

The case of the programs that were launched with impossible command line options

 devblogs.microsoft.com/oldnewthing/20211231-00

December 31, 2021



Raymond Chen

Several years ago, a bug was filed automatically due to a spike in failures in the Start menu with a new crash profile. Investigation of this bug was rather complicated, because the crash was “impossible”.

Then again, a lot of failures seem to be “impossible”, but the fact that they’re happening proves that it’s possible, and you just have to do some sleuthing and adopt a more creative mindset to figure them out.

One of the tools for investigating these types of failures is seeing what other programs are running at the time, or what other programs crashed shortly before or after the failure occurred. In this case, whenever the crash occurred, there was one specific third-party program running. This program billed itself as a utility that boosts your system’s gaming performance by terminating all processes it deemed to be non-essential. Its advertising copy calls out useless “productivity apps” as one of those non-essential processes. (Yeah, how dare you let a computer be used for productivity? Can’t you see I’m playing a game?)

Apparently, what happens that when the program detects that you’re playing a game, it runs around and simply terminates all the non-essential processes.

I sure hope you saved your work.

And when the game is over, it relaunches all the programs it terminated, presumably on the assumption either that you didn’t care about your unsaved data, or that the program would go into crash recovery mode and restore its state from the most recent auto-save point.

And one of the programs that this utility decide was not worthy of keeping around was the Start menu, so it terminated it when your game started. And then when the game was over, it relaunched it.

And that’s what was causing the problem.

For you see, the utility relaunched the program as a normal program with no command line arguments. But the Start menu is a UWP program, so it is supposed to be launched a special way, with specific command line arguments, and it is supposed to be run in a low integrity app container. As a result, the Start menu found itself running in an unexpected environment, with an unexpected command line, and it realized that something was messed up and failed fast, leaving a nice corpse for Watson to analyze.

The issue was transferred to the vendor outreach team for them to inform the vendor of the problem and get them to stop terminating the Start menu.¹

Epilogue: It seems that the vendor got the message on their own, because they issued an update that addressed this problem before the vendor outreach team got a hold of them.

¹ More generally, instead of terminating UWP apps, just minimize them, which causes them to save their state and suspend. If the system is placed under memory pressure, the system will terminate suspended UWP apps automatically.

Raymond Chen

Follow

