

Other uses for bitmap brushes

 devblogs.microsoft.com/oldnewthing/20031009-00

October 9, 2003



Raymond Chen

Bitmap brushes used to be these little 8×8 monochrome patterns that you could use for hatching and maybe little houndstooth patterns if you were really crazy. But you can do better.

CreatePatternBrush lets you pass in any old bitmap – even a huge one, and it will create a brush from it. The bitmap will automatically be tiled, so this is a quick way to get bitmap tiling. Let GDI do all the math for you!

This is particularly handy when you're stuck with a mechanism where you are forced to pass an `HBRUSH` but you really want to pass an `HBITMAP`. Convert the bitmap to a brush and return that brush instead.

For example, let's take our scratch program and give it a custom tiled background by using a pattern brush.

```

HBRUSH CreatePatternBrushFromFile(LPCTSTR pszFile)
{
    HBRUSH hbr = NULL;
    HBITMAP hbm = (HBITMAP)LoadImage(g_hinst, pszFile,
        IMAGE_BITMAP, 0, 0, LR_LOADFROMFILE);
    if (hbm) {
        hbr = CreatePatternBrush(hbm);
        DeleteObject(hbm);
    }
    return hbr;
}
BOOL
InitApp(LPSTR lpCmdLine)
{
    BOOL fSuccess = FALSE;
    HBRUSH hbr = CreatePatternBrushFromFile(lpCmdLine);
    if (hbr) {
        WNDCLASS wc;
        wc.style = 0;
        wc.lpfnWndProc = WndProc;
        wc.cbClsExtra = 0;
        wc.cbWndExtra = 0;
        wc.hInstance = g_hinst;
        wc.hIcon = NULL;
        wc.hCursor = LoadCursor(NULL, IDC_ARROW);
        wc.hbrBackground = hbr;
        wc.lpszMenuName = NULL;
        wc.lpszClassName = CLASSNAME;
        fSuccess = RegisterClass(&wc);
        // Do not delete the brush - the class owns it now
    }
    return fSuccess;
}

```

With a corresponding adjustment to `WinMain` :

```
if (!InitApp(lpCmdLine)) return 0;
```

Pass the path to a *.bmp file on the command line, and bingo, the window will tile its background with that bitmap. Notice that we did not have to change anything other than the class registration. No muss, no fuss, no bother.

Here's another way you can use bitmap brushes to save yourself a lot of work. Start with a new scratch program and change it as follows:

```

HBRUSH g_hbr; // the pattern brush we created
HBRUSH CreatePatternBrushFromFile(LPCTSTR pszFile)
{ ... same as above ... }
void
PaintContent(HWND hwnd, PAINTSTRUCT *pps)
{
    BeginPath(pps->hdc);
    Ellipse(pps->hdc, 0, 0, 200, 100);
    EndPath(pps->hdc);
    HBRUSH hbrOld = SelectBrush(pps->hdc, g_hbr);
    FillPath(pps->hdc);
    SelectBrush(pps->hdc, hbrOld);
}

```

And add the following code to WinMain before the call to `CreateWindowEx` :

```

g_hbr = CreatePatternBrushFromFile(lpCmdLine);
if (!g_hbr) return 0;

```

This time, since we are managing the brush ourselves we need to remember to

```

DeleteObject(g_hbr);

```

at the end of WinMain before it returns.

This program draws an ellipse filled with your bitmap. The `FillPath` function uses the currently-selected brush, so we select our bitmap brush (instead of a boring solid brush) and draw with that. Result: A pattern-filled ellipse. Without a bitmap brush, you would have had to do a lot of work manually clipping the bitmap (and tiling it) to the ellipse.

Raymond Chen

Follow

