

# Even if you open a file with GUID, you can still get its name, or at least one of its names

---

 [devblogs.microsoft.com/oldnewthing/20190410-00](https://devblogs.microsoft.com/oldnewthing/20190410-00)

April 10, 2019



Raymond Chen

Some time ago, [I showed how you could eschew file names entirely and use GUIDs to open your files.](#) But even if you choose to open the file with a GUID, you can still get its name.

Take the second program from that earlier article and make these changes:

```

#define UNICODE
#define _UNICODE
#include <windows.h>
#include <stdio.h>
#include <tchar.h>
#include <ole2.h>

int __cdecl _tmain(int argc, PTSTR *argv)
{
    HANDLE hRoot = CreateFile(_T("C:\\"), 0,
        FILE_SHARE_READ | FILE_SHARE_WRITE |
        FILE_SHARE_DELETE, NULL,
        OPEN_EXISTING,
        FILE_FLAG_BACKUP_SEMANTICS, NULL);
    if (hRoot != INVALID_HANDLE_VALUE) {
        FILE_ID_DESCRIPTOR desc;
        desc.dwSize = sizeof(desc);
        desc.Type = ObjectIdType;
        if (SUCCEEDED(CLSIDFromString(argv[1], &desc.ObjectId))) {
            HANDLE h = OpenFileById(hRoot, &desc, GENERIC_READ,
                FILE_SHARE_READ | FILE_SHARE_WRITE |
                FILE_SHARE_DELETE, NULL, 0);
            if (h != INVALID_HANDLE_VALUE) {
                TCHAR buffer[MAX_PATH];
                DWORD result = GetFinalPathNameByHandle(h,
                    buffer, ARRAYSIZE(buffer));
                if (result > 0 && result < ARRAYSIZE(buffer)) {
                    _tprintf(_T("Final path is %s\n"), buffer);
                }
                CloseHandle(h);
            }
        }
        CloseHandle(hRoot);
    }
    return 0;
}

```

There's a catch: If the file has multiple names (say, due to hard links), then only one of the names is returned, and you don't get to pick which one. The system will pick one arbitrarily.

You can get the other names with the `FindFirstFileName` function, which I discussed [some time ago](#).

[Raymond Chen](#)

**Follow**

