Why does the Directory.GetFiles method sometimes ignore *.html files when I ask for *.htm?

devblogs.microsoft.com/oldnewthing/20140313-00

March 13, 2014

Raymond Chen

The documentation for the `Directory.GetFiles` method says

> When using the asterisk wildcard character in a *searchPattern*, such as "*.txt", the matching behavior when the extension is exactly three characters long is different than when the extension is more or less than three characters long. A *searchPattern* with a file extension of exactly three characters returns files having an extension of three or more characters, where the first three characters match the file extension specified in the *searchPattern*. A *searchPattern* with a file extension of one, two, or more than three characters returns only files having extensions of exactly that length that match the file extension specified in the *searchPattern*. When using the question mark wildcard character, this method returns only files that match the specified file extension. For example, given two files, "file1.txt" and "file1.txtother", in a directory, a search pattern of "file?.txt" returns just the first file, while a search pattern of "file*.txt" returns both files.

A customer reported that one of their programs stopped working, and they traced the problem to the fact that a search for `*.htm` on some machines was no longer return files like `awesome.html`, contrary to the documentation. What's going on? What's going on is that the documentation is trying too hard to explain an observed behavior. (My guess is that some other customer reported the behavior, and the documentation team incorporated the customer's observations into the documentation without really thinking it through.) The real issue is that the `GetFiles` method matches against both short file names and long file names. If a long file name has an extension that is longer than three characters, the extension is truncated to form the short file name. And it is that short file name that gets matched by `*.htm` or `*.txt`. Even as originally written, in the presence of short file names, the documentation is wrong, because it would imply that a search for `reallylong*.txt` could match `reallylong_filename.txtother`. But try it: It doesn't. That's because the short name is probably `REALLY~1.TXT`, and that doesn't match `reallylong*.txt`. What happened is that short file name generation was disabled on the drive at the time the files were created, so there was no short file name available, so there was consequently no `SHORTN~1.HTM` file to match against.

The documentation should really say something more like this:

> Because this method checks against file names with both the 8.3 file name format (if available) and the long file name format, a search pattern like "*.txt" may return unexpected results. For example, the file `longfilename.txtother` may be returned if the short file name for the file is `LONGFI~1.TXT`.

Update: It looks like the documentation has added my alternate remarks, but they kept the original misleading remarks as well, so now it's double-confusing. And to make things even more confusing, the original misleading remark has been made *even more misleading* in the part where it talks about question marks overriding the three-character rule. This is another failed attempt to explain observed behavior. If you search for "file?.txt", it will not match "file1.txtother". But the reason is not that the question mark overrides the three-character rule. The reason is that the short name for "file1.txtother" is "FILE1~1.TXT", and the question mark matches only one character.

Raymond Chen

Follow