

# The normal string manipulation functions stop on a null terminator, so be careful when manipulating double-null-terminated strings

 [devblogs.microsoft.com/oldnewthing/20100219-00](http://devblogs.microsoft.com/oldnewthing/20100219-00)

February 19, 2010



Raymond Chen

One of the many gotchas of working with double-null-terminated strings is accidentally using functions on them which were designed to operate on single-null-terminated strings. Now, you do need to use those single-null-terminated strings, but you also need to know when they won't do what you want.

One of the responses to my psychic detection that somebody passed a single-null-terminated string to SHFileOperation is, "Oh, no, I double-null-terminated it. Look:

```
sprintf(szDeletePath, "%s\0", file_to_delete);
```

See, I put an extra `\0` at the end."

Well, yeah, you put an extra `\0` at the end, but all that does is terminate the format string. The `sprintf` function accepts its format string as a traditional null-terminated string. When it sees the `\0` you stuck into the string, it thinks it found the end of the string. It can't read your mind and say, "Oh, this null is not a terminator. It's an embedded null."

A simple mistake, but something that needs to be called out. To be fair, most people recognize this mistake as soon as it's pointed out to them. You just have to remember to point it out to them.

Raymond Chen

**Follow**

