

Puzzle: Can you explain this program's crash profile?

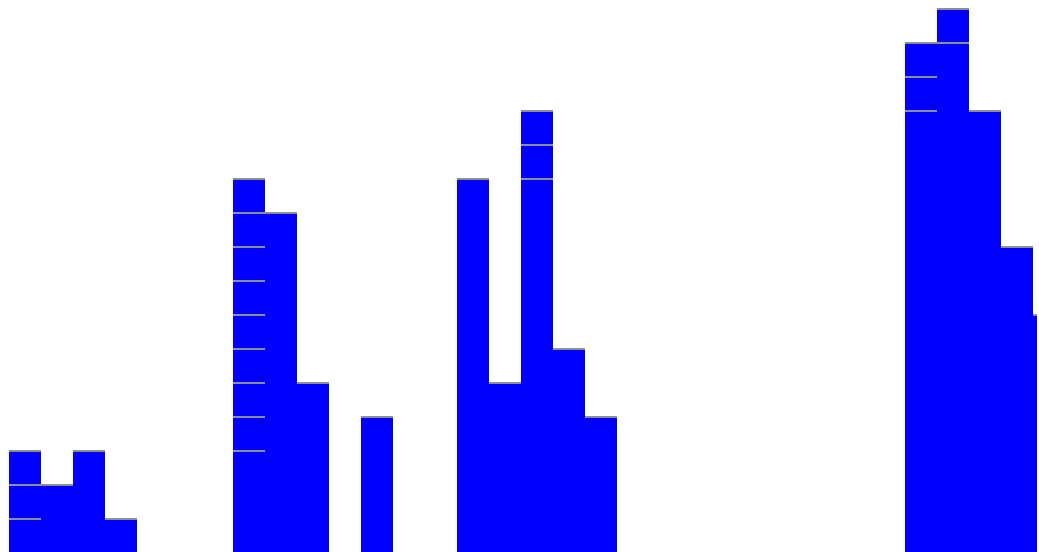
devblogs.microsoft.com/oldnewthing/20100602-00

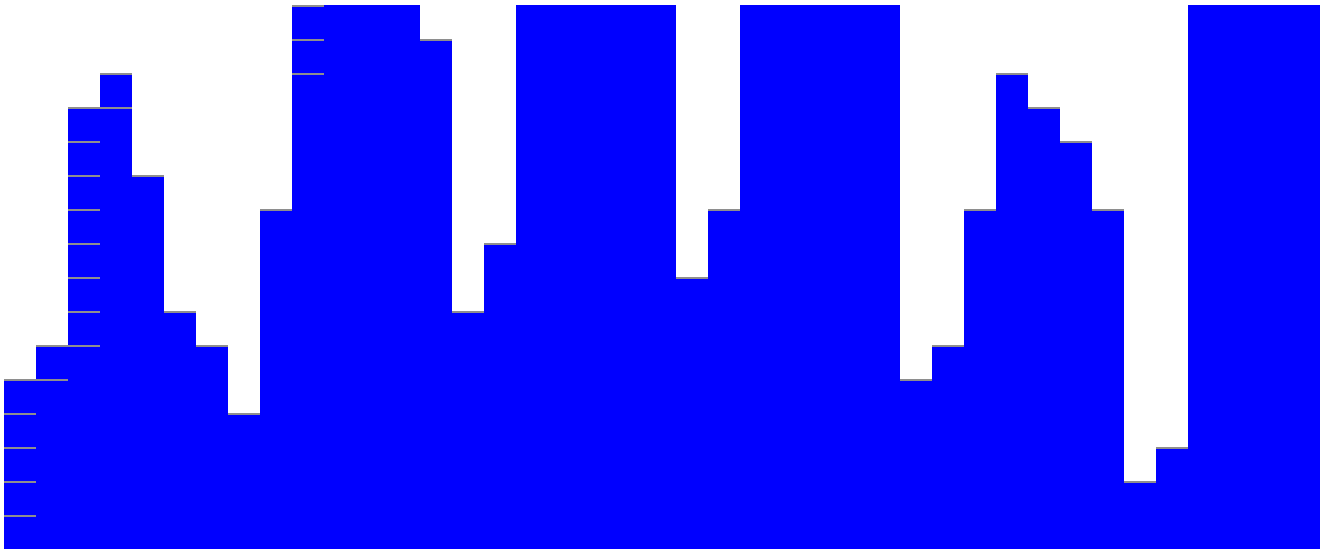
June 2, 2010



Raymond Chen

Some time ago, I was asked to help a customer study a set of crashes that had been collected by Windows Error Reporting. ([You too can sign up to obtain access to crash data for your application.](#)) The issue itself was the 325th most common crash in the ISV crash database, so fixing it would mean a lot toward improving the overall perceived stability of Windows. Fortunately, the issue was resolved relatively easily, but that's not what made the story interesting. What I found interesting was a little puzzle that faced me when I called up their crash profile. One of the items in the crash profile report is a histogram plotting how many crashes per day were reported over the past three months. Most crash profiles take the form of an erratic graph with random day-to-day fluctuations. Sometimes you'll see a gradual trend (for example, as more and more people upgrade to a newer version). But this one had a strong pattern:





The number of crashes per day remains high for several days, and then plummet for two days, then return to their high values, repeating on a regular cycle. It took me a bit of thought, but soon I understood why. Perhaps you can figure it out, too. Hints after the break.

If you actually stop and count (I didn't; I just eyeballed it), there are five days with high crash frequency, followed by two days with low crash frequency.

What happens on a seven-day cycle?

The five days with high crash frequency correspond to Monday through Friday; the two days with low crash frequency correspond to Saturday and Sunday.

The program in question targets a business audience. People use the program when they're at work (during the work week), but they don't use the program when they're at home (on the weekend). A program that isn't running can't crash.