

How do I determine whether Explorer is showing or hiding file extensions?



Say you want to determine in your program whether Explorer is showing or hiding file extensions.

You can do this by calling `SHGetSettings`:

```
bool AreExtensionsShown()
{
    SHELLFLAGSTATE sfs;
    SHGetSettings(&sfs, SSF_SHOWEXTENSIONS);
    return sfs.fShowExtensions;
}
```

But this is answering the question without understanding the problem.

Even if Explorer is showing extensions, that doesn't mean that *all* extensions are shown. Some extensions are *never* shown, like `.lnk`. Some extensions are *always* shown, like `.dll`. And some rules are contextual: For example, if a file name consists only of the extension, then the extension is always shown.

So maybe what we have is an XY problem. "I have the name of a file and I want to display it in the same way that Explorer does. In order to do that, I need to know whether Explorer is showing extensions, so I will ask how to query that setting."

But querying the setting doesn't tell you everything you need to know in order to format a file name for display in the same way that Explorer does. To take all the display rules into account, you can ask `SHGetFileInfo` to produce the name that Explorer would show if it encountered a file with a specific name.

```

SHFILEINFO sfi;
if (SHGetFileInfo(L"something.txt", FILE_ATTRIBUTE_NORMAL,
    &sfi, sizeof(sfi),
    SHGFI_USEFILEATTRIBUTES | SHGFI_DISPLAYNAME)) {
    // use the name in sfi.szDisplayName
}

```

Note that the file attributes should be `FILE_ATTRIBUTE_NORMAL` if you want to display the name for a hypothetical file named `something.txt`, or `FILE_ATTRIBUTE_DIRECTORY` if you want to display the name for a hypothetical directory named `something.txt`.

But wait, this still might be an XY problem. “I got the name of a file from the system, and I want to display it the same way that Explorer does. So I’ll ask how to obtain the name that Explorer would show when faced with a file with a specific name.” But that’s still not enough, because the name that Explorer displays for a file depends on more than just the file’s name.

For example, the file named `C:\ProgramData\Microsoft\Windows\Start Menu\Programs\System Tools\Task Manager.lnk` displays as *Aktivitetshanteraren* on Swedish systems. To get this name, you need to provide a full path to `SHGetFileInfo` and say that it should consult the actual file (by removing the `SHGFI_USEFILEATTRIBUTES` flag).

```

SHFILEINFO sfi;
if (SHGetFileInfo(L"C:\\ProgramData\\Microsoft\\"
    L"Windows\\Start Menu\\Programs\\"
    L"System Tools\\Task Manager.lnk",
    0,
    &sfi, sizeof(sfi),
    SHGFI_DISPLAYNAME)) {
    // use the name in sfi.szDisplayName
}

```

If you received the file in the form of a pidl or `IShellItem`, you can take advantage of the information already cached in the pidl / `IShellItem` and obtain the display name without further disk access.

```

// If you have an IShellFolder + child pidl
IShellFolder* psf = [ ... ];
PIDLIST_CHILD pidl = [ ... ];
STRRET str;

if (SUCCEEDED(psf->GetDisplayNameOf(pidl,
    SHGDN_NORMAL, &str))) {
    wil::unique_cotaskmem_string name;
    if (SUCCEEDED(StrRetToStrW(&str, pidl, name.put()))) {
        // use the name in "name"
    }
}

// If you have an absolute pidl
PIDLIST_ABSOLUTE pidlAbsolute = [ ... ];
SHFILEINFO sfi;
if (SHGetFileInfo((PCWSTR)pidlAbsolute,
    0,
    &sfi, sizeof(sfi),
    SHGFI_PIDL | SHGFI_DISPLAYNAME)) {
    // use the name in sfi.szDisplayName
}

// If you have an IShellItem
IShellItem* psi = [ ... ];
wil::unique_cotaskmem_string name;
if (SUCCEEDED(psi->GetDisplayName(SIGDN_NORMALDISPLAY,
    name.put()))) {
    // use the name in "name"
}

```

If your real question is “How can I display this thing the same way that Explorer does?” then ask that question. Don’t ask, “How can I determine whether Explorer is showing or hiding file extensions?” and try to replicate Explorer’s name-generation algorithm from that one Boolean value.