Some known folders cannot be moved, but others can, and you'll just have to accept that

devblogs.microsoft.com/oldnewthing/20100806-00

August 6, 2010

Raymond Chen

Locations in the shell known folder system can be marked as `KF_CATEGORY_FIXED`, which makes them immovable. Conversely, if a file system folder is not immovable, then it can be moved. This dichotomy appears simple and unworthy of discussion, except that customers sometimes have trouble incorporating this concept into their world view.

> I have some code that calls `SHSetFolderPath`, and it works for most of the folders I'm interested in, but for some `CSIDL` values like `CSIDL_COMMON_APPDATA`, it fails with `E_INVALIDARG`. Doesn't matter if I run elevated or not. What am I doing wrong?

The difference is that `CSIDL_COMMON_APPDATA` (known under the New World Order as `FOLDERID_ProgramData`) is marked as `KF_CATEGORY_FIXED` so it cannot be moved.

> Is there a way that we can override the `KF_CATEGORY_FIXED` flag and move it anyway?

Nope. It's immovable. Sorry. You'll just have to accept that that folder will not go where you want it. The very next day, we got a complementary question from an unrelated customer:

> We have a program that monitors a known folder, and it goes haywire if the user changes the location of the folder while it's being monitored. Is there a way to prevent the user from moving the folder?

If a known folder can be moved, then you have to accept that it can be moved. You can't override its category and force it to be `KF_CATEGORY_FIXED` just because your life would be easier if it were. I found it interesting that we got two requests on consecutive days asking for what appear to be opposite things: "I want to force this folder to be movable" and "I want to force this folder to be immovable." I can only imagine what would happen if both programs are running at the same time!

What the program can do is register an `IFileIsInUse` on the directory so that it will be called when somebody wants to move it. At least that way it knows when scary things are about to happen and can prepare itself for the changes that lie ahead. I'm told that a sample program illustrating `IFileIsInUse` is in the Windows 7 SDK under the directory `winui\Shell\AppPlatform\FileIsInUse`. There's also an old article on the subject over on the now-defunct Shell Revealed blog.