Why does IsPathRelative return FALSE for paths that are drive-relative?

devblogs.microsoft.com/oldnewthing/20180222-00

February 22, 2018

Raymond Chen

MSDN distinguishes between fully-qualified and relative paths. According to that discussion:

> A file name is relative to the current directory if it does not begin with one of the following:
>
> - A UNC name,
> - A disk designator with a backslash,
> - A single backslash, for example, "\directory" or "\file.txt". This is also referred to as an *absolute path*.
>
> If a file name begins with only a disk designator but not the backslash after the colon, it is interpreted as a relative path to the current directory on the drive with the specified letter.
>
> A path is also said to be relative if it contains "double-dots"; that is, two periods together in one component of the path.
>
> Relative paths can combine both example types, for example "C:..\tmp.txt".

Okay, so what exactly is the definition of a relative path?

It's sort of like art. You know it when you see it.

There are some things that are clearly absolute paths. These are known as fully-qualified paths. They specify both a volume designator and a path relative to the root of that volume.

There are also some things that are clearly relative paths, like `dir\file.txt`.

And then there are these weird intermediate cases that everybody understands but nobody really knows how to classify like `C:..\dir\file.txt`.

Here's a table of possibilities.

Example	Classification	IsPathRelative?

`\\Server\Share\Dir\File.txt`	UNC absolute	No
`C:\Dir\File.txt`	Drive absolute	No
`C:Dir\File.txt`	Drive relative	No
`\Dir\File.txt`	Rooted	No
`Dir\File.txt`	Relative	Yes

The first two rows and the last row are not controversial. The third and fourth rows, on the other hand, are problematic. They live in this shadowy world, half-absolute and half-relative.

Raymond Chen

Follow