GUIDs are designed to be unique, not random

devblogs.microsoft.com/oldnewthing/20120523-00

May 23, 2012

Raymond Chen

A customer liaison asked, “My customer is looking for information on the GUID generation algorithm. They need to select *N* items randomly from a pool of *M* (jury selection), and their proposed algorithm is to assign each item a GUID, then sort the items by GUID and take the first *N*.” (I’ve seen similar questions regarding using GUIDs for things like passwords or other situations where the programmer is looking for a way to generate a value that cannot be predicted.) The GUID generation algorithm was designed for uniqueness. It was not designed for randomness or for unpredictability. Indeed, if you look at an earlier discussion, you can see that so-called Algorithm 1 is *non-random* and *totally predictable*. If you use an Algorithm 1 GUID generator to assign GUIDs to candidates, you’ll find that the GUIDs are assigned in numerically ascending order (because the timestamp increases). The customer’s proposed algorithm would most likely end up choosing for jury duty the first *N* people entered into the system after a 32-bit timer rollover. Definitely not random. Similarly, the person who wanted to use a GUID for password generation would find that the passwords are *totally predictable* if you know what time the GUID was generated and which computer generated the GUID (which you can get by looking at the final six bytes from some other password-GUID). Totally-predictable passwords are probably not a good idea. Even the Version 4 GUID algorithm (which basically says “set the version to 4 and fill everything else with random or pseudo-random numbers”) is not guaranteed to be unpredictable, because the algorithm does not specify the quality of the random number generator. The Wikipedia article for GUID contains primary research which suggests that future and previous GUIDs can be predicted based on knowledge of the random number generator state, since the generator is not cryptographically strong. If you want a random number generator, then *use a random number generator*.

Bonus reading: Eric Lippert’s GUID Guide, part 1, part 2, and part 3.

Raymond Chen

Follow