

CoCoALib - Feature #84

Names for random streams

01 Feb 2012 15:31 - John Abbott

Status:	Closed	Start date:	01 Feb 2012
Priority:	Normal	Due date:	
Assignee:	John Abbott	% Done:	100%
Category:	Renaming	Estimated time:	0.00 hour
Target version:	CoCoALib-0.9953	Spent time:	23.50 hours
Description While revising old names containing ZZ into names containing BigInt, we stumbled upon RandomZZStream, and felt that RandomBigIntStream was becoming too long. As a consequence we decided to review the names given to the various random generators. This issue is for containing the discussion about possible new names for the random generators. closed: see CoCoALib documentation for random to see the final outcome.			
Related issues: Related to CoCoALib - Support #285: Notes for adding/removing files from docu... Closed 06 Dec 2012			

History

#1 - 01 Feb 2012 15:40 - John Abbott

At the moment the names all have the prefix **Random**; this could be abbreviated into **Rnd** or **Rand** -- Anna thinks **Rand** is a better choice. It is to be understood that all generators are pseudorandom (but that is far too long to be a prefix!).

The suffix is **Stream**. This may be a poor choice because in a C++ context "stream" is customarily associated with I/O (and the related syntax using << and >>). It is certainly possible to regard random number generation as *reading* from a random stream, but the current syntax is not compatible with the standard C++ streams.

CoCoALib's random streams have been designed to have C++ *input iterator* syntax. Can we find a name which suggests this?

The fully generic random generator uses the suffix **Source**; this is the same length as **Stream** and would avoid any possible confusion with C++'s notion of streams.

#2 - 01 Feb 2012 15:46 - John Abbott

Here is another idea. At the moment the names of the three random generators include the name of the type of random value created: **RandomBoolStream**, **RandomLongStream** and **RandomZZStream**. It should be easy to use templates and template specialization to allow names like these: **RandomStream<bool>**, **RandomStream<long>** and **RandomStream<BigInt>**

Advantage: This might make it easier to write templated code which needs to use random streams.

Disadvantage: we must write an (uninstantiable?) master template **RandomStream<...>**, and then write specializations. Also a user might naively think that the master template can be instantiated with any type.

Added 2012-12-04 JAA now thinks the template idea is not such a good one.

#3 - 01 Feb 2012 16:52 - John Abbott

We need to improve documentation to make the differences between RandomSource and the Random*Stream functions clearer.

#4 - 01 Feb 2012 16:59 - Anna Maria Bigatti

- Category set to Renaming
- Assignee set to John Abbott
- % Done changed from 0 to 10

After a long skype discussion I make my proposal for names:

```
RandomSource (universal)
RandomSourceBool
RandomSourceLong
RandomSourceBigInt
```

or

```
RandSource etc
```

for short, but I think it is more important to have short names for functions than for classes.

I would avoid templates, unless it is a real advantage in writing the code.

I think we should give greater emphasis to the universal one in the documentation while suggesting the specialized ones for optimization and debugging (with only minor changes in the syntax). So the user starts with the easy, universal, global one and then changes later, if needed.

added 2012 dec 04:

another proposal for names

```
RandomStreamLong
RandomStreamBool
RandomStreamBigInt
```

RandomSource has a different meaning, so it should have a different name.

#5 - 01 Feb 2012 17:06 - John Abbott

Here is another proposal for the new names:

```
RandGen
RandGenBool
RandGenLong
RandGenBigInt
```

Here my aim was to keep the names as short as possible.

Added 2012-12-05

Another proposal for names: (Seq is short for *sequence*)

```
RandomSeqLong
RandomSeqBool
RandomSeqBigInt
```

#6 - 01 Feb 2012 18:19 - Anna Maria Bigatti

these might be sources of inspiration

http://www.gnu.org/software/gsl/manual/html_node/Random-Number-Generation.html

<http://docs.scipy.org/doc/numpy/reference/routines.random.html>

<http://caml.inria.fr/pub/docs/manual-ocaml/libref/Random.html>

#7 - 01 Feb 2012 18:28 - Anna Maria Bigatti

A few ideas (brainstorming)

```
RandomBool(RndSrc)
RandomBool() // calling GlobalRandomSource()
```

```
RandomLong(RndSrc) // remove??
RandomLong(RndSrc, lwb, upb)
RandomLong(lwb, upb) // calling GlobalRandomSource()
```

I have implemented **RandomLong(lwb, upb)** and it works fine. I think it is the nicest interface.

#8 - 02 Feb 2012 11:24 - John Abbott

Anna: I have implemented **RandomLong(lwb, upb)** and it works fine. I think it is the nicest interface.

Good idea!!

#9 - 02 Feb 2012 16:52 - John Abbott

Anna: remove RandomLong(RndSrc)

OK, at least until someone says they really need it.

Added 2012-12-04: JAA confirms that the fn has been removed.

#10 - 04 Dec 2012 11:27 - John Abbott

Following Alessandra Caleo's criticisms (see issue [#75](#)), we have modified the documentation and examples to make them clearer.

JAA has undone Anna's wording in the doc which discouraged use of RandomLongStream (and also for booleans and BigInts).

We held a further verbal discussion about names. Anna pointed out that it is less important to make class names short than function names. JAA accepts this.

The remaining issue is whether to keep the names **RandomXXXStream** or switch to **RandomSourceXXX** as Anna suggested in point 4 above.

Note: I have not yet effected the renaming of files as this depends on our final choice for the class names.

Note 2: since the header files are all quite short, we could **put all the random functions into a single header file** -- is this a good idea? This would fit nicely if their names all begin with the same prefix, e.g. RandomSource

#11 - 04 Dec 2012 11:57 - Anna Maria Bigatti

Discussion via Skype:

instead of **sample** use **NextValue** to access a new element from a **RandomStream..**

#12 - 04 Dec 2012 12:02 - John Abbott

- % Done changed from 10 to 70

Approved the idea of **unifying the sources** after verbal discussion.

The best unified name seems to be **random**.

#13 - 04 Dec 2012 18:22 - Anna Maria Bigatti

- Target version set to CoCoALib-0.9953

AMB will unify the documentation files -- and update the names of the random sequence classes!

JAA will unify the code and simplify examples names.

Update 2012-12-05

JAA has unified code & simplified names of files, and changed names of the classes.

JAA has added a naming proposal to note 5 -- after verbal discussion we accept **RandomSeqLong** and friends. Implemented!

#14 - 05 Dec 2012 10:55 - John Abbott

- % Done changed from 70 to 80

#15 - 07 Dec 2012 14:52 - Anna Maria Bigatti

- Status changed from New to Resolved

#16 - 19 Dec 2012 17:25 - John Abbott

- Status changed from Resolved to Feedback

- % Done changed from 80 to 100

The unified documentation had not been finished. It is now much better, and can be regarded as good enough (though some more cleaning will eventually be needed).

Feedback @Anna please check the new doc for any problems and remove all od files; then we can close.

#17 - 23 May 2013 18:44 - John Abbott

- Status changed from Feedback to Closed

I have made a few final minor improvements to the documentation, otherwise no problems have arisen in the last 5 months; so I'm closing this issue.