Why do NTFS and Explorer disagree on filename sorting?

devblogs.microsoft.com/oldnewthing/20050617-10

June 17, 2005

Raymond Chen

Some people have noticed that NTFS automatically sorts filenames, but does so in a manner different from Explorer. Why is that? For illustration purposes, I created files with the following names:

Name	Code point	Description
a	U+0061	Latin small letter A
b	U+0062	Latin small letter B
×	U+00D7	Multiplication sign
å	U+00E5	Latin small letter A with ring above
ø	U+00F8	Latin small letter O with stroke

And here's the sort order for various scenarios, at least on my machine. (You'll later see why it's important whose machine you test on.)

Plain "dir" command		
a	U+0061	Latin small letter A
b	U+0062	Latin small letter B
å	U+00E5	Latin small letter A with ring above
×	U+00D7	Multiplication sign
ø	U+00F8	Latin small letter O with stroke

"dir /on"

×	U+00D7	Multiplication sign
a	U+0061	Latin small letter A
å	U+00E5	Latin small letter A with ring above
b	U+0062	Latin small letter B
ø	U+00F8	Latin small letter O with stroke

Explorer sorted by name		
×	U+00D7	Multiplication sign
a	U+0061	Latin small letter A
å	U+00E5	Latin small letter A with ring above
b	U+0062	Latin small letter B
ø	U+00F8	Latin small letter O with stroke

First, notice that Explorer and "dir /on" agree on the alphabetic sort order. (Once you throw digits into the mix, things diverge.) This is not a coincidence. Both are using the default locale's word sort algorithm. Why does the raw NTFS sort order differ? Because NTFS's raw sort order has different goals. The "dir /on" and Explorer output are sorting the items for humans. When sorting for humans, you need to respect their locale. If my computer were in Sweden, Explorer and "dir /on" would have sorted the items in a different order:

×	U+00D7	Multiplication sign
a	U+0061	Latin small letter A
b	U+0062	Latin small letter B
å	U+00E5	Latin small letter A with ring above
ø	U+00F8	Latin small letter O with stroke

You can ask a Swede why this is the correct sort order if you're that curious. My point is that different locales have different sorting rules. NTFS's raw sort order, on the other hand, is not for humans. As we saw above, sorting for humans can result in different results depending on which human you ask. But there is only one order for files on the disk, and NTFS needs to apply a consistent rule so that it can find a file when asked for it later. In order to maintain this consistency, the NTFS raw sort order cannot be dependent upon such fickle properties as

the current user's locale. It needs to lock in a sort algorithm and stick to it. As Michael Kaplan pointed out earlier, NTFS captures the case mapping table at the time the drive is formatted and continues to use that table, even if the OS's case mapping tables change subsequently. Once the string has been converted to uppercase, it then needs to be sorted. Since this is not for humans, there's no need to implement the complex rules regarding secondary and tertiary keys, the interaction between alphanumerics and punctuation, and all the other things that make sorting hard. It just compares the code points as binary values, also known as an ordinal sort.

In summary, therefore, Explorer sorts the items so you (a human) can find them. NTFS sorts the items so it (the computer) can find them. If you're writing a program and you want the results of a directory listing to be sorted, then sort it yourself according to the criteria of your choice.

Raymond Chen

Follow