

## CoCoALib - Design #427

### Error names and error messages (current design)

28 Jan 2014 10:39 - Anna Maria Bigatti

<b>Status:</b>	In Progress	<b>Start date:</b>	28 Jan 2014
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	10%
<b>Category:</b>	Data Structures	<b>Estimated time:</b>	20.00 hours
<b>Target version:</b>	CoCoALib-1.0	<b>Spent time:</b>	2.70 hours
<b>Description</b>			
As a sub-issue of re-designing errors here we note the suggested names.			
<b>Related issues:</b>			
Related to CoCoALib - Feature #385: Design new errors using inheritance		<b>In Progress</b>	<b>08 Jul 2013</b>
Related to CoCoALib - Design #308: Error: ERR::NotNonZero instead of ERR::Zer...		<b>In Progress</b>	<b>12 Feb 2013</b>
Related to CoCoALib - Feature #721: CheckForInterrupt: string arg to specify ...		<b>Closed</b>	<b>29 May 2015</b>
Related to CoCoALib - Feature #743: Better errors: give supplementary info ab...		<b>In Progress</b>	<b>30 Jun 2015</b>
Related to CoCoALib - Feature #92: Error Codes		<b>In Progress</b>	<b>14 Feb 2012</b>
Related to CoCoALib - Design #582: Error codes: use same code for "not poly r...		<b>New</b>	<b>07 Jul 2014</b>
Related to CoCoALib - Design #1098: Ctors for exceptions/errors		<b>New</b>	<b>06 Sep 2017</b>

### History

#### #1 - 28 Jan 2014 13:59 - Anna Maria Bigatti

Many errors now are called **Not...** (e.g. **NotNonZero**) but that's ambiguous/confusing.  
Suggestion: rename them **Need...** (e.g. **NeedNonZero**)

Joke: does that mean that **NotYetImplemented** should become **NeedImplementing**? ;-)

#### #2 - 01 Apr 2014 17:37 - Anna Maria Bigatti

- Target version set to CoCoALib-0.99533 Easter14

#### #3 - 04 Apr 2014 17:12 - John Abbott

- Target version changed from CoCoALib-0.99533 Easter14 to CoCoALib-0.99534 Seoul14

#### #4 - 17 Jul 2014 14:34 - John Abbott

- Target version changed from CoCoALib-0.99534 Seoul14 to CoCoALib-1.0

#### #5 - 13 May 2015 09:57 - Redmine Admin

- Category set to Data Structures

#### #6 - 13 Apr 2017 17:19 - Anna Maria Bigatti

- Related to Feature #743: Better errors: give supplementary info about the error added

#### #7 - 19 Apr 2017 08:16 - Anna Maria Bigatti

Anna Maria Bigatti wrote:

Many errors now are called **Not...** (e.g. **NotNonZero**) but that's ambiguous/confusing.  
Suggestion: rename them **Need...** (e.g. **NeedNonZero**)

I'm quite keen to change the names (I'm confused myself in the meaning of the negation)  
**NeedNonZero** or **MustBeNonZero**?

**#8 - 19 Apr 2017 08:32 - Anna Maria Bigatti**

- Related to Feature #92: Error Codes added

**#9 - 19 Apr 2017 08:36 - Anna Maria Bigatti**

- Subject changed from Error names to Error names and error messages

- Status changed from New to In Progress

**#10 - 19 Apr 2017 08:39 - Anna Maria Bigatti**

In the error messages there are some "is not" and some "must be"

```
DEFINE_ERROR(NotPositive, "Value is not positive");  
DEFINE_ERROR(NotPositiveGrading, "Grading must be positive");
```

I prefer **must be** in the text.

Similarly, I prefer **MustBe** or **Need** in the name, instead of **Not**.

**#11 - 19 Apr 2017 09:29 - Anna Maria Bigatti**

- Related to Design #582: Error codes: use same code for "not poly ring" and "not elem poly ring" added

**#12 - 19 Apr 2017 09:42 - Anna Maria Bigatti**

- Subject changed from Error names and error messages to Error names and error messages (current design)

I think that, while thinking of the design using inheritance, it would help to simplify the current design. If we decide that the second argument of the error (the name of the function) could also give some extra indication, i.e. which argument is bad, than we could unify some errors. This is a step towards a classification.

I'll list here some proposals, and wait for John's approval.

**BadIndex, BadRowIndex, BadColIndex, BadDegIndex, BadComptIndex,...**

They are all **Index out of range**. I would make just **IndexOutOfRange**, and make sure the second argument of the error gives sufficient information to understand the case.

Indeed I'd rather have the two cases **NeedNonNegative/ArgTooBig**, but I prefer going in steps: this is a big change in the organization.

**#13 - 19 Apr 2017 09:42 - Anna Maria Bigatti**

- % Done changed from 0 to 10

- Estimated time set to 20.00 h

**#14 - 20 Apr 2017 09:32 - John Abbott**

My current thoughts about the design of errors is that CoCoALib errors will be represented as instances of very few distinct classes -- right now, I am thinking of just a **single class** (see [#385](#)).

If my current ideas come to fruition then essentially all details about the error will have to be in the string, but I would like the strings for different messages to be fairly uniform *e.g.* where appropriate we always write **must be positive**, rather than using other variants with much the same meaning (such as should be positive, expected positive, and so on).

I think it will be useful to look at the existing error messages, and determine "subphrases" which we should use. For instance, at the moment I believe I would like to keep the following subphrases distinct:

- out of range
- must be positive
- must be non-negative

They do all convey the idea of being "out of range", but I think it may be kinder to the user to have a "positive"/"non-negative" message -- because those messages give an indication about which range is valid!

#### #15 - 21 Apr 2017 17:57 - Anna Maria Bigatti

I simplified a few error messages and clarified error for PolyRingHom. Excerpt (

```
const std::string FnName = "PolyRingHom(Rx,S,CoeffHom,IndetImages)";
(...)
if (codomain(CoeffHom) != S)
    CoCoA_ERROR(ERR::BadCodomain, FnName + ": argument CoeffHom");
if (NumIndets(Rx) != len(IndetImages))
    CoCoA_ERROR(ERR::BadArraySize, FnName + ": arguments Rx, IndetImages");
```

This is a bit like what we have in CoCoA-5, where the faulty argument is underline.

I'm not changing the names now (good names are for developers, good messages are for the users! ... and the developers ;-). Anyway **BadArraySize** should be **IncompArraySize**.

#### #16 - 21 Apr 2017 17:59 - Anna Maria Bigatti

.... just to explain what I'm doing and why: I'm doing some other things, and making mistakes (... ehm... I'm making them on purpose ;-)  
So I'm fixing the error messages I do not like as they are, trying to make them more helpful.

#### #17 - 11 May 2017 10:57 - Anna Maria Bigatti

Just a trivial task (but may be annoying).  
Currently in error.H we comment an error name with its string: isn't this a pointless duplication of code?

#### #18 - 23 May 2019 10:53 - John Abbott

- Related to Design #1098: Ctors for exceptions/errors added