Subject: simpleEvtGen default EvtRandomEngine
Posted by Marius Mertens on Thu, 06 Aug 2009 15:57:17 GMT
View Forum Message <> Reply to Message

Hi all,

the default EvtRandomEngine used by simpleEvtGen is basically the generator below:

```
double EvtRandomEngine::random(){
  static unsigned long int next = 1;
  next=next*1103515245+123345;
  unsigned temp=(unsigned)(next/65536) % 32768;
  return ( temp + 1.0 ) / 32769.0;
}
```

At the first glance, it does seem rather simple, but I don't have sufficient experience with pseudo random number generation to estimate the properties of the sequences generated with above code. Can anybody tell me something about the properties/random quality of the sequences and their suitability for event generation?

Best regards,

Marius

Subject: Re: simpleEvtGen default EvtRandomEngine
Posted by StefanoSpataro on Thu, 06 Aug 2009 16:03:26 GMT
View Forum Message <> Reply to Message

Maybe it could be better to try to contact the original authors of EvtGen (I can give you their mail addresses, or you can find them in the web).
Hiowever, their idea is ho give a simple random generator, and leave then to the experiments to create their proper random generator.

Maybe somebody could comment on what is used in Babar or in the fast simulation framework.

Subject: Re: simpleEvtGen default EvtRandomEngine
Posted by Marius Mertens on Sat, 08 Aug 2009 10:57:56 GMT
View Forum Message <> Reply to Message

This is the recommendation I got from David Lange (EvtGen developer and support contact together with Anders Ryd):

Quote:Our recommendation is to definitely not use EvtRandomEngine for anything production-like. [our documentation I think says this..]

Marius

Subject: Re: simpleEvtGen default EvtRandomEngine
Posted by StefanoSpataro on Sat, 08 Aug 2009 11:01:34 GMT

View Forum Message <> Reply to Message

Exactly what I was saying,
each experiment should write its own random generator.

Subject: Re: simpleEvtGen default EvtRandomEngine
Posted by Marius Mertens on Sat, 08 Aug 2009 13:09:54 GMT
View Forum Message <> Reply to Message

Then, is there already somebody working on a default random engine which is suitable for production use?

Subject: Re: simpleEvtGen default EvtRandomEngine
Posted by StefanoSpataro on Sat, 08 Aug 2009 14:02:47 GMT
View Forum Message <> Reply to Message

Not inside our colaboration.
I was hoping we could steal something from Babar, Belle or Bes which are also using evtgen...
Thieves and spies are welcome

Subject: Re: simpleEvtGen default EvtRandomEngine
Posted by Bertram Kopf on Mon, 10 Aug 2009 13:16:48 GMT
View Forum Message <> Reply to Message

Hi Marius and Stefano,
as far as I know the default engine in the BaBar-like full simulation software is the "Ranecu" generator. It is also possible to use all other engines provided by CLHEP. Everything is steered via a random control mechanism which makes sure to reproduce sequences when it is needed to re-run specific jobs or even single events from one specific job. I posted already a link to a documentation on the random control package developed by BaBar (see posting:
http://forum.gsi.de/index.php?t=tree&goto=7708&rid=431&S=e6e3768b7c9b983ec4aae27f79f25413&srch=RandControl#msg_7708
)

Cheers,
Bertram.

Subject: Re: simpleEvtGen default EvtRandomEngine
Posted by Jens Sören Lange on Mon, 10 Aug 2009 13:48:46 GMT
View Forum Message <> Reply to Message

Hi Marius et al, if I am correct, this is actually the standard ANSI C random generator (just written out in numbers), as originally proposed by Kerninghan/Ritchie. The period is 2^31 and it is even used for openssl encryption, so I think it is actually not so bad. cheers, Soeren

Subject: Re: simpleEvtGen default EvtRandomEngine
Posted by Marius Mertens on Mon, 10 Aug 2009 14:20:44 GMT
View Forum Message <> Reply to Message

Hi Bertram,

thanks for your suggestion and the link!
Was this also "simply" plugged into EvtGen for BaBar? I understood from both your description and the documentation you included that the Ranecu based classes are rather a complete random sequence generation framework than just a simple generator?
I'm asking because for the way I use EvtGen up to now just a simple generator would perfectly suffice, granted its properties are considered sufficient for production use.

Maybe it's time for a Panda random task force

Subject: Re: simpleEvtGen default EvtRandomEngine
Posted by Bertram Kopf on Mon, 10 Aug 2009 16:02:43 GMT
View Forum Message <> Reply to Message

Hi Marius,

EvtGen as well as all other event generators are internal packages in the BaBar-like software. An abstract interface makes sure that these generators can be treated in the same way and don't contain any BaBar-related code.

Cheers,
Bertram.

Subject: Re: simpleEvtGen default EvtRandomEngine
Posted by Marius Mertens on Mon, 10 Aug 2009 16:05:02 GMT
View Forum Message <> Reply to Message

Hi Soeren,

yes you are correct, it's basically ANSI C rand() with a different b parameter.
Actually I didn't want to say the generator is bad in itself, rather that I'm not sure if the random sequences generated by it are suitable for our purpose.
For safety reasons, I'd feel better though if for production use we'd follow both Stefano's and the EvtGen authors' recommendation to use a more advanced generator.

Subject: Re: simpleEvtGen default EvtRandomEngine
Posted by Marius Mertens on Mon, 10 Aug 2009 16:32:06 GMT
View Forum Message <> Reply to Message

Hi Bertram,

sorry, my question was a bit unclear apparently, in fact I meant something much more trivial:

As the EvtGen EvtRandom interface is extremely simple (basically just the random() function which returns a random double) I was wondering if the more powerful Ranecu based framework including random number management is as easy to be understood and plugged into this single interface class as (just as a random example) one of the PRNGs from Boost? My motivation here is the learning curve. Getting random numbers from a boost library function is very easy (and would be sufficient for how I invoke EvtGen) but maybe at a later stage the management functions might be helpful?

Subject: Re: simpleEvtGen default EvtRandomEngine
Posted by Bertram Kopf on Mon, 10 Aug 2009 19:40:37 GMT
View Forum Message <> Reply to Message

Hi Marius,
mertens wrote on Mon, 10 August 2009 18:32
As the EvtGen EvtRandom interface is extremely simple (basically just the random() function which returns a random double) I was wondering if the more powerful Ranecu based framework including random number management is as easy to be understood and plugged into this single interface class as (just as a random example) one of the PRNGs from Boost? My motivation here is the learning curve. Getting random numbers from a boost library function is very easy (and would be sufficient for how I invoke EvtGen) but maybe at a later stage the management functions might be helpful?

As you can see in the class EvtRandomEngine the function random() is defined as a virtual method. That means that you simple have to write a new class derived from EvtRandomEngine and to overwrite this virtual function. In principle you can specify a specific random engine there. In addition you have to set your new defined engine via the static function "EvtRandom::setRandomEngine(newEngine)". An example can be found in the PandaRoot EvtGen package: the test program "ggEvtGen.cc" makes use of the CLHEP-generator "JamesRandom".

Cheers,
Bertram.

Subject: Re: simpleEvtGen default EvtRandomEngine
Posted by Marius Mertens on Mon, 10 Aug 2009 20:41:55 GMT
View Forum Message <> Reply to Message

Hi Bertram,

actually even simpler once again, sorry to keep bothering
How to plug different random engines into the EvtRandomEngine interface is perfectly clear.
I don't know much (despite the documentation you referred to) about the Ranecu framework with its management abilities, though.
As Stefano pointed out earlier that it would be advantageous to benefit from e.g., BaBar experience with randomness generation it seems worth a look.

So I think the question basically breaks down to this:
Would you consider it useful to plug that framework into EvtRandomEngine or would you consider it kind of an overkill in that context and rather implement a small, simple engine?

I took a look at JamesRandom which (purely from a point of provided function count) is also fairly simple. My guess from the Ranecu documentation is that in comparison it needs much more setup/options (as in complexity being the price one has to pay for more power)?
As I said, to me it'd be reasonable and perfectly sufficient to use something very simple there (like a Boost PRNG, which is what I'm actually using right now), but maybe the additional features of Ranecu might be useful in the future so the higher complexity in implementing this could pay off later?

As a general note I think it'd be useful to have a reasonable default PRNG implementation which is suitable for production use "as is", otherwise at a later stage other people might run into the same issue again and probably start similar considerations on what might be the "best" choice for that purpose.

Subject: Re: simpleEvtGen default EvtRandomEngine
Posted by Bertram Kopf on Tue, 11 Aug 2009 09:31:20 GMT
View Forum Message <> Reply to Message

Hi Marius,

mertens wrote on Mon, 10 August 2009 22:41
So I think the question basically breaks down to this:
Would you consider it useful to plug that framework into EvtRandomEngine or would you consider it kind of an overkill in that context and rather implement a small, simple engine?

In my point of view this question is very simple to answer. We need a proper random management tool for the event mass production (i.e. for the event generation, simulation, digitization, reconstruction as well as for the analysis). In addition it would be also very helpful for the code development to have the possibility to exactly reproduce single events of one job. Consider the following: If someone reports a bug, let's say the application crashes in event number xyz. Then it would be extremely helpful to reproduce this crash by starting directly at event xyz. Also for this purpose it would be nice to have something more powerful. Therefore I think it is important to have/to develop:

o a proper interface to all event generators

o to have a random control mechanism with the possibility to reproduce just specific events of one job

o (at least) to provide the possibility to switch between different random engines

o to provide an easy to use user interface

Cheers,

Bertram.

Subject: Re: simpleEvtGen default EvtRandomEngine
Posted by Marius Mertens on Tue, 11 Aug 2009 09:50:54 GMT
View Forum Message <> Reply to Message

Hi Bertram,

thanks a lot for your detailed answer (and your patience, I warned you the actual question was very simple)

I agree with the points you brought up and somehow this brings us back to that a common default solution would be good, as full randomness management is certainly more complex than the poor man's solution with Boost I just did for personal use because there seems to be no canonical way for acquiring randomness in Panda yet.

Subject: Re: simpleEvtGen default EvtRandomEngine - PndMTRandomEngine Suggestion
Posted by Marius Mertens on Wed, 19 Aug 2009 14:26:42 GMT
View Forum Message <> Reply to Message

Hi all,

I was asked to upload the random engine I'm currently using to have an alternative default until an elaborate solution is deployed.
This one is making use of the Boost libraries. So in case it is not available as a default, it'd be nice to have it added (maybe to the external packages or another appropriate place?).

So if anybody with write access to the corresponding source folders feels like replacing the current default, feel free to use this one

Best regards,

Marius

File Attachments
1) PndMTRandomEngine.hh, downloaded 459 times
2) PndMTRandomEngine.cc, downloaded 443 times

Subject: Re: simpleEvtGen default EvtRandomEngine - PndMTRandomEngine Suggestion
Posted by StefanoSpataro on Wed, 26 Aug 2009 11:49:46 GMT
View Forum Message <> Reply to Message

Just one question:
once I compile my code with this class, how to use this random generator in evtGen macro?

Subject: Re: simpleEvtGen default EvtRandomEngine - PndMTRandomEngine Suggestion
Posted by Marius Mertens on Wed, 26 Aug 2009 12:00:03 GMT
View Forum Message <> Reply to Message

Hi Stefano,

basically you just create an instance of it (seed is mandatory) and then pass it to the EvtGen constructor like this:

PndMTRandomEngine myRandomEngine(seed);

//Initialize the generator - read in the decay table and particle properties
EvtGen myGenerator("DECAY.DEC","evt.pdl", &myRandomEngine);

(this is an excerpt from the simpleEvtGen source)

Subject: Re: simpleEvtGen default EvtRandomEngine - PndMTRandomEngine Suggestion
Posted by StefanoSpataro on Wed, 26 Aug 2009 13:32:13 GMT
View Forum Message <> Reply to Message

It seems that in SL4.7 there is already an installed version of bost (or at least it appears in my /usr/include), which creates some conflicts. I have spent some time to modify the config.mk to be able to compile your class, but without anye ffort.
Which boost have you used? And how have you changed your config.mk?

Subject: Re: simpleEvtGen default EvtRandomEngine - PndMTRandomEngine Suggestion
Posted by Marius Mertens on Wed, 26 Aug 2009 14:04:13 GMT
View Forum Message <> Reply to Message

Hi Stefano,

I did not have to make any changes to the config.mk. When the Pnd* files are placed within EvtGenBase, a
make lib
should automatically compile them so they should be usable for a subsequent
make simple if you use it in simpleEvtGen.

Where does your compile fail? Are there any error messages referring to the Boost inclusion?

My boost version is rather old, 1.31.0. However, the interface should not have changed in such a way that it becomes incompatible with more recent releases.

Subject: Re: simpleEvtGen default EvtRandomEngine - PndMTRandomEngine

Suggestion
Posted by StefanoSpataro on Wed, 26 Aug 2009 14:14:57 GMT
View Forum Message <> Reply to Message

this is what I get without changing the config.mk:

Toggle Spoiler

```
make[1]: Entering directory `/home/stefano/july09/pandaroot/pgenerators/EvtGen/EvtGenBase'
Checking makedepend in EvtGenBase
makedepend: warning:  /usr/include/gnu/stubs.h:  non-portable whitespace encountered at line 7
makedepend: warning:  /usr/include/c++/3.4.6/i386-redhat-linux/bits/c++config.h:  non-portable whitespace encountered at line 140
makedepend: warning:  /usr/include/c++/3.4.6/i386-redhat-linux/bits/c++config.h:  non-portable whitespace encountered at line 142
makedepend: warning:  /usr/include/c++/3.4.6/bits/stl_algobase.h:  non-portable whitespace encountered at line 135
makedepend: warning:  /usr/include/c++/3.4.6/bits/stl_algobase.h:  non-portable whitespace encountered at line 136
makedepend: warning:  /usr/include/c++/3.4.6/bits/locale_facets.h:  non-portable whitespace encountered at line 132
makedepend: warning:  /usr/include/c++/3.4.6/bits/locale_facets.h:  non-portable whitespace encountered at line 1508
makedepend: warning:  /usr/include/c++/3.4.6/bits/locale_facets.h:  non-portable whitespace encountered at line 1533
makedepend: warning:  /usr/include/c++/3.4.6/bits/codecvt.h:  non-portable whitespace encountered at line 475
makedepend: warning:  /usr/include/c++/3.4.6/bits/locale_facets.h:  non-portable whitespace encountered at line 2963
makedepend: warning:  /usr/include/c++/3.4.6/bits/locale_facets.h:  non-portable whitespace encountered at line 4483
makedepend: warning:  /usr/include/c++/3.4.6/typeinfo:  non-portable whitespace encountered at line 49
makedepend: warning:  /usr/include/c++/3.4.6/typeinfo:  non-portable whitespace encountered at line 52
"/usr/include/boost/static_assert.hpp":66:
!defined(BOOST_BUGGY_INTEGRAL_CONSTANT_EXPRESSIONS) &&
!BOOST_WORKAROUND(__MWERKS__, < 0x3003)
                                                    ^---
expecting variable or number
makedepend: warning:  PndMTRandomEngine.cc (reading /usr/include/boost/cstdint.hpp), line 159: #    error defaults not correct; you must hand modify boost/cstdint.hpp
makedepend: warning:  PndMTRandomEngine.cc (reading /usr/include/boost/cstdint.hpp), line 189: #    error defaults not correct; you must hand modify boost/cstdint.hpp
makedepend: warning:  PndMTRandomEngine.cc (reading /usr/include/boost/cstdint.hpp), line 209: #    error defaults not correct; you must hand modify boost/cstdint.hpp
makedepend: warning:  PndMTRandomEngine.cc (reading /usr/include/boost/cstdint.hpp), line 247: #       error defaults not correct; you must hand modify boost/cstdint.hpp
"/usr/include/boost/random/linear_congruential.hpp":139:  !defined(__SGI_STL_PORT) &&
BOOST_WORKAROUND(__GNUC__, == 2)
                                                        ^--- expecting variable
or number
```

```
"/usr/include/boost/random/variate_generator.hpp":28:
BOOST_WORKAROUND(__BORLANDC__, <= 0x564)
                                                    ^--- expecting variable or number
"/usr/include/boost/random/variate_generator.hpp":114:
BOOST_WORKAROUND(__BORLANDC__, <= 0x564)
                                                    ^--- expecting variable or number
gcc -I. -DEVTSTANDALONE  -I/home/stefano/july09/cern/clhep/include
-I/home/stefano/july09/tools/root/include -I../ -I/home/stefano/july09/cern/clhep/include
-I/home/stefano/july09/tools/root/include -I../ -I/home/stefano/july09/cern/clhep/include
-I/home/stefano/july09/tools/root/include -I../ -c PndMTRandomEngine.cc
/usr/include/boost/random/mersenne_twister.hpp: In member function `void
boost::random::mersenne_twister<UIntType, w, n, m, r, a, u, s, b, t, c, l,
val>::seed(Generator&) [with Generator = const long unsigned int, UIntType = uint32_t, int w =
32, int n = 624, int m = 397, int r = 31, UIntType a = -1727483681u,int u = 11, int s = 7,
UIntType b = -1658038656u, int t = 15, UIntType c = -272236544u, int l = 18, UIntType val =
-948541730u]':
PndMTRandomEngine.cc:20:   instantiated from here
/usr/include/boost/random/mersenne_twister.hpp:96: error: `gen' cannot be used as a function
make[1]: *** [PndMTRandomEngine.o] Error 1
make[1]: Leaving directory `/home/stefano/july09/pandaroot/pgenerators/EvtGen/EvtGenBase'
```

Subject: Re: simpleEvtGen default EvtRandomEngine - PndMTRandomEngine Suggestion
Posted by Marius Mertens on Wed, 26 Aug 2009 14:26:26 GMT
View Forum Message <> Reply to Message

Hi Stefano,

so at least it does find your Boost libraries without any changes, which is basically good. However, the error you get was if I remember correctly a known bug in older Boost versions where the copy constructor was accidentally invoked upon instantiation. Actually, the explicit static_cast was supposed to work around that problem. A wild (not verified) guess is that casting to UIntType instead might help.

Subject: Re: simpleEvtGen default EvtRandomEngine - PndMTRandomEngine Suggestion
Posted by Elwin Dijck on Wed, 26 Aug 2009 15:22:00 GMT
View Forum Message <> Reply to Message

Hello all,

Perhaps you found this out already, but it seems that actually the Boost interface has changed: they initialize the Mersenne Twister (which actually needs a few hundred integers as initialization) now in a different way than in older versions. Instead of using a single integer, they want a function/functor that gives a sequence of values to initialize the internal state of the Mersenne Twister.

The Boost documentation says "The seeding from an integer was changed in April 2005 to

address a weakness." and refers to here.

Also see how ROOT initializes the Mersenne Twister.

So casting will not help, for the newer Boost versions the argument to seed() must be a function or functor. Simplest is to make a small class that has an operator() that just returns the seed number, though it might be better to use one of the more advanced schemes.

Best regards,
Elwin Dijck

Subject: Re: simpleEvtGen default EvtRandomEngine - PndMTRandomEngine Suggestion
Posted by Marius Mertens on Wed, 26 Aug 2009 15:22:56 GMT
View Forum Message <> Reply to Message

Hi again, you've probably already run into this, but when I cleaned up the class names, I accidentally deleted the "public" from the class declaration. It must say

class PndMTRandomEngine : public EvtRandomEngine

in PndMTRandomEngine.hh. Sorry!

Subject: Re: simpleEvtGen default EvtRandomEngine - PndMTRandomEngine Suggestion
Posted by Marius Mertens on Wed, 26 Aug 2009 15:29:48 GMT
View Forum Message <> Reply to Message

Hi Elwin,

thanks a lot, I was not aware of that interface change. I'm indeed using an older version, not sure about Stefano, though.

However, as it turns out to be more complex than expected to interface to Boost in a sufficiently generic way, what about then really changing to Root's TRandom3 as was suggested this morning? Especially since they also use a Mersenne Twister anyway? Stefano, would that maybe easier for you to integrate?

Subject: Re: simpleEvtGen default EvtRandomEngine - PndMTRandomEngine Suggestion
Posted by StefanoSpataro on Wed, 26 Aug 2009 15:32:26 GMT
View Forum Message <> Reply to Message

I think so.
With boost the problem is that by default under sl you have it installed, then you have conflicts not so easy to solve.

I will try to study the "TRandom" way...