

Getting the display name for a shell property

 devblogs.microsoft.com/oldnewthing/20130429-00

April 29, 2013



Raymond Chen

Today's Little Program takes the symbolic name for a shell property and returns a string suitable for display to the end-user, translated into the user's specified display language.

```
#include <windows.h>
#include <ole2.h>
#include <propsys.h>
#include <propkey.h>
#include <atlbase.h>
#include <atlalloc.h>

int __cdecl wmain(int argc, PWSTR argv[])
{
    CCoInitialize init;
    if (SUCCEEDED(init) && argc == 2) {
        CComPtr<IPropertyDescription> spdesc;
        if (SUCCEEDED(PSGetPropertyDescriptionByName(
            argv[1], IID_PPV_ARGS(&spdesc)))) {
            CComHeapPtr<wchar_t> spszName;
            if (SUCCEEDED(spdesc->GetDisplayName(&spszName))) {
                wprintf(L"%ls\n", static_cast<PWSTR>(spszName));
            }
        }
    }
    return 0;
}
```

Run this program with the string `System.Music.AlbumArtist` on the command line, and the result is the message `Album artist` on English-language systems.

The actual workings of the program is pretty straightward. We ask the property system for an interface that describes the property name, and ask that interface to give us the display name, which we print out.

Nothing fancy here. The trick is just knowing that the function exists in the first place.

[Raymond Chen](#)

Follow

