

FLOOD FILL

[Alyce](#)

Adapted from:

[APIs for Liberty BASIC](#)

[FLOOD FILL](#) | [GDI](#) | [GetDC](#) | [ExtFloodFill](#) | [Long Color Value](#)

GDI

GDI is an abbreviation for Graphics Device Interface. Windows uses GDI32.DLL to accomplish many graphics tasks.

GetDC

A DC is a Device Context. Windows sends graphics commands to a Device Context, so to use API graphics commands from the GDI32.DLL we must first retrieve a device context handle. To get a DC handle, we pass the graphicbox handle to the API function from User32.DLL called GetDC.

```
h = hwnd(#1.graphicbox)
call dll #user32, "GetDC",_
    h as ulong,_
    hdc as ulong
```

ExtFloodFill

Graphic objects can be filled with the ExtFloodFill (extended flood fill) function. It is the same as an older API function called FloodFill, with extra functionality. It looks like this.

```
' wFillType style flags
' _FLOODFILLBORDER (=0)
' _FLOODFILLSURFACE (= 1)
call dll #gdi32, "ExtFloodFill",_
    hdc As ulong,        'device context
    x As Long,           'x location to start filling
    y As Long,           'y location to start filling
```

```
crColor As Long,_ 'long color value of border, or color to replace
wFillType As Long_ 'flag for type of fill
result As Long      'nonzero if successful
```

The first argument for ExtFloodFill is the handle for the graphicbox device contest.

The x and y location to begin the fill are the next arguments. These coordinates are counted from the upper left corner of the graphics area.

The arguments for color and fill type are next. There are two ways to fill an area. With _FLOODFILLBORDER type of fill, the fill area is bounded by the color specified by the crColor parameter. With _FLOODFILLSURFACE type of fill, the fill area is defined by the color that is specified by crColor. Filling continues outward in all directions as long as the color is encountered. This style is useful for filling areas with multicolored boundaries.

Long Color Value

The crColor argument requires an RGB color in one long numeric value. We can create a long color value from the RGB components like this:

```
crColor = MakeRGB(100,200,30)
print crColor

wait

function MakeRGB(red,green,blue)
  if red<0 then red=0
  if red>255 then red=255
  if green<0 then green=0
  if green>255 then green=255
  if blue<0 then blue=0
  if blue>255 then blue=255
  MakeRGB=(blue*256*256)+(green*256)+red
end function
```