

How do I get the user-customized name of a mapped network drive?

devblogs.microsoft.com/oldnewthing/20151019-00

October 19, 2015



Raymond Chen

Last time, we displayed the names of This PC and Recycle Bin. This time, we'll look at mapped volumes, because they are a little tricky.

When you map a network drive, the name in Explorer defaults to something like *sharename* (`\\server`) (Z:). But you can right-click the label, select Rename, and change it to *Awesome* if you like.

Let's try to retrieve the name *Awesome*. Take the program from last time and make these changes:

```
int __cdecl wmain(int argc, wchar_t **argv)
{
    CoInitialize(0);
    IShellItem* item;
    SHCreateItemFromParsingName(argv[1], nullptr,
                               IID_PPV_ARGS(&item));
    PrintDisplayName(item, SIGDN_NORMALDISPLAY, L"name");
    item->Release();
    CoUninitialize();
    return 0;
}
```

This prints the display name of whatever you pass on the command line. Let's say that drive Z: is mapped to `\\server\sharename`.

Run the program with the command line parameter `Z:`,

```
name = sharename (\\server) (Z:)
```

Now go to Explorer and rename the drive to *Awesome*. Then run the program again with `Z:` on the command line.

```
name = Awesome (Z:)
```

Close. We got the *Awesome* part, but the non-awesome drive letter is still there. That sort of makes sense, since Explorer also shows the non-awesome drive letter.

But what if you really want it without the drive letter? Well, you can ask for a different kind of display name.

```
int __cdecl wmain(int argc, wchar_t **argv)
{
    CoInitialize(0);
    IShellItem* item;
    SHCreateItemFromParsingName(argv[1], nullptr,
                               IID_PPV_ARGS(&item));
    PrintDisplayName(item, SIGDN_PARENTRELATIVEEDITING, L"name");
    item->Release();
    CoUninitialize();
    return 0;
}
```

This time, we ask for the parent-relative editing name. This is the name used by the Rename command when you rename an item that is displayed relative to its parent.

Run the program with `Z:` on the command line, and see what happens:

```
name = Awesome
```

Awesome.

For those who want to do things the classic way, you can use the `SHGDN_INFOLDER | SHGDN_FOREDITING` flags.

Take the second program (the one that uses the classic style) and make these changes:

```
int __cdecl wmain(int argc, wchar_t **argv)
{
    CoInitialize(0);
    PIDLIST_ABSOLUTE absolute;
    SHParseDisplayName(argv[1], nullptr, &absolute, 0, nullptr);
    PrintDisplayName(absolute, SHGDN_INFOLDER | SHGDN_FOREDITING, L"name");
    CoTaskMemFree(absolute);
    CoUninitialize();
    return 0;
}
```

[Raymond Chen](#)

Follow

