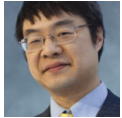


# How can I tell whether two paths reside on the same underlying volume, so I can know whether they can be hard-linked?

 [devblogs.microsoft.com/oldnewthing/20170707-00](https://devblogs.microsoft.com/oldnewthing/20170707-00)

July 7, 2017



Raymond Chen

A customer wanted to know whether there was a way to determine whether two paths reside on the same underlying volume. They were concerned that due to tricks like `SUBST` and volume mount points, the naïve algorithm of comparing drive letters may prove inadequate.

You can use `GetFileInformationByHandle` to get the volume serial number and the file index. Even better is to call `GetFileInformationByHandleEx` with the `FileIdInfo` information code, because ReFS use a 64-bit volume identifier and a 128-bit file identifier (double the size of NTFS).

There is a small chance that two volumes might happen to have the same volume identifier, seeing as a 32-bit value (or a 64-bit value in the case of ReFS) is not quite enough to guarantee global uniqueness. You can try to filter out that case by using `OpenFileById` and passing the first file's file handle and the second file's ID. The `OpenFileById` uses the file handle to decide what volume to use, so passing the first file's file handle means "Open the new file from the same volume that the old file is on." You can then compare the newly-opened file's object GUID against the object GUID of the intended second file and see if they match.

Actually, I guess you could have gone straight for the object GUID from the outset: Obtain the object GUID for the second file, then try to open the file by its GUID (relative to the first file).

But wait, we're so distracted by answering the customer's question that we don't understand the customer's *problem*.

The customer explained that they want to create a content-addressable database, but if they find that there's already a database on the same volume, then they can just hard-link the contents together and save disk space.

Oh, so the problem isn't "Determine whether two files are on the same volume (so I can hard-link them)." The problem is "Determine whether two files can be hard-linked."

That problem is much easier to solve. The way to determine whether two files can be hard-linked is to try hard-linking them and seeing if it works. As [Jared Parsons](#) noted, the file system is unpredictable.

Do the operations you want and deal with the consequences of failure [...]. To do anything else involves an unreliable prediction in which you still must handle the resulting [failure].

(Jared's article is written for a C# audience, so he talks about exceptions, because that's the C# paradigm for error handling.)

[Raymond Chen](#)

**Follow**

