

CoCoALib - Design #815

GlobalManager: Imminent disaster should be a proper error

23 Nov 2015 13:23 - John Abbott

<b>Status:</b>	Closed	<b>Start date:</b>	23 Nov 2015
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	John Abbott	<b>% Done:</b>	100%
<b>Category:</b>	Tidying	<b>Estimated time:</b>	2.01 hours
<b>Target version:</b>	CoCoALib-0.99550 spring 2017	<b>Spent time:</b>	2.10 hours
<b>Description</b>			
During destruction GlobalManager checks the ref counts of RingQQ and RingZZ; if they are too large then "Imminent disaster" is announced.			
Currently the announcement is simply printing a message on cerr; would it be better to throw an exception?			

History

#1 - 23 Nov 2015 13:25 - John Abbott

I've just made some big changes to matrices for orderings: everything compiles, but test-F5 produces "imminent disaster". Since it is just a printed message, it is a little awkward telling the debugger where to interrupt execution.

Anyway, it seems sensible to make it an exception.

What do you think?

#2 - 23 Nov 2015 13:57 - John Abbott

The real problem is that the check for still-existing RingElem values occurs inside the dtor for GlobalManager, and it is usually not a good idea for a dtor to throw exceptions... if a dtor throws during stack unwinding for another exception then C++ calls terminate immediately -- I think that means that Schwarzenegger comes through the door and "deals with" the programmer 8-{

#3 - 23 Nov 2015 14:04 - John Abbott

- Status changed from New to In Progress
- Assignee set to John Abbott
- % Done changed from 0 to 20

The real problem with terminate is that it gives the program no chance to handle the situation; in fact the only "bad things" which happen are some chunks of memory are not deallocated -- not ideal, but surely less severe than a brutal termination of the process.

Perhaps a reasonable compromise would be to put the lines which print out "imminent disaster" into a separate function; then I can tell the debugger to intercept that function.

Note that the GlobalManager dtor will call the dtors for RingQQ and RingZZ anyway; I'm not sure this can be avoided without throwing inside the dtor.

#4 - 23 Nov 2015 14:40 - John Abbott

I have put the lines into a separate (static member) function. What should it be called?

Currently I have called it **GlobalManager::DtorError**; note that I have deliberately not used the our prefix which we normally use for static mem fns.

It was OK telling the debugger to intercept CoCoA::GlobalManager::DtorError; a shorter name would be nicer though... any ideas?

#### #5 - 24 Nov 2015 14:22 - John Abbott

The message "Imminent disaster" is entertaining but not so useful to the user (especially if the message has not been seen before). What could be a more helpful message?

Error (GlobalManager Dtor): some CoCoALib RingElems still exist!

The problem is that such RingElem objects can no longer be properly destroyed: either the destructor will never be called, or if it is called then the owning ring has been destroyed (so something nasty will likely happen).

The full explanation is rather long to be an error message, but maybe it could be put in the documentation?

#### #6 - 21 Mar 2016 16:00 - John Abbott

- Status changed from *In Progress* to *Resolved*

- Target version changed from *CoCoALib-0.99540 Feb 2016* to *CoCoALib-0.99550 spring 2017*

- % Done changed from *20* to *80*

I have changed the "humorous" message **IMMINENT DISASTER** into the possibly more comprehensible **CoCoA: FAILURE DURING FINAL CLEAN-UP** (at least it is good that it now refers to CoCoALib).

I'm regarding this as having been resolved, and will close soon unless there are objections.

#### #7 - 21 Mar 2016 16:47 - John Abbott

I have added a new section to the documentation about the possible error message during destruction of GlobalManager.

#### #8 - 21 Sep 2016 14:37 - John Abbott

- Status changed from *Resolved* to *Closed*

- % Done changed from *80* to *100*

I am closing (rather than rejecting) this issue. Here is a summary:

- throwing an exception is not possible (because the problem is detected inside a dtor)
- the (humorous) "imminent disaster" message has been replaced by a more comprehensible one
- I have added a global flag GlobalManager::DtorFailed which is set if "imminent disaster" is detected
- the "imminent disaster" message is now printed by a fn which can be set as a breakpoint in a debugger

#### #9 - 28 Apr 2017 09:32 - Anna Maria Bigatti

- Estimated time set to *2.01 h*