

# Creating tree view check boxes manually: Themed check boxes

---

 [devblogs.microsoft.com/oldnewthing/20171201-00](https://devblogs.microsoft.com/oldnewthing/20171201-00)

December 1, 2017



Raymond Chen

Originally, we created our check box state image list with `DrawFrameControl`. But that gives you the old and busted check boxes. What if you want the new hotness of visual styles?

Enter `DrawThemeBackground`.

```

#include <uxtheme.h>
#include <vsstyle.h>

HIMAGELIST CreateTreeViewCheckBoxes(HWND hwnd, int cx, int cy)
{
    const int frames = 6;

    // Get a DC for our window.
    HDC hdcScreen = GetDC(hwnd);

    // Get a button theme for the window, if available.
    HTHEME htheme = OpenThemeData(hwnd, L"button");

    // If there is a theme, then ask it for the size
    // of a checkbox and use that size.
    if (htheme) {
        SIZE siz;
        GetThemePartSize(htheme, hdcScreen,
            BP_CHECKBOX, CBS_UNCHECKEDNORMAL, nullptr,
            TS_DRAW, &siz);
        cx = siz.cx;
        cy = siz.cy;
    }

    // Create a 32bpp bitmap that holds the desired number of frames.
    BITMAPINFO bi = { sizeof(BITMAPINFOHEADER), cx * frames, cy, 1, 32 };
    void* p;
    HBITMAP hbmCheckboxes = CreateDIBSection(hdcScreen, &bi,
        DIB_RGB_COLORS, &p, nullptr, 0);

    // Create a compatible memory DC.
    HDC hdcMem = CreateCompatibleDC(hdcScreen);

    // Select our bitmap into it so we can draw to it.
    HBITMAP hbmOld = SelectBitmap(hdcMem, hbmCheckboxes);

    // Set up the rectangle into which we do our drawing.
    RECT rc = { 0, 0, cx, cy };

    // Frame 0 is not used. Draw nothing.
    OffsetRect(&rc, cx, 0);

    if (htheme) {
        // Frame 1: Unchecked.
        DrawThemeBackground(htheme, hdcMem, BP_CHECKBOX,
            CBS_UNCHECKEDNORMAL, &rc, nullptr);
        OffsetRect(&rc, cx, 0);

        // Frame 2: Checked.
        DrawThemeBackground(htheme, hdcMem, BP_CHECKBOX,
            CBS_CHECKEDNORMAL, &rc, nullptr);
        OffsetRect(&rc, cx, 0);
    }
}

```

```

// Frame 3: Indeterminate.
DrawThemeBackground(htheme, hdcMem, BP_CHECKBOX,
    CBS_MIXEDNORMAL, &rc, nullptr);
OffsetRect(&rc, cx, 0);

// Frame 4: Disabled, unchecked.
DrawThemeBackground(htheme, hdcMem, BP_CHECKBOX,
    CBS_UNCHECKEDDISABLED, &rc, nullptr);
OffsetRect(&rc, cx, 0);

// Frame 5: Disabled, checked.
DrawThemeBackground(htheme, hdcMem, BP_CHECKBOX,
    CBS_CHECKEDDISABLED, &rc, nullptr);

// Done with the theme.
CloseThemeData(htheme);
} else {
// Flags common to all of our DrawFrameControl calls:
// Draw a flat checkbox.
UINT baseFlags = DFCS_FLAT | DFCS_BUTTONCHECK;

// Frame 1: Unchecked.
DrawFrameControl(hdcMem, &rc, DFC_BUTTON,
    baseFlags);
OffsetRect(&rc, cx, 0);

// Frame 2: Checked.
DrawFrameControl(hdcMem, &rc, DFC_BUTTON,
    baseFlags | DFCS_CHECKED);
OffsetRect(&rc, cx, 0);

// Frame 3: Indeterminate.
DrawFrameControl(hdcMem, &rc, DFC_BUTTON,
    baseFlags | DFCS_CHECKED | DFCS_BUTTON3STATE);
OffsetRect(&rc, cx, 0);

// Frame 4: Disabled, unchecked.
DrawFrameControl(hdcMem, &rc, DFC_BUTTON,
    baseFlags | DFCS_INACTIVE);
OffsetRect(&rc, cx, 0);

// Frame 5: Disabled, checked.
DrawFrameControl(hdcMem, &rc, DFC_BUTTON,
    baseFlags | DFCS_INACTIVE | DFCS_CHECKED);
}

// The bitmap is ready. Clean up.
SelectBitmap(hdcMem, hbmOld);
DeleteDC(hdcMem);
ReleaseDC(hwnd, hdcScreen);

```

```

// Create an imagelist from this bitmap.
HIMAGELIST himl = ImageList_Create(cx, cy, ILC_COLOR, frames, frames);
ImageList_Add(himl, hbmCheckboxes, nullptr);

// Don't need the bitmap any more.
DeleteObject(hbmCheckboxes);

return himl;
}

void OnThemeChange(HWND hwnd)
{
    // Rebuild the state images to match the new theme.
    HIMAGELIST himl = CreateTreeViewCheckBoxes(g_hwndChild,
        16, 16);
    ImageList_Destroy(TreeView_SetImageList(g_hwndChild,
        himl, TVSIL_STATE));
}

// WndProc
case WM_THEMECHANGE:
    OnThemeChange(hwnd);
    break;

```

We spruce up the `CreateTreeViewCheckBoxes` function by having it check whether visual styles are enabled for the window. If so, it uses the button theme to draw the check boxes in the various states. If not, then it falls back to our existing `DrawFrameControl` version.

We also respond to the `WM_THEMECHANGE` message by creating a new state image list which match the new theme. We then exchange that state image list into place and destroy the previous (old and busted) state image list.

That's it. The rest is the same as before.

Next time, we'll engage in some historical speculation to help explain why the built-in tree view check boxes are so quirky.

Raymond Chen

**Follow**

