

# Don't forget to double-null-terminate those strings you pass to SHFileOperation

[devblogs.microsoft.com/oldnewthing/20100218-00](http://devblogs.microsoft.com/oldnewthing/20100218-00)

February 18, 2010



Raymond Chen

About once every two months for the past six months (I stopped checking further back), somebody reports a problem with the `SHFileOperation` function. Often, they don't include very much information at all. They just say, "I call the function and it doesn't work." Here's an example:

I'm hitting a problem with `SHFileOperation` when using it to frob files in the gonzo directory when the user's SID ends in an odd number.

```
//  
// Delete the file.  
// szDeletePath contains the full path to the file.  
//  
shFileOp.hwnd = NULL;  
shFileOp.wFunc = FO_DELETE;  
shFileOp.pFrom = szDeletePath;  
shFileOp.pTo = NULL;  
shFileOp.fFlags = FOF_NO_UI;  
iRet = SHFileOperation( &shFileOp );
```

The function returns file not found, but the file is definitely there.

If you read the variable names carefully, you can see the problem.

The `pFrom` and `pTo` members of the `SHFILEOPSTRUCT` structure are double-null-terminated strings. (There's even a callout box for this in the MSDN documentation.) But a variable named `szDeletePath` is probably a single-null-terminated string. (The name for a double-null-terminated string would be `szzDeletePath`.)

My psychic powers tell me that `szDeletePath` is not double-null-terminated.

So far, my psychic powers haven't failed.

Now, you might say that the fact that people make this mistake so often is a sign that the function is flawed. And if the function were designed today, I would agree with you. But this function in its public form is over fifteen years old (and in its private form, is around 20 years old), and back in those days, programmers were assumed to have the time to understand the subtleties of what they were doing.

Raymond Chen

**Follow**

