

## CoCoA-5 - Feature #1479

### CoCoA release for linux: CoCoAInterpreter: with and without libreadline?

07 Aug 2020 15:25 - John Abbott

<b>Status:</b>	Closed	<b>Start date:</b>	07 Aug 2020
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	John Abbott	<b>% Done:</b>	100%
<b>Category:</b>	Portability	<b>Estimated time:</b>	6.80 hours
<b>Target version:</b>	CoCoA-5.4.0	<b>Spent time:</b>	6.80 hours
<b>Description</b>			
I do recall some portability problems with libreadline. One possible solution is to mimic what we used to do for 32-bit and 64-bit versions: i.e. the distribution contains both, and the script tries first with readline, if that fails it tries without readline.  Is it worth it?			
<b>Related issues:</b>			
Related to CoCoA-5 - Design #1348: cocoa5 script for releases		<b>Closed</b>	<b>23 Oct 2019</b>
Related to CoCoALib - Design #933: Separate configure scripts for CoCoALib an...		<b>In Progress</b>	<b>30 Sep 2016</b>
Related to CoCoA-5 - Bug #1029: Readline: does not recognize interrupts		<b>Closed</b>	<b>14 Mar 2017</b>

#### History

##### #1 - 07 Aug 2020 15:25 - John Abbott

- Related to Design #1348: cocoa5 script for releases added

##### #2 - 09 Oct 2020 15:42 - John Abbott

- Description updated

##### #3 - 09 Oct 2020 15:48 - Anna Maria Bigatti

- Subject changed from CoCoAInterpreter: with and without libreadline? to CoCoA release for linux: CoCoAInterpreter: with and without libreadline?

- Status changed from New to In Progress

- % Done changed from 0 to 10

Decision: distribution with two executables.

John does the cocoa-5 script

Anna does the release script (and the releases)

##### #4 - 10 Oct 2020 23:15 - John Abbott

- Subject changed from CoCoA release for linux: CoCoAInterpreter: with and without libreadline? to CoCoAInterpreter: with and without libreadline?

An alternative solution is to ask the user to use an install script: this script could then determine which executable works on the current platform, and then "install" just that one (here "install" could probably mean "make a symbolic link (from CoCoAInterpreter to the actual version which works)).

A disadvantage of checking which version to use each CoCoA is started is that it could become "costly" to start cocoa:

(1) try executable 1 with no packages and empty input (to see if dynamic linking works)

(2) ditto for executable 2...

(3) finally start the working executable properly (i.e. with packages & any user specified input files).

This "check every time" approach might have some advantages: e.g if a missing dynamic library is subsequently installed...

I'm not sure how large the overhead might be: 2 or 3 executables must be started *each time* CoCoA is used. I do recall one user (Migliore?) wanting to use CoCoA to make thousands of quick, simple computations, starting a new session for each one.

**#5 - 10 Oct 2020 23:17 - John Abbott**

- Subject changed from *CoCoAInterpreter: with and without libreadline?* to *CoCoA release for linux: CoCoAInterpreter: with and without libreadline?*

**#6 - 11 Oct 2020 22:10 - John Abbott**

I did a quick speed test: specifying as input /dev/null, and where the directory NoPackages contains only init.cocoa5 (which is empty)  
100 times `./CoCoAInterpreter --no-preamble --packageDir NoPackages` took about 0.32s (real) of which 0.22s (user)  
100 times `./CooAInterpreter --no-preamble` took about 7.5s (real) of which 6.9s (user)

Conclusion: starting CoCoAInterpreter with no packages has a negligible cost compared to starting it with the packages.

**NOTE** initially I made the mistake of starting CoCoA by calling the script cocoa5; that incurred a significant overhead (e.g. the "no packages" version took about twice as much time)

**#7 - 12 Oct 2020 14:16 - John Abbott**

- Assignee set to John Abbott

- % Done changed from 10 to 20

It is not entirely clear to me how to make `--packageDir NoPackages` work portably.  
One solution is to make NoPackages an absolute path, but that means the script has to be changed if moved to elsewhere.  
If it is a relative path, then where is it relative to? The dir containing the cocoa5 script, or the "current directory" when cocoa5 was called?

I now think it is probably cleaner/neater to have a new command line option, say `--do-nothing`, which simply tells CoCoAInterpreter to exit (without trying to read packages or init files). If we do this, what should the option be called?

If `--do-nothing` is specified then no other args are allowed.

**UPDATE:** I have made a first impl of `--do-nothing`; it is faster than `--packageDir NoPackages`

**#8 - 12 Oct 2020 20:14 - John Abbott**

- % Done changed from 20 to 30

I have made a first attempt at modifying the start-up script. It took me a lot longer than anticipated :-)

It is hard to produce helpful error mesg's given that so many things could go wrong... :-/

**#9 - 14 Oct 2020 22:07 - John Abbott**

I have checked in the minor change to Main.C which impl the `--do-nothing` option.

**#10 - 20 Oct 2020 11:58 - John Abbott**

- % Done changed from 30 to 60

I now have a new cocoa5 script for GNU/linux. I have developed it "blind", so it probably does not work correctly (yet).

I have also realised a disadvantage of specifying readline in the configuration for cocoa5: to make two versions of CoCoAInterpreter with/without readline, we are forced to recompile cocoa5 too (even though this is unnecessary). I believe the correct solution is to have two separate

configuration procedures for cocoa5 (see [#933](#)).

**#11 - 20 Oct 2020 11:58 - John Abbott**

- *Related to Design #933: Separate configure scripts for CoCoALib and CoCoA-5 added*

**#12 - 20 Oct 2020 15:33 - John Abbott**

- *% Done changed from 60 to 70*

I have tried testing with the current public version of CoCoA-5 for linux: it fails because the interpreter does not recognise --do-nothing. Anyway, the error mesgs now seem reasonable. Hope to check in soon.

**#13 - 20 Oct 2020 17:12 - John Abbott**

I have checked in my prototype... hope it is a good one!

**#14 - 03 Nov 2020 22:11 - John Abbott**

- *Status changed from In Progress to Feedback*

- *% Done changed from 70 to 90*

Here is a little doc (where should it really go?)

The new script expects one of two possibilities for the bin/ directory:

- there is just 1 file called CoCoAInterpreter (which must be executable)
- there are exactly 2 files, called CoCoAInterpreter-with-readline and CoCoAInterpreter-without-readline

The script checks how many files are in bin/: error if fewer than 1 or more than 2. Then it checks that one of the two expected situations is actually present (o/w error). If there are two files, it tries first **with** readline and if that does not start then it tries without.

I wonder if we will test it with the imminent interim release?

**#15 - 04 Nov 2020 12:06 - John Abbott**

Currently the script reports a very succinct error message if the executables cannot run (probably due to shared library problems). Should I change the script so that in such cases, it also prints out whatever error messages are produced when attempting to run the executable? Then we should also ask the user to send a copy of the entire error message to the CoCoA Team... I wonder how long such error mesgs can be?

**#16 - 22 Jan 2021 09:35 - Anna Maria Bigatti**

- *Target version changed from CoCoA-5.4.2 to CoCoA-5.4.0*

**#17 - 16 Feb 2022 20:03 - John Abbott**

- *Related to Bug #1029: Readline: does not recognize interrupts added*

**#18 - 16 Feb 2022 20:07 - John Abbott**

- *Status changed from Feedback to Closed*

- *% Done changed from 90 to 100*

- *Estimated time set to 6.80 h*

Closing after 1 year in feedback.

The only thing left is to remember to make 2 executables (if we want to do that).