

How does Explorer find the “first” icon in a file

 devblogs.microsoft.com/oldnewthing/20250210-00

February 10, 2025



An old MSDN article from 1995 talks about how Explorer chooses the icon to use for an executable.

Windows NT simply chooses the first resource listed in the application’s RC script. On the other hand, Windows 95’s algorithm is to choose the alphabetically first named group icon if one exists. If one such group resource does not exist, Windows chooses the icon with the numerically lowest identifier.

The first thing I want to do is make a correction. The description of the Windows NT algorithm is incorrect. Both Windows NT and Windows 95 use the same algorithm:

- Choose the alphabetically first named group icon, if available.
- Else, choose the group icon with the numerically lowest identifier.

The reason for this is that the Portable Executable file format requires resources to be listed in the resource dictionary with named resources first (sorted alphabetically), followed by numbered resources (sorted numerically). Explorer calls `EnumResourceNames` and grabs the first one, and since the icon groups are enumerated in the order they appear in the resource dictionary, the effect is the one described above.

If you want to replicate this algorithm, it’s not hard.

```

HICON FindFirstIcon(HMODULE hmod)
{
    HICON icon = nullptr;
    EnumResourceNamesW(hmod, RT_GROUP_ICON,
        [](HMODULE, PCWSTR, PCWSTR id, LONG_PTR param)
        {
            *reinterpret_cast<HICON*>(param) =
                LoadIconW(hmod, id);
            return FALSE;
        }, reinterpret_cast<LONG_PTR>(&icon));
    return icon;
}

```

The callback function tries to load the first icon it finds, and then returns **FALSE** to stop the enumeration. This returns control to **FindFirstIcon**, which returns the icon that the callback function loaded (or at least tried to load).