

# What are these weird internal Visual C++ runtime functions named NLG?

[devblogs.microsoft.com/oldnewthing/20220126-00](https://devblogs.microsoft.com/oldnewthing/20220126-00)

January 26, 2022



Raymond Chen

If you're looking at dump files or doing live debugging, you may encounter functions whose names begin with NLG: `_NLG_Notify`, `_NLG_Notify1`, `_NLG_Dispatch`, `_NLG_Return`, `_NLG_Return2`<sup>1</sup> and a few more. What are they for, and what does NLG stand for?

These functions are part of the Visual C++ runtime's internal exception handling infrastructure. The “notify” functions are used to inform the debugger of the progress of exception handling, and the “return” functions are used to tell the debugger that an exception handler returned.

The sneaky thing is that the “notify” functions aren't really functions. They are labels, and the debugger is expected to set a breakpoint at those labels to learn of the progress of exception handling.

But what does NLG stand for?

It stands for “non-local goto”, which is a fancy way to say “jumping from one function directly to another.”

<sup>1</sup> I find it amusing that the second `_NLG_Notify` is called `_NLG_Notify1`, but the second `_NLG_Return` is called `_NLG_Return2`.

[Raymond Chen](#)

**Follow**

