The danger of making the chk build stricter is that nobody will run it

devblogs.microsoft.com/oldnewthing/20110718-00

July 18, 2011

Raymond Chen

Our old pal Norman Diamond suggested that Windows should go beyond merely detecting dubious behavior on debug builds and should kill the application when it misbehaves. The thing is, if you make an operating system so strict that the slightest misstep results in lightning bolts from the sky, then nobody would run it. Back in the days of 16-bit Windows, as today, there were two builds, the so-called *retail* build, which had assertions disabled, and the so-called *debug* build, which had assertions enabled and broke into the debugger if an application did something suspicious. (This is similar to today's terms checked and free.) Now, the Windows development team is big on self-hosting. After all, if you are writing the operating system, you should be running it, too. What's more, it was common to self-host the *debug* version of the operating system, since that's the one with the extra checks and assertions that help you flush out the bugs. As it happens, the defect tracking system we used back in the day triggered a lot of these assertions. As I recall, refreshing a query resulted in about 50 parameter validation errors caught and reported by Windows. This made using the defect tracking system very cumbersome because you had to babysit the debugger and hit "i" (for *ignore*) 50 times each time you refreshed a query. (As I noted in my talk at Reflections|Projections 2009, the great thing about defect tracking systems is that you will hate every single one you use. Sure, the new defect tracking system may have some new features and be easier to use and run faster, but all that does is delay the point at which you begin hating it.) If Windows had taken the stance that the slightest error resulted in the death of the application, then it would have been impossible for a member of the Windows development team to run the defect tracking system program itself, because once it hit the first of those 50 parameter validation error reports, the program would have been killed, and the defect tracking system would have been rendered useless. Remember, don't change program semantics in the debug build. That just creates Heisenbugs. I remember that at one point the Windows team asked the people who supported the defect tracking system, "Hey, your program has a lot of problems that are being reported by the Windows debug build. Can you take a look at it?"

The response from the defect tracking system support team was somewhat ironic: “Sorry, we don’t support running the defect tracking system on a debug build of Windows. We found that the debug version of Windows breaks into the debugger too much.”

Raymond Chen

Follow