Why does CreateEvent fail with ERROR_PATH_NOT_FOUND if I give it a name with a backslash?

devblogs.microsoft.com/oldnewthing/20120112-00

January 12, 2012

Raymond Chen

A customer reported that the `CreateEvent` function was failing with the unusual error code `ERROR_PATH_NOT_FOUND`:

```
HANDLE h = CreateEvent(0, FALSE, TRUE, "willy\\wonka");
if (h == NULL) {
 DWORD dwError = GetLastError(); // returns ERROR_PATH_NOT_FOUND
 ...
}
```

The customer continued, "The documentation for `CreateEvent` says that the `lpName` parameter must not contain the backslash character. Clearly we are in error for having passed an illegal character, but why are we getting the strange error code? There is no file path involved. Right now, we've added `ERROR_PATH_NOT_FOUND` to our list of possible error codes, but we'd like an explanation of what the error means."

Okay, first of all, building a table of all known error codes is another compatibility problem waiting to happen. Suppose in the next version of Windows, a new error code is added, say, `ERROR_REJECTED_BY_SLASHDOT`. What will your program do when it gets this new error code?

Now back to the error code. There is no file path involved here, so why is there a path-not-found error?

Because it's not a file system path that failed. It's an object namespace path.

If a backslash appears in the name of a named object, it is treated as a namespace separator. (If there is no backslash, the name is interpreted as part of the Local namespace.) And the call fails with a path-not-found error since there is no namespace called `willy`, so the path traversal inside the object namespace fails.

The treatment of the backslash as a namespace separator is sort of alluded to in the very next sentence of the documentation: “For more information, see *Kernel Object Namespaces*.” The following paragraph also expands upon this idea: “The object can be created in a private namespace. For more information, see *Object Namespaces*.” The documentation sort of assumes you’ll follow the links and learn more about those namespacey things, at which point you’ll learn what that backslash in the object name really means (and why there is the rule about not allowing backslashes).

But here it is if you don’t want to try to figure it out:

“If you put a backslash in the name, it is treated as a namespace separator, and if you don’t know what a namespace is, then that’s probably not what you want. So don’t use backslashes unless you know what you’re doing.”

Raymond Chen

Follow