

If the `FormatMessage` function fails, and I requested that it allocate a buffer, do I have to free the buffer?

 devblogs.microsoft.com/oldnewthing/20250703-00/?p=111335

July 3, 2025



A customer called the `FormatMessage` function with the `FORMAT_MESSAGE_ALLOCATE_BUFFER` flag, and they weren't sure what to do if the function fails (by returning 0). Do they have to free the buffer by calling `LocalFree`?

No, you don't have to free the buffer. In fact, on failure, there is no buffer. The function failed to perform the desired operation, so there is nothing to clean up.

You can make things easier on yourself by pre-initializing the output pointer to `NULL`. That way, if the function fails, the pointer is still null. Then your logic can be "Go ahead and free the buffer," because the `LocalFree` function allows you to pass `NULL`, and it just ignores it. (This trick allows things like `wil::unique_hlocal_string` to work with `FormatMessage`.)

Thinking about the original question: You can't tell whether the reason for the function failure is that something went wrong during formatting or that something went wrong during allocation of the final output buffer. You could call `GetLastError()`, but if it returns `ERROR_OUT_OF_MEMORY`, you still don't know whether it ran out of memory during the formatting phase or during the final buffer allocation phase. Therefore, even if you wanted to free the buffer, you don't know whether you even got one in the first place.