CoCoALib - Feature #1395

SHA checksum for released code

17 Jan 2020 15:24 - John Abbott

Status:	Closed	**Start date:**	17 Jan 2020
Priority:	Normal	**Due date:**	
Assignee:	John Abbott	**% Done:**	100%
Category:	Safety	**Estimated time:**	5.01 hours
Target version:	CoCoALib-0.99800	**Spent time:**	4.90 hours

Description

Winfried Bruns requests that we make available also an SHA-256 checksum for code that we distribute.

This seems to be a reasonable request. How should we implement it?

Related issues:

Related to CoCoA-5 - Support #1387: John's visit Feb 2020	**Closed**	**07 Jan 2020**

History

#1 - 17 Jan 2020 15:32 - John Abbott

- Status changed from New to In Progress

- % Done changed from 0 to 10

Original request came from Winfried Bruns (via email on 2020-01-17).

The purpose of the SHA-256 checksum is to provide a "good guarantee" that the distributed file (*e.g.* cocoalib-XYZ.tgz) is uncorrupted.
While SHA-256 is not 100% safe, it is surely good enough for our purposes; and programs to generate and check SHA-256 sums should be widely available.

It seems that the relevant command is **shasum -a 256**. This is available on my GNU/Linux "Ubuntu" box, and reportedly available also on MacOS X.

Checking the checksum is done by a call like **shasum -a 256 -c SUM_FILE**.

The checksums should be made cut-and-pastable from the relevant download web pages. I hope this will not be too tricky.

#2 - 17 Jan 2020 15:32 - John Abbott

- Related to Support #1387: John's visit Feb 2020 added

#3 - 17 Jan 2020 15:43 - John Abbott

To release source code we should do the following:

1. generate TARGZ file (maybe also TAR.XZ which is usually smaller)
2. run **sha -a 256 TARGZFILE** and store the resulting output
3. upload TARGZ file to server, and revise links on the website
4. upload the CHECKSUM, and put it on the website (the checksum should be easy to download/copy-and-paste)

Here is what the BOOST people do:

```
https://www.boost.org/users/history/version_1_72_0.html
```

We should probably also produce SHA-256 checksums for the compiled versions too -- use **shasum -a 256 -b** for binary file!!

#4 - 29 Jan 2020 12:07 - John Abbott

I have just stumbled across "GPG" signatures for software. There is a description at https://www.gnupg.org/gph/en/manual/x135.html

I do not know what the relative pros and cons of GPG compared to SHA are...

PS I encountered the GPG signatures on the website https://julialang.org/downloads/

#5 - 31 Jan 2020 10:29 - John Abbott

I suppose the GPG signatures give a better guarantee of correctness.

Probably we should try both ways, and see how easy it is for a user to do the check (and how easy it is for us to generate the "signature").

#6 - 04 Feb 2020 13:48 - John Abbott

Florian has reported a problem with our being "naughty" about making a new release without changing the version number.

Apparently there is an *Arch Linux* package for CoCoALib (and maybe CoCoA?) which maintains a link to our download page, but which also includes an SHA checksum of the file (to guard against corrupted/hacked TGZ files).

Currently the Arch Linux package system reports CoCoA as being hacked because the SHA checksum was calculated with one version of the TGZ file, but then we changed the file without changing the release number.

#7 - 13 Feb 2020 14:08 - John Abbott

Anna confirms that she has shasum on her computer.

The main question remaining is how to use the SHA sum in a sensible way (*i.e.* such that it gives some assurance of integrity).

#8 - 15 Feb 2020 10:42 - John Abbott

- % Done changed from 10 to 30

~~A fairly simple approach would be to put the SHA checksum in the release notes~~ (perhaps near the start?) **<--- BAD IDEA**
The checksum should be cut-and-pastable (that's all, I think).
We can try this, and see if it is enough for Winfried.

#9 - 17 Feb 2020 12:29 - John Abbott

My idea in comment 8 is not so clever... well, the SHA cannot be inside the TGZ file.
Just put it on the website in a place where it is perfectly clear to which file it refers.
It is also possible to make the SHA-256 checksum available as a separate download; but if this is on the same server as the TGZ file (very likely) then this does not really offer any extra security, since if a hacker can change the TGZ file, then it should be easy to change the SHA file too.

#10 - 06 Mar 2020 16:19 - John Abbott

If we want to do this, we should delay the release a few days, so we can find out what to do.

#11 - 09 Mar 2020 15:26 - John Abbott

- % Done changed from 30 to 50

So far I have made a small change to release-linux.sh.
I have added a line after the MakeTGZ line:

```
shasum -a 256 "$COCOA_TEXT-linux.tgz"  >  "$COCOA_TEXT-linux.SHA"
```

The extra line produces a file with a name like **cocoa-5.3-linux.SHA**.
The file contains 1 line, in this case

```
63863371d6f16d2ef7a6542b8626d5d905b905e394a95df4c75366d0b90ec7b4  cocoa-5.3-linux.tgz
```

We can copy the hexadecimal part to the download page, somewhere near the link to download the TGZ file.
Or we can also make the SHA file downloadable (from a link next to the TGZ link).

#12 - 09 Mar 2020 16:31 - John Abbott

- Status changed from In Progress to Resolved

- Assignee set to John Abbott

- % Done changed from 50 to 80

I have also added a quick check that there are no writable files in the subtree to be released.

I have modified release-mac.sh too.

#13 - 11 Mar 2020 12:26 - John Abbott

I think I might be slightly confused. What I wrote above in comments 11 and 12 actually refers to the "binary" release.

Do we have a script that makes the source TGZ release?

#14 - 11 Mar 2020 17:42 - Anna Maria Bigatti

adjusted the page and the release-xxx.sh file to produce the correct files and instructions for publishing.

#15 - 20 Mar 2020 13:04 - John Abbott

I suggest putting the SHA-256 checksum in the "Notes" section on the web page for source releases.
Probably the best place is just before the usual release notes: those who don't care about it, can easily skip over it; those who do care, will see the checksum immediately.

#16 - 20 May 2020 13:33 - John Abbott

- Status changed from Resolved to Feedback

- % Done changed from 80 to 90

- Estimated time set to 4.77 h

I see that the SHA cecksum is in the "version" column; I think it might be more readable in the "notes" column.
Is that relatively easy to achieve?

Perhaps I'll ask Bruns for his opinion.

#17 - 20 May 2020 14:17 - Anna Maria Bigatti

John Abbott wrote:

> I see that the SHA cecksum is in the "version" column; I think it might be more readable in the "notes" column.
> Is that relatively easy to achieve?

easy to achieve, but difficult to maintain ;-)
I prefer to leave it there, I know the line is long, but it also make more sense closer to the file it refers to

> Perhaps I'll ask Bruns for his opinion.

#18 - 09 Oct 2020 14:12 - Anna Maria Bigatti

- Status changed from Feedback to Closed

- Target version changed from CoCoALib-0.99700 to CoCoALib-0.99800

- % Done changed from 90 to 100

- Estimated time changed from 4.77 h to 5.01 h