

CoCoALib - Design #982

Catching interrupts in example progs?

25 Nov 2016 17:35 - John Abbott

<b>Status:</b>	Closed	<b>Start date:</b>	25 Nov 2016
<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>	1.99 hour
<b>Target version:</b>	CoCoALib-0.99700	<b>Spent time:</b>	1.95 hour
<b>Description</b>			
With the current design (where the exception produced by an interrupt is not of type CoCoA::ErrorInfo), if you interrupt an example program it produces a message about an Uncaught unknown exception.			
What do we think is the "right" behaviour?			
<b>Related issues:</b>			
Related to CoCoALib - Bug #971: CheckForInterrupt does not work in the expect...		<b>Closed</b>	<b>14 Nov 2016</b>
Related to CoCoALib - Feature #1388: GlobalManager: create SignalWatcher inte...		<b>New</b>	<b>09 Jan 2020</b>

History

#1 - 25 Nov 2016 17:44 - John Abbott

Perhaps the current behaviour is not so bad: CoCoALib converts any interrupts that it detects into an exception of type CoCoA::InterruptReceived (which is deliberately not derived from CoCoA::ErrorInfo). The examples then say that an "uncaught unknown exception" has reached main; while technically (almost) correct, this could be a little disconcerting for a novice.

A possibility would be to add a few lines to every example (and every test?) to catch CoCoA::InterruptReceived, and then print out a more "friendly" message. I have already placed some candidate lines in ex-empty.C, but they are currently commented out. It would take some time to add the lines to every example (and test). Perhaps this would be less worrying for a novice?

Another option is to catch all CoCoA::exception objects (i.e. both InterruptReceived and ErrorInfo) then print out a common message. With the current design ANNOUNCE works only for ErrorInfo; so some small change would be needed (either make ANNOUNCE work for both, or insert an if dynamic\_cast in the catch block to pick out the cases ANNOUNCE can handle).

Ideas? Comments? Other suggestions?

#2 - 25 Nov 2016 17:49 - John Abbott

- Related to Bug #971: CheckForInterrupt does not work in the expected way added

#3 - 05 Nov 2019 11:49 - John Abbott

- Status changed from New to In Progress
- Target version changed from CoCoALib-1.0 to CoCoALib-0.99700
- % Done changed from 0 to 10

There are two example progs for interrupts (ex-interrupt1.C and ex-interrupt2.C). I am not sure that they are so well written; perhaps the SignalWatcher should be **inside** the potentially long function?

Also reconsider the main topic of this issue; and hopefully resolve everything soon.

At the moment, it seems pointless to try to handle interrupts in very quick examples (e.g. less than 5 seconds run time?)

#### **#4 - 05 Nov 2019 14:26 - John Abbott**

On my machine all examples are faster than 5s, but several took more than 2s.

I have made some (local) changes to make some of the tests a bit faster; so now the only one which takes more than 2s is ex-hilbert1, so I have added the necessary lines to let it catch "interrupted exceptions".

I hope to check in soon.

#### **#5 - 14 Nov 2019 19:22 - John Abbott**

- Assignee set to John Abbott

- % Done changed from 10 to 50

I have modified ex-empty.C so that it creates a SignalWatcher immediately after GlobalManager, and also has a handler for InterruptReceived so that a sensible error message can be printed out.

If you don't want to do any signal handling, then you can just delete the relevant bits of code.

#### **#6 - 09 Jan 2020 13:42 - John Abbott**

- Status changed from In Progress to Resolved

- % Done changed from 50 to 70

- Estimated time set to 1.99 h

I now disagree with my comment 3 (above), and think that it is (usually) best to create the SignalWatcher just after the GlobalManager.

I think that the current situation for the examples is satisfactory (though perhaps not ideal).

#### **#7 - 09 Jan 2020 13:48 - John Abbott**

- Related to Feature #1388: GlobalManager: create SignalWatcher internally? added

#### **#8 - 09 Jan 2020 22:29 - John Abbott**

- Status changed from Resolved to Closed

- % Done changed from 70 to 100