

# Does DebugBreak work to launch the debugger, or doesn't it?

[devblogs.microsoft.com/oldnewthing/20170529-00](http://devblogs.microsoft.com/oldnewthing/20170529-00)

May 29, 2017



Raymond Chen

Jorge asked why the DebugBreak function stopped working. Specifically, why it doesn't launch the Visual Studio debugger.

Okay, first of all, the primary purpose of the `DebugBreak` function is not to launch the debugger. The primary purpose of the `DebugBreak` function is to trigger a break into any attached debugger. That still works the same as it always did.

The behavior Jorge asks about is a second-order side effect of the `DebugBreak` function.

When you perform a `DebugBreak`, the `EXCEPTION_BREAKPOINT` exception is raised. If a debugger is installed, it will intercept this exception and interpret it as a request to break into the debugger. If a debugger is not installed, then the exception is processed like any other exception, and any installed handler could step in and say, "Oh, yeah, `EXCEPTION_BREAKPOINT`. No problem. I'll handle that."

If a debugger is not installed and no code in the process handles the exception, then the exception goes to the current unhandled exception filter, and the default unhandled exception filter checks with Windows Error Reporting, and then displays a dialog box to inform the user of the problem.

If a just-in-time debugger is installed, that dialog box has an extra *Debug* button:



Contoso Deluxe

Contoso Deluxe has stopped working

A problem caused the program to stop working correctly. Windows will close the program and notify you if a solution is available.

Debug Close program

Clicking the *Debug* button launches your just-in-time debugger.

Notice that once you get past the point where no attached debugger has intercepted the `EXCEPTION_BREAKPOINT` exception, the rest of the processing doesn't treat the breakpoint exception differently from any other exception. In other words, instead of calling `DebugBreak`, you could have dereferenced a null pointer, written to a read-only page, executed an invalid instruction, or called `RaiseException`. Anything that causes an exception to be raised goes through the same sequence of events, and at the end of the day will show the same dialog box.

The `DebugBreak` function is not a "Launch the just-in-time debugger" function. It's a very specific kind of "try to crash the program" function that debuggers understand and intercept. But if there's no debugger attached, then the program crashes, and when a program crashes (for whatever reason), the just-in-time debugger gets a chance to step in.

Raymond Chen

**Follow**

