

The story of MUI, as told by others (with some commentary)

 devblogs.microsoft.com/oldnewthing/20101104-00

November 4, 2010



Raymond Chen

First, the story as told by others:

Now the question you're all going to ask so I may as well answer it: Why is this information kept in a `desktop.ini` file instead of being attached to the file itself (say, in an alternate stream)? If it were in an alternate stream, then it would track the file when it was moved or copied. First answer: Because alternate streams require NTFS. Localized names were introduced in Windows 2000, and Windows 2000 gave you the option of formatting your drive as FAT or NTFS. (It wasn't until Windows Vista that NTFS became mandatory.) Therefore the mechanism for localized names needed to work when your drive was formatted as FAT. Okay, fine, maybe you tell people, "Sorry, this feature requires NTFS." All those multinational corporations who use FAT-formatted drives in the year 2000 are screwed. Well, placing the information in an alternate data stream means that each file would have to be accessed *just to get its name*. Remember, it's more efficient when you buy in bulk. Consider a directory with 500 files. A simple directory listing like one provided by `cmd.exe` takes, say, seven I/O requests (open directory, five "give me the next 100 files", close directory). If each file had to be opened in order to probe for an alternate stream, you now have 507 I/O requests: open directory, five "give me the next 100 files", close directory; and then 500 "open alternate stream on this file" calls that fail. And that was the optimistic case where the localized name doesn't exist. For the pessimistic case where every file has a localized name, the I/O count balloons to 1507: open directory, five "give me the next 100 files", close directory; and then $500 \times$ (open alternate stream, read localized name, close alternate stream). You increased the number of I/O requests by a factor of over 200. And when you are looking at files on a slow network (hello, multinational company with servers all over the world), that factor can be deadly. Suppose the ping time to the server is 500ms. The `cmd.exe` program gets you the directory listing in three and a half seconds. The alternate data stream localized name version takes *twelve minutes*. On the other hand, if all the localized names are placed in a single file, then the I/O count is only 10: open directory, five "give me the next 100 files", close directory; open `desktop.ini`, read contents, close `desktop.ini`. We're down to five seconds. Not as good as three and a half seconds, but way better than twelve minutes. And if, in the common case, the directory contains no localized

names, the `open desktop.ini` step fails, and we save two I/O's, bringing our time down to four seconds. And then you can be clever and say, "Wait a second, I already have the directory listing; I can just look at the results to see if `desktop.ini` is on the list. If not, then I don't even have to bother trying to open it!" Now you are back to three and a half seconds. Okay, maybe you tell people, "Sorry, this feature requires a high-speed network." All those multinational corporations with servers around the world are screwed. The theoretically highest-speed network connection between New York and Tokyo has a 72ms ping time because that's how fast it takes light itself to travel that distance and back. Your 500-file directory takes nearly two minutes to display. Seeing as multinational corporations were the initial target audience for the MUI feature (since they're the ones who are most likely to have users with different language preferences), designing your feature so that your target audience can't use it seems like a pretty bad execution plan. A file's name is traditionally considered metadata, and traditionally, metadata is visible without requiring access to the file. A file can show up in a directory listing even though you don't have access to it. But if the file's localized name is stored in the file itself, you now have the situation where you don't even know the name of the file because you can't access it. This is even worse than *I'm sorry, you don't have permission to know where this shortcut file should show up in the Explorer window.* This would be *I'm sorry, you don't have permission to know the name of the file in this directory listing.* Remember that reading from an alternate data stream triggers a recall if the file had been archived to tape. You don't want to have to restore an entire directory from tape just because somebody opened the folder in Explorer.

And finally, alternate data streams are very fragile. Any tool that processes a file has a decent chance of inadvertently destroying the alternate data streams. (And good luck if your backup program doesn't preserve the alternate data streams.)

Raymond Chen

Follow

