A `<random>` Nomenclature Tweak

Document #: WG21 P0346R0
Date: 2016-05-25
Revises: None
Project: JTC1.22.32 Programming Language C++
Audience: LWG
Reply to: Walter E. Brown <webrown.cpp@gmail.com>

Contents

Abstract

We propose a very modest global change in C++ `<random>` nomenclature, namely to replace the term *uniform random number generator* by the more precise term *uniform random bit generator*.

Random numbers are a pain in the butt. However, they're terribly interesting, and very useful in many applications. . . .
— JULIENNE WALKER

1 Introduction[1]

At the time we were designing what became the C++11 random number facility, we had become somewhat concerned about the amount of nomenclature we were introducing. As part of an effort to keep at least some generally familiar terminology, we were induced to recycle the conventional term "random number generator." We combined it with a descriptive adjective to obtain *uniform random number generator* (URNG for short), a term of art now used in clause 26 to denote the algorithmic interface for types and objects that produce bit sequences in which each possible bit value is uniformly likely.[2]

A single call to a URNG object is allowed to produce and deliver many (typically 32 or more) bits, returning these bits as a single packaged value of an unsigned integer type.[3] Experimentation showed that this design consistently produces a considerable performance improvement when compared to our earliest (unpublished) efforts in which each call to a URNG object yielded but a single bit. However, the adopted design (sometimes characterized as *bit-production-in-bulk*) now appears to be misleading many programmers as to a URNG's intended use.

Copyright © 2016 by Walter E. Brown. All rights reserved.

[1] Adapted from §2, "Nomenclature matters," of [N3847]

[2] That is, each generated bit is (ideally) exactly as likely to be a 0 as it is to be a 1.

[3] Conceptually, we wanted many of the semantics a `vector<bool>` would give us. Since many of the algorithms we were standardizing already used similar encodings, we opted to mimic part of a typical `vector<bool>` implementation.

We have observed over the last few years that many programmers mistakenly use a URNG[4] as if it were a *random number distribution*, a term of art that C++ uses (since TR1 [ISO07]) to specify a very different set of requirements.[5] When invoked, a distribution object produces what is known in probability and statistics as a *random variate*; in the computing disciplines, such a variate has long been known as a *random number*. We conjecture that it is the extensive tradition behind the *random number* term that is misleading many programmers as to the purpose and correct use of what we termed a URNG.

With benefit of hindsight, we wish we had used a slightly different term in place of URNG, a term that more accurately conveyed the designed purpose of such types and objects as sources of randomness, rather than as sources of random variates. We had identified such a term in [N3847], and have since then discovered that our preferred term was already being used in the literature, e.g., in [Geisler].

Accordingly, **we now propose to use the similar but more precise term *uniform random bit generator*** (abbreviated URBG) in place of the current term *uniform random number generator*. With this small adjustment in nomenclature, it should be much clearer that there is no entity in the standard library that by itself directly corresponds to the traditional general notion of a random number generator.

2 Proposed wording

1. Make the following editorial adjustments in order to improve nomenclature throughout [N4582] (in particular, in library clauses 25 and 26 and corresponding headings and indexes):

- *uniform random ~~number~~bit generator*,
- *Uniform random ~~number~~bit generator*,
- **`UniformRandom~~Number~~BitGenerator`**, and
- UR~~N~~BG.

2. Also add a new subclause to Annex C's [diff.cpp14], as follows:

C.4.x Clause 26 Random number library

26.6
Change: Replace all occurences of the term *uniform random number generator*, and of its abbreviation URNG, by the term *uniform random bit generator* and corresponding abbreviation URBG.

Rationale: The new term more accurately reflects the design and intended use of the specified feature.

Effect on original feature: No code is affected by the change in nomenclature.

3 Acknowledgments

Many thanks to the readers of early drafts of this paper for their thoughtful comments.

4 Bibliography

[Geisler] Martin Geisler, Mikkel Krøigård, and Andreas Danielsen: "About Random Bits." 2004. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.90.3779.

[4] Calls to **`rand()`** have the same issues; see [Walker].

[5] Please see our earlier paper [N3551] for additional exposition regarding the roles of URNGs and distributions, their interaction, and the all-too-common anti-pattern that misuses a URNG.

[ISO07] International Organization for Standardization: "Information technology — Programming languages — Technical Report on C++ Library Extensions." ISO/IEC document TR 19768:2007.

[N3551] Walter E. Brown: "Random Number Generation in C++11." ISO/IEC JTC1/SC22/WG21 document N3551 (pre-Bristol mailing), 2013-03-12.
http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2013/n3551.pdf.

[N3847] Walter E. Brown: "Random Number Generation is Not Simple!" ISO/IEC JTC1/SC22/WG21 document N3847 (pre-Issaquah mailing), 2014-01-01.
http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2014/n3847.pdf.

[N4582] Richard Smith: "Working Draft, Standard for Programming Language C++." ISO/IEC JTC1/SC22/WG21 document N4582 (post-Jacksonville mailing), 2016-03-27.
http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2016/n4582.pdf.

[Walker] Julienne Walker: "Using **`rand()`**." In blog *Eternally Confuzzled*, undated.
http://eternallyconfuzzled.com/arts/jsw_art_rand.aspx.

5 Revision history

Version	Date	Changes
0	2016-05-25	• Published as P0346R0.