If you can use GUIDs to reference files, why not use them to remember "recently used" files so they can survive renames and moves?

June 12, 2019

Raymond Chen

You can ask for a GUID identifier for a file, and use that GUID to access the file later. You can even recover a (perhaps not *the*) file name from the GUID.

David Trapp wishes programs would use GUIDs to reference files so that references to recently used files can survive renames and moves.

Be careful what you wish for.

It is a common pattern to save a file by performing two steps.

- Create a temporary file with the new contents.
- Rename the original file to a `*.bak` or some other name.
- Rename the temporary file to the original name.
- (optional) Delete the `*.bak` file.

Programs use this multi-step process so that an the old copy of the file remains intact until the new file has been saved successfully. Once that's done, they swap the new file into place.

Unfortunately, this messes up your GUID-based accounting system.

If you tracked the file by its GUID, then here's what you see:

- Create a temporary file, which gets a new GUID.
- Rename the original file. It retains its GUID but has a new name.
- Rename the temporary file file. It retains its GUID but has a new name.

The GUID that you remembered does not refer to the new file; it refers to the old file. Even worse, if the program took the optional step of deleting the renamed original, you now have a GUID that refers to a deleted file, which means that when you try to open it, the operation will fail.

Programs can avoid this problem by using the `ReplaceFile` function to promote the temporary file. The `ReplaceFile` function preserves the file identifier, among other things.

In practice, use of the `ReplaceFile` function is not as widespread as you probably would like, so using only GUIDs to track files will technically track the file, but may not track the file you intend. Because people still think of the file name as the identifier for a file, not its GUID.

Raymond Chen

Follow