

The format of icon resources, revisited



Raymond Chen

Some time ago, I explained that a logical icon is represented in resources as a group icon directory and a series of individual icon directory entries, each of which in turn refers to another icon image resource.

But I forgot to discuss the format of those second-level resources.

If the icon image resource represents a cursor, then the resource data begins with two 16-bit signed integers representing the x- and y-coordinates of the cursor hotspot.

Next, after the optional hotspot information, comes one of various image formats.

One possibility is the legacy **BITMAPCOREHEADER**, followed by the color table (possibly empty), and then the bitmap pixels.

Another possibility is a **BITMAPINFOHEADER**, again followed by the possibly-empty color table, and then the bitmap pixels.

And a third possibility (starting in Windows Vista) is a PNG-compressed image.

Here's a diagram (not to scale):

Legacy format	Common format	PNG format
<code>int16_t xHotspot;</code> <code>int16_t yHotspot;</code> (Present only for cursors.)		
BITMAPCOREHEADER color table (possibly empty) pixels	BITMAPINFOHEADER color table (possibly empty) pixels	PNG image

You can access this raw icon image resource data by finding, loading, and locking the icon image resource ID with a resource of type **RT_ICON**. You can then pass a pointer to the raw icon image resource data to **CreateIconFromResource** or **CreateIconFromResourceEx** to

convert it to a proper **HICON**.

Alternatively, you can synthesize one of these formats in memory and pass it to **CreateIconFromResource(Ex)**.

Note that in practice, you will want to use **CreateIconFromResourceEx** because **CreateIconFromResource** creates icons at the system default icon size, rather than respecting the size reported by the the icon image data.

The fact that the icon image source data can take the form of a PNG image gives you a sneaky way to load a PNG image as an icon: Load the PNG image into memory and pass it directly to **CreateIconFromResourceEx!**

```
wil::unique_hicon LoadPngAsIcon(PCWSTR fileName)
{
    auto file = wil::unique_hfile{
        CreateFileW(fileName, GENERIC_READ, 0, nullptr,
            OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, nullptr) };
    THROW_LAST_ERROR_IF(!file);

    LARGE_INTEGER size;
    THROW_IF_WIN32_BOOL_FALSE(GetFileSizeEx(file.get(), &size));
    THROW_HR_IF(HRESULT_FROM_WIN32(ERROR_FILE_TOO_LARGE),
        size.HighPart != 0);
    auto bits = std::make_unique<BYTE[]>(size.LowPart);

    DWORD actual;
    THROW_IF_WIN32_BOOL_FALSE(ReadFile(file.get(), bits.get(),
        size.LowPart, &actual, nullptr));
    auto icon = wil::unique_hicon{
        CreateIconFromResourceEx(bits.get(), actual, true,
            0x00030000, 0, 0, 0) };
    THROW_LAST_ERROR_IF(!icon);

    return icon;
}
```