

# How do I find the original name of a hard link?

---

 [devblogs.microsoft.com/oldnewthing/20110720-00](http://devblogs.microsoft.com/oldnewthing/20110720-00)

July 20, 2011



Raymond Chen

A customer asked, “Given a hardlink name, is it possible to get the original file name used to create it in the first place?”

Recall that hard links create an alternate name for a file. Once that alternate name is created, there is no way to tell which is the original name and which is the new name. The new file does not have a “link back to the original”; they are both links to the underlying file content. This is an old topic, so I won’t go into further detail. Though this question does illustrate that many people continue to misunderstand what hard links are.

Anyway, once you figure out what the customer is actually asking, you can give a meaningful answer: “Given the path to a file, how can I get all the names by which the file can be accessed?” The answer is `FindFirstFileNameW`.

Note that the names returned by the `FindFirstFileNameW` family of functions are relative to the volume mount point. To convert it to a full path, you need to append it to the mount point. Something like this:

```

typedef void (*ENUMERATEDNAMEPROC)(__in PCWSTR);
void ProcessOneName(
    __in PCWSTR pszVolumeRoot,
    __in PCWSTR pszLink,
    __in ENUMERATEDNAMEPROC pfnCallback)
{
    wchar_t szFile[MAX_PATH];
    if (SUCCEEDED(StringCchCopy(szFile, ARRAYSIZE(szFile), pszVolumeRoot)) &&
        PathAppend(szFile, pszLink)) {
        pfnCallback(szFile);
    }
}
void EnumerateAllNames(
    __in PCWSTR pszFileName,
    __in ENUMERATEDNAMEPROC pfnCallback)
{
    // Supporting paths longer than MAX_PATH left as an exercise
    wchar_t szVolumeRoot[MAX_PATH];
    if (GetVolumePathName(pszFileName, szVolumeRoot, ARRAYSIZE(szVolumeRoot))) {
        wchar_t szLink[MAX_PATH];
        DWORD cchLink = ARRAYSIZE(szLink);
        HANDLE hFind = FindFirstFileNameW(pszFileName, 0, &cchLink, szLink);
        if (hFind != INVALID_HANDLE_VALUE) {
            ProcessOneName(szVolumeRoot, szLink, pfnCallback);
            while (cchLink = ARRAYSIZE(szLink),
                FindNextFileNameW(hFind, &cchLink, szLink)) {
                ProcessOneName(szVolumeRoot, szLink, pfnCallback);
            }
            FindClose(hFind);
        }
    }
}
// for demonstration purposes, we just print the name
void PrintEachFoundName(__in PCWSTR pszFile)
{
    _putws(pszFile);
}
int __cdecl wmain(int argc, wchar_t **argv)
{
    for (int i = 1; i < argc; i++) {
        EnumerateAllNames(argv[i], PrintEachFoundName);
    }
    return 0;
}

```

**Update:** Minor errors corrected, as noted by [acq](#) and [Adrian](#).

[Raymond Chen](#)

**Follow**



