How do I get the current directory for a non-current drive?

April 12, 2017

Raymond Chen

As we learned some time ago, each drive has its own current directory for backward-compatibility with DOS 1.0 (which predates the concept of directories). This is simulated internally by magic environment variables. But those magic environment variables are internal and are therefore not contractual.

What is the supported way of finding out the current directory for a non-current drive?

One approach I've seen is to perform this sequence of operations:

```
GetCurrentDirectory(ARRAYSIZE(originalDirectory), originalDirectory);
SetCurrentDirectory(TEXT("X:"));
GetCurrentDirectory(ARRAYSIZE(currentDirectoryOnX), currentDirectoryOnX);
SetCurrentDirectory(originalDirectory);
```

One problem with this approach is that it may not be able to return to the original directory if the original directory was deleted or is otherwise not settable.

Another problem with this approach is that it applies a global solution to a local problem: The current directory is a process-wide resource, so if there is another thread that is performing an operation that depends on the current directory, you just messed up that thread.

The supported way to get the current directory for a non-current drive is to call `GetFull-PathName` and pass just the drive letter and a colon. This will resolve the drive letter to the current directory for that drive.

```
TCHAR buf[BUFFERSIZE];
TCHAR *file;
GetFullPathName(TEXT("X:"), BUFFERSIZE, buf, &file);
```

Assuming the call succeeds, the result in the buffer is the current directory for drive `X:`.

Raymond Chen

Follow