

GDI and the Device Context

-

[Alyce](#)

[GDI](#) | [GetDC](#) | [GetWindowDC](#) | [ReleaseDC](#) | [Long Color Values](#) | [SetPixelV](#) | [Demo](#)

Some text below is copied from the Microsoft Developers Network Library.

For an eBook or printed book on using the API with Liberty BASIC, see:

[APIs for Liberty BASIC](#)

GDI

GDI stands for Graphics Device Interface. This is the way that Windows interfaces with the graphics hardware, such as the monitor and printer.

GetDC

GDI communicates with the hardware device using the handle to a Device Context (DC). To get a DC handle, pass the graphicbox handle to the API function from User32.DLL called GetDC. Note that handles are always of type "ulong".

The GetDC function retrieves a handle to a device context (DC) for the client area of a specified window or for the entire screen. You can use the returned handle in subsequent GDI functions to draw in the DC. The device context is an opaque data structure, whose values are used internally by GDI.

```
h = hwnd(#1.graphicbox)
calldll #user32, "GetDC", _
h as ulong, _      'handle of window or graphicbox client area
hdc as ulong      'returns handle of Device Context - 0=failure
```

GetWindowDC

The GetWindowDC function retrieves the device context (DC) for the entire window, including title bar, menus, and scroll bars. A window device context permits painting anywhere in a window, because the origin of the device context is the upper-left corner of the window instead of the client area.

GetWindowDC assigns default attributes to the window device context each time it retrieves the device context. Previous attributes are lost.

```
hMain = hwnd(#main)
callDll #user32, "GetWindowDC", _
h as ulong, _          'handle of window
hWinDC as ulong
'returns handle of DC for entire window - 0=failure
```

The first parameter is the handle to the window with a device context that is to be retrieved. If this value is NULL, GetWindowDC retrieves the device context for the entire screen.

ReleaseDC

At the close of the program, or when the Device Context is no longer needed, call ReleaseDC. The application must call the ReleaseDC function for each call to the GetWindowDC function and for each call to the GetDC function that retrieves a common DC.

```
callDll #user32, "ReleaseDC", _
h as ulong, _          'window handle
hdc as ulong, _        'device context
ret as long            'nonzero=success
```

Long Color Values

GDI API functions specify colors as a long integer value, made up of the red, green and blue components, each in the range of 0 - 255. 0 is no amount of the specified color, and 255 is complete saturation of the specified color. True red is red=255, green=0, blue=0, for instance.

Here is a Liberty BASIC function that creates a long color value from the red, green and blue components.

```
'red, green and blue values MUST be in the range 0-255
function MakeRGB(red,green,blue)
    MakeRGB=(blue*256*256)+(green*256)+red
end function
```

SetPixelV

The SetPixelV function sets the pixel at the specified coordinates to the closest approximation of the specified color. SetPixelV is faster than the older SetPixel function.

```
callDll #gdi32, "SetPixelV", _  
hdc as ulong, _ 'device context  
x as long, _ 'x location of pixel  
y as long, _ 'y location of pixel  
color as long, _ 'long color value  
ret as long 'nonzero=success
```

Demo

```
nomainwin  
winWide=700:winHigh=500  
WindowWidth=winWide+50:WindowHeight=winHigh+50  
UpperLeftX=1:UpperLeftY=1  
  
graphicbox #1.g, 0,0,winWide,winHigh  
open "GDI Demo" for window as #1  
    #1 "trapclose [quit]"  
    #1.g "down"  
  
    h=hwnd(#1.g) 'graphicbox handle  
  
    'get device context for window:  
    callDll #user32, "GetDC", _  
    h as ulong, _ 'graphicbox handle  
    hdc as ulong 'returns handle to device context  
  
    color = MakeRGB(255,190,0) 'create a long int color value  
  
    'draw lines of pixels in a loop  
    for x = 1 to 500  
        for y = 40 to 140 step 10  
            callDll #gdi32, "SetPixelV", _  
            hdc as ulong, _ 'device context  
            x as long, _ 'x location of pixel  
            y as long, _ 'y location of pixel  
            color as long, _ 'long color value  
            ret as long 'nonzero=success  
        next  
    next  
  
wait  
  
[quit]
```

```
callDll #user32, "ReleaseDC", _  
h as ulong, _      'window handle  
hdc as ulong, _    'device context  
ret as long
```

```
close #1:end
```

```
function MakeRGB(red,green,blue)  
    MakeRGB=(blue*256*256)+(green*256)+red  
end function
```

[GDI Tutorials Home](#)