

# Can I be sure that turning on automatic generation of short file names will get me short file names?

[devblogs.microsoft.com/oldnewthing/20181004-00](https://devblogs.microsoft.com/oldnewthing/20181004-00)

October 4, 2018



Raymond Chen

A customer wanted to know if administrative permissions were sufficient to enable generation of short file names (8 dot 3). Their plan was to set the registry key, call `GetShortFileName` to get the short file names, and then restore the registry key to the value it had.

They wanted to know if there were any group policies that would override the registry key and thereby foil their fiendish plan.

That registry key is the one controlled by group policy, so you could in theory get unlucky and a group policy refresh could occur just after you updated the value.

But wait, let's try to understand the customer's problem before coming up with a solution.

First of all, the customer's explanation didn't make sense. You don't have to change the policy in order to call `GetShortFileName`. The short file name exists (or doesn't) regardless of the policy setting. Short file names are generated automatically at the time a file is created, and it is at the point of file creation that the policy is consulted to determine whether to auto-generate a short file name. Changing the policy for short file name generation does not retroactively add or remove short file names for existing files.

The customer clarified that they did leave out a step in their quick description:

- Set the registry key.
- **Install their program.**
- Call `GetShortFileName`.
- Restore the registry key.

They installed their program with short file names enabled so that the `GetShortFileName` would indeed get a short name. They use this short name to ensure that there aren't any spaces-related command line parsing errors.

Okay, the correct way to fix spaces-related command line parsing errors is not to make the spaces go away. It's to fix your command line parser so you don't choke on file names with spaces in their name!

The customer explained that their program is just fine with files that have spaces in their name, but their program is a utility program, and it is used by other programs and scripts. Those other programs and scripts have a Unix heritage, and Unix file names rarely have spaces in their name. Consequently, those programs and scripts tend to have poor support for files with spaces in their name.

The customer was hoping to force generation of short file names during their program installation, so that those external programs can be given a spaces-free path to the program.

Okay, so the first note is that there is no way to absolutely guarantee that there will be a spaces-free short name for a file, because support for short file names is gradually fading away. Short file names were originally created to maintain backward compatibility with 16-bit programs, but the population of 16-bit programs has been dwindling for quite some time, especially since 64-bit Windows doesn't support them natively. ReFS and exFAT don't support short file names. Network shares from Unix systems rarely do. In general, any file system invented in the past 15 years or so will most likely not support short file names.

Furthermore, since short names are auto-assigned at the point of creation, it means that if the user installs the program into a pre-existing directory that lacks a short name, then you're not going to have a short name for the full path. Setting the registry key will not be sufficient by itself.

The customer thought about the situation for a while and came up with a different solution: If the user chooses to install the program into a path that contains spaces, then their installer also creates a symbolic link in a path with no spaces<sup>1</sup> that points to the installation directory.

<sup>1</sup> They didn't say exactly where, but I suspect they put it in `C:\ProgramData`.

Raymond Chen

**Follow**

