system colors of the host machine. ***This is NOT advisable!*** `SetSysColor` will be applied to all open windows, not just your application window. At best, your user will become annoyed at having the computer settings changed. More likely, your user will suspect your program contains some type of virus or malware lurking in the background and refuse to further test your program. Better to conform your program to the user's preference than to conform your user's presence to your program.

Demo

Here is a short demo that retrieves the colors of the

1. Active Caption (Title) Bar

2. Inactive Caption (Title) Bar
3. Buttonface
4. Highlighted Item

and then uses those colors in the graphical text display.

```
    WindowWidth = 300
    WindowHeight = 300

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

    Menu #main, "Color to Match", "&1 - Active Caption", _
        Option1, "&2 - Inactive Caption", Option2, _
        "&3 - Buttonface", Option3, "&4 - Highlighted Item", _
        Option4,|, "E&xit", XbyMenu

    Stylebits #main.g, 0, _WS_BORDER, 0, 0
    Graphicbox #main.g, 50, 50, 200, 150
    Open "Getting System Colors" for Window as #main
    #main "Trapclose XbyTrap"

    #main.g "Down; Cls; Fill Black; Flush"

    RGB = GetSysColor(_COLOR_ACTIVECAPTION)
    RGB$(1) = LongPixelToRGB$(RGB)
    RGB = GetSysColor(_COLOR_INACTIVECAPTION)
    RGB$(2) = LongPixelToRGB$(RGB)
    RGB = GetSysColor(_COLOR_BTNFACE)
    RGB$(3) = LongPixelToRGB$(RGB)
    RGB = GetSysColor(_COLOR_HIGHLIGHT)
    RGB$(4) = LongPixelToRGB$(RGB)
Wait

Sub XbyTrap handle$
    Close #main
    End
End Sub

Sub XbyMenu
    Call XbyTrap "#main"
End Sub

Sub Option1
```

```
    #main.g "Cls; Fill ";RGB$(1)
    #main.g "Backcolor ";RGB$(1)
    #main.g "Color ";RGB$(2)
    #main.g "Place 30 50"
    #main.g "\";RGB$(1)
    #main.g "\\Active Caption Color"
    #main.g "Flush"
End Sub

Sub Option2
    #main.g "Cls; Fill ";RGB$(2)
    #main.g "Backcolor ";RGB$(2)
    #main.g "Color ";RGB$(1)
    #main.g "Place 30 50"
    #main.g "\";RGB$(2)
    #main.g "\\Inactive Caption Color"
    #main.g "Flush"
End Sub

Sub Option3
    #main.g "Cls; Fill ";RGB$(3)
    #main.g "Backcolor ";RGB$(3)
    #main.g "Color ";RGB$(4)
    #main.g "Place 30 50"
    #main.g "\";RGB$(3)
    #main.g "\\Buttonface"
    #main.g "Flush"
End Sub

Sub Option4
    #main.g "Cls; Fill ";RGB$(4)
    #main.g "Backcolor ";RGB$(4)
    #main.g "Color ";RGB$(3)
    #main.g "Place 30 50"
    #main.g "\";RGB$(4)
    #main.g "\\Highlighted Item Color"
    #main.g "Flush"
End Sub

Function GetSysColor(nIndex)
    CallDLL #user32, "GetSysColor", _
        nIndex as Long, _
        GetSysColor as Ulong
End Function

Function LongPixelToRGB$(LongPixel)
```

```
    b = Int(LongPixel / 256 ^ 2)
    LongPixel = LongPixel - b * 256 ^ 2
    g = Int(LongPixel / 256)
    r = LongPixel - g * 256
    LongPixelToRGB$ = Str$(r);" ";Str$(g);" ";Str$(b)
End Function
```

`nIndex` identifies the desired control to be queried. The color is returned in long pixel format. This long pixel integer must be converted to the Red, Green and Blue (RGB) values before that color can be assigned. This conversion takes place in the function `LongPixelToRGB$(LongPixel)`.

More Windows Constants

`GetSysColor` will retrieve more control colors than these four shown. For a full listing and description of available `Windows Constants`, visit MSDN Library: GetSysColor

What About BMP Buttons?

`Bmpbuttons` loaded from file can also be theme color customized. To find more information about changing the background of a `bmpbutton`, read Color-Matched BMPButtons at Alyce's Restaurant.

Return to LBPE Home Page
Return to GUI Programming
Return to Windows API
