Why do you have to wait for Windows Error Reporting to check for solutions before it restarts the application?

devblogs.microsoft.com/oldnewthing/20120611-00

June 11, 2012

Raymond Chen

Leo Davidson wonders why you have to wait for Windows Error Reporting to check for solutions before it restarts the application. Why not do the two in parallel? Well, for one thing, when the application restarts, it might freak out if it sees a dead copy of itself. I know for sure that I would freak out if I woke up one morning and saw my own dead body lying next to me. While Windows Error Reporting is checking for a solution, it still has access to the carcass of the crashed application, because it may need to refer to it in order to answer follow-up questions from the server. (“Hey, was version 3.14 of PI.DLL loaded into the process when it crashed? If so, then I may have an idea what went wrong.”) And so that, if you ask it to submit the crash to Microsoft, it can grab the information it needs in order to generate the crash report. Now suppose you start up a new copy of the application right away. If the application is a single-instance program, it will look around for another copy of itself, and hey look, it’ll find its own lifeless body in the middle of the computer version of an autopsy. It will then try to send messages to the dead program, saying, “Hey, the user wants to open document X; could you do that for me?” And it won’t get a response because, well, the program is dead. It’s never going to respond. Some programs don’t even try to pass information along. They just find the existing copy of the program, and call `Set-ForegroundWindow` on its main window, thereby switching to it. Of course, what they tried to do was switched to a crashed program. Even worse, what if the second copy of the program tries to extract information from the existing copy of itself? If the existing copy crashed, it’s highly likely that the crash was caused by corruption in the program’s internal data structures. When the second copy tries to extract the corrupted data, it may itself crash. Immediately launching the replacement program creates a very quickly-growing pile of dead programs, and your screen basically gets spammed with Windows Error Reporting dialogs faster than you can click OK. The crashed program has effectively launched a denial of service attack against itself. Before trying to start the program again, Windows makes sure that the previous copy has received a proper burial. Because few programs are prepared to see their own cadaver.

Bonus chatter: Another common scenario is that the program crashes at startup. Automatically restarting the program would just launch another copy that immediately crashes. Again, you get into the situation where you get a dozen copies of the program launched per second, all of which immediately crash.

Raymond Chen

Follow