

## CoCoALib - Design #1462

### Change CoCoA\_ERROR into CoCoA\_THROW\_ERROR

16 Jun 2020 16:55 - John Abbott

<b>Status:</b>	Closed	<b>Start date:</b>	16 Jun 2020
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	John Abbott	<b>% Done:</b>	100%
<b>Category:</b>	Improving	<b>Estimated time:</b>	4.71 hours
<b>Target version:</b>	CoCoALib-0.99800	<b>Spent time:</b>	4.65 hours
<b>Description</b>			
I suggest changing name of the macro to <b>CoCoA_THROW_ERROR</b> so that it is obvious to the reader that the error object is thrown.			
Also suggest creating new macro <b>CoCoA_THROW</b> which normally does a <b>throw</b> but will call <b>CoCoA::ThrowError</b> if debugging is active (CoCoA_DEBUG).			
Comments? Suggestions?			
<b>Related issues:</b>			
Related to CoCoALib - Feature #385: Design new errors using inheritance		<b>In Progress</b>	<b>08 Jul 2013</b>

#### History

##### #1 - 16 Jun 2020 16:55 - John Abbott

- Related to Feature #385: Design new errors using inheritance added

##### #2 - 16 Jun 2020 20:03 - John Abbott

- Description updated

- Status changed from New to In Progress

- % Done changed from 0 to 10

I have started. Defined the new macros CoCoA\_THROW\_ERROR (drop in replacement for current CoCoA\_ERROR). Also defined CoCoA\_THROW.

No testing yet.

##### #3 - 17 Jun 2020 20:08 - John Abbott

- Assignee set to John Abbott

- % Done changed from 10 to 70

I have changed all calls to CoCoA\_ERROR into CoCoA\_THROW\_ERROR; I do think the code is a bit more readable with THROW appearing explicitly.

All tests pass. Checked in.

##### #4 - 17 Jun 2020 21:02 - John Abbott

- Status changed from In Progress to Feedback

- % Done changed from 70 to 90

I have updated doc too. Also checked that everything works with debugging active.

#### #5 - 19 Jun 2020 20:27 - John Abbott

I have removed the macro **CoCoA\_THROW** for the following reasons:

- a macro is an ugly implementation trick
- it is not so important to avoid a single function call overhead when throwing an object (inherent overheads of throwing and stack unwinding will likely cost rather more)
- my first impl of `ThrowException` did not work as wanted: the type of the object being thrown is the *static* type when `throw` was called (but my design wanted the dynamic type)
- there is no great advantage to the user to call a macro `CoCoA_THROW` rather than a function `ThrowException`

I have anyway had to work around the 3rd issue: soln make `ThrowException` a template function.

All tests an example have passed with the current design. Must now update the doc (again! sigh!)

#### #6 - 26 Jun 2020 17:28 - John Abbott

Upon Anna's request I have put back `CoCoA_ERROR`, hopefully in such a way that it gives useful error mesgs if someone tries to call it.

With gcc on my linux box, the error mesg was OK but not super clear :-(  
Anna is testing now.

#### #7 - 14 Oct 2020 14:06 - John Abbott

- *Status changed from Feedback to Closed*
- *% Done changed from 90 to 100*
- *Estimated time set to 4.71 h*

Effectively removed old macro **CoCoA\_ERROR** (still exists, but triggers compile-time error saying to use `CoCoA_THROW_ERROR`).

New macro **CoCoA\_THROW\_ERROR** is drop in replacement for old/obsolete `CoCoA_ERROR`.