

Why does `FormatMessage` say that `%0` terminates the message without a trailing newline? Is it secretly adding newlines?

devblogs.microsoft.com/oldnewthing/20191025-00

October 25, 2019



Raymond Chen

The documentation for the `FormatMessage` function says that `%0` terminates the message without a trailing newline. Why does there need to be a special format to prevent it from adding a newline? Does it secretly *add* newlines to my format strings?

Well, it's not `FormatMessage` that's adding the newline.

The `%0` sequence causes the `FormatMessage` function to stop processing the format string, as if the format string had terminated right there, rather than wherever it actually does terminate.¹

But why does this feature need to exist in the first place? I mean, if you don't want a newline at the end of the output, then don't put a newline at the end of your format string.

There are two ways to specify a format string to the `FormatMessage` function. One is to pass the format string explicitly, in which case you can certainly control whether there is a newline at the end of the format string or not.

The other is to ask that the format string come from a message resource. And that's the case that the `%0` format specifier was intended for.

The syntax for specifying message strings to the Message Compiler is to list each line of the message on a line by itself, with the message terminated by a line that begins with a period.

```
MessageId=0x1
Severity=Error
Facility=Runtime
SymbolicName=MSG_BAD_COMMAND
Language=English
You have chosen an incorrect command.
```

.

Since the way to specify the end of the message is to put a period at the start of a line, it means that the character that comes before the period is always a newline. (Well, except for the case where you have no lines at all.) The `%0` sequence is a workaround in the `Format-Message` function to let you stop processing the format string before it reaches that newline.

There are other workarounds in the `FormatMessage` function for limitations of the Message Compiler: You can use `%.` to generate a single period, if you want to start a message with a period. You can use `%b` to generate a single space. Check the documentation for the complete list.

If you're passing your own format string, then many of the special sequences in the format string aren't of much use to you, because you could have applied those sequences to the string before passing it to the `FormatMessage` function. The weird special sequences are primarily for the case where the strings came from the Message Compiler, and they give you a way to "pass through" special values that the Message Compiler syntax doesn't allow for.

¹ Note, however, that the format string must still be a proper string with a null terminator. The `%0` sequence does not absolve you of the responsibility to pass a proper null-terminated string. It does tell the `FormatMessage` to ignore the rest of the string, but there must still be a "rest of the string".

Raymond Chen

Follow

