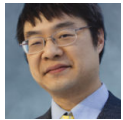


The case of the SHGetFolderPath(CSIDL_COMMON_DOCUMENTS) that returned ERROR_PATH_NOT_FOUND

 devblogs.microsoft.com/oldnewthing/20200520-00

May 20, 2020



Raymond Chen

A customer was experiencing a problem with the `SHGetFolderPath` function. Specifically, they had a program that called the function like this:

```
SHGetFolderPath(NULL, CSIDL_COMMON_DOCUMENTS, NULL,  
                SHGFP_TYPE_CURRENT, pathBuffer);
```

but it failed with error `0x80070003` which is the `HRESULT` version of `ERROR_PATH_NOT_FOUND`. The error occurs only when run from a Jenkins pipeline. If they run the program standalone, then the function succeeds and returns the expected result.

A procmon trace showed that the application tried to access the folder `C:\Windows\SysWOW64\autobuild\Documents`, which failed with `NAME_NOT_FOUND`. And that was the clue that broke things open.

The Common Documents folder defaults to `%PUBLIC%\Documents`. The `PUBLIC` environment variable's normal value is `C:\Users\Public`, but when the program runs as part of a Jenkins pipeline, the environment variable is set to `autobuild` for some reason.

This means that when the program calls `SHGetFolderPath` and asks for `CSIDL_COMMON_DOCUMENTS`, the system looks for `autobuild\Documents`, which doesn't exist, hence error `0x80070003`: "The system cannot find the path specified."

There are a number of environment variables that have special meaning, and you change them at your peril. You probably know about variables like `windir`, `ProgramFiles`, and `TEMP`, but there are quite a number of other special environment variables, and `PUBLIC` is one of them.

Armed with this information, the customer went back to see who was messing with the `PUBLIC` environment variable and try to get them to stop.

Raymond Chen

Follow

