

OpenNebula - Backlog #3853

minified javascripts

06/27/2015 12:41 PM - Dmitry Smirnov

Status:	Closed	Start date:	06/27/2015
Priority:	Normal	Due date:	
Assignee:	Jaime Melis	% Done:	0%
Category:	Packaging	Estimated time:	0.00 hour
Target version:	Release 5.4		
Description			
<p>There are many minified javascripts committed to source tree -- `*.min.js` as well as</p> <ul style="list-style-type: none">- src/sunstone/public/bower_components/datatables/media/js/jquery.js- src/sunstone/public/bower_components/foundation/js/vendor/jquery.js- src/sunstone/public/bower_components/foundation/js/vendor/modernizr.js- src/sunstone/public/bower_components/foundation/js/vendor/placeholder.js <p>and others including some minified CSS files.</p> <p>Minified files are effectively the same as pre-compiled binaries -- un-readable, non-modifiable and non-distributable in Debian.</p> <p>It is the best (practice) to have no minified files in source tree at all.</p> <p>Please replace all minified javascripts with their original uncompressed versions.</p>			
Related issues:			
Duplicated by Bug # 3891: non-free file "WebSocketMain.swf"		Closed	07/27/2015

Associated revisions

Revision 58c7d859 - 07/17/2015 01:49 PM - Daniel Molina

bug #3853: Remove external js files, user bower to install them

Revision cb24ac8a - 07/17/2015 01:49 PM - Daniel Molina

bug #3853: Remove minified files, generate them using grunt instead

Revision f7defd07 - 07/17/2015 01:49 PM - Daniel Molina

bug #3853: Update dependency versions in bower.json and package.json

Revision a5eb40f7 - 07/17/2015 01:49 PM - Daniel Molina

bug #3853: Add -p option in install.sh to install only minified files

Revision c34113de - 07/17/2015 01:49 PM - Daniel Molina

bug #3853: Add sunstone option to scones to compile css and js

Revision 0f93cea2 - 07/17/2015 02:07 PM - Daniel Molina

bug #3853: Add README on how to install sunstone deps and compile js and css

Revision c83ec494 - 07/17/2015 02:44 PM - Daniel Molina

bug #3853: Add -p option when only sunstone is installed

Revision 03b17ede - 05/20/2016 01:47 PM - Javi Fontan

backlog #3853: crypto-js is not needed anymore

History

#1 - 07/10/2015 07:46 AM - EOLE Team

+1

It's fine to generate the minified version during the build process.

There is a [Debian Javascript Policy Draft](#) about it.

Regards.

#2 - 07/17/2015 03:06 PM - Daniel Molina

Hi,

We have removed all the minified and external js files. You can find a detailed explanation in the following link, if this is not enough or you have any doubts please let me know:

<https://github.com/OpenNebula/one/blob/master/src/sunstone/public/README.md>

#3 - 07/17/2015 05:09 PM - Dmitry Smirnov

Fantastic, thank you. Without minified files it should be much easier to work on Debian package.

As for external files I hope their uncompressed versions will be present in source tarball.

Remember that tarball should be comprehensive but without pre-built files.

I'll provide some feedback once those changes make it into next release (hopefully).

#4 - 07/17/2015 05:12 PM - Dmitry Smirnov

Just to make it clear that when Debian package is built it happens in clean environment without internet access so nothing can be downloaded on build time.

That's why source tarball should be comprehensive or declare dependencies satisfiable from native repository.

#5 - 07/28/2015 02:09 AM - Dmitry Smirnov

Daniel, I had a quick look at the changes and frankly I'm terrified...

- Because essential requirements were dropped it is no longer possible to build Opennebula from release tarball downloaded from GitHub.
A special script/procedure will be required to generate comprehensive release tarball and upload it somewhere. It complicates release process.
- Even worse, now it is impossible to build Opennebula from source in restricted environment (i.e. no internet access).
Consider the following scenario: I download release tarball from GitHub; copy tarball to restricted server; install *packaged* dependencies but can't build Opennebula anyway because build system needs to download components directly from Internet.
- What bothers me most is that such change makes it impossible to build Debian packages
so I'm afraid that instead of updating Opennebula in Debian we will be forced to remove it entirely. :(
- Further security concerns are in the possibility to have different binaries made from the same sources because something may change in the downloadable components between two builds. Therefore there is no guarantee that build will be reproducible.
- In addition to `-p`` option it would be nice to have an option to install only uncompressed files (no minification).

Please remember that *comprehensiveness of tarball is very important*.

Please restore essential external (i.e. 3rd party) files but ensure FLOSS compliance by including all sources (there shall be no pre-built source-less files).

Thank you.

P.S. Could someone upgrade my access so I could change bug's priority please?

#6 - 07/28/2015 08:51 AM - Daniel Molina

Dmitry Smirnov wrote:

Daniel, I had a quick look at the changes and frankly I'm terrified...

- *Because essential requirements were dropped it is no longer possible to build Opennebula from release tarball downloaded from GitHub.
A special script/procedure will be required to generate comprehensive release tarball and upload it somewhere. It complicates release process.*
- *Even worse, now it is impossible to build Opennebula from source in restricted environment (i.e. no internet access).
Consider the following scenario: I download release tarball from GitHub; copy tarball to restricted server; install packaged dependencies but can't build Opennebula anyway because build system needs to download components directly from Internet.*
- *What bothers me most is that such change makes it impossible to build Debian packages
so I'm afraid that instead of updating Opennebula in Debian we will be forced to remove it entirely. :(*

The tarball generated for each release will include the 3rd party libraries with all the files provided by running `bower instal <package_name>`. Therefore, you won't need internet access to build the minified files. But these libraries will be only included for each release when the tarball is generated but not in the repository.

- If we remove the .min and binary files from the tarball, would that be debian compliant?
- How do you deal with ruby gems that are not provided as a debian package?
 - *Further security concerns are in the possibility to have different binaries made from the same sources because something may change in the downloadable components between two builds. Therefore there is no guarantee that build will be reproducible.*

The bower.json file uses fixed version for each of the libraries, hence the same minified files are always generated for the same release.

- In addition to `-p` option it would be nice to have an option to install only uncompressed files (no minification).

If you don't provide the `-p` option a `cp -R` is done on the sunstone folder, therefore if the minified files are not generated, they will not be copied. Is that enough?

Cheers

#7 - 07/28/2015 08:53 AM - Daniel Molina

- Duplicated by Bug #3891: non-free file "WebSocketMain.swf" added

#8 - 07/28/2015 10:10 AM - Dmitry Smirnov

The tarball generated for each release will include the 3rd party libraries with all the files provided by running `bower instal <package_name>`. Therefore, you won't need internet access to build the minified files. But these libraries will be only included for each release when the tarball is generated but not in the repository.

Excellent. That is exactly what we need. :)

If we remove the `.min` and binary files from the tarball, would that be debian compliant?

Probably it would be compliant unless we forget about something else like #3892, etc.

How do you deal with ruby gems that are not provided as a debian package?

Easy. :) I just package 'em as standalone library. For example here is new opennebula-flow dependency that was recently uploaded for review:

- <https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=793617>
- https://ftp-master.debian.org/new/ruby-parse-cron_0.1.4-1.html

The bower.json file uses fixed version for each of the libraries, hence the same minified files are always generated for the same release.

Assuming there are no changes upstream, man-in-the-middle attacks, stable internet connection, all web sites are up and serving downloads, etc. Something not to be taken as granted...

If you don't provide the `-p` option a `cp -R` is done on the sunstone folder, therefore if the minified files are not generated, they will not be copied. Is that enough?

Yes, that will be perfect. Thank you.

#9 - 07/31/2015 06:29 PM - Dmitry Smirnov

Debian Upstream Guide [1] answers many questions regarding best practices for release tarballs, bundled libraries, pre-built binaries and other issues.

[1]: <https://wiki.debian.org/UpstreamGuide>

#10 - 08/03/2015 12:19 PM - Dmitry Smirnov

I've just tried to build Debian package from ``https://github.com/OpenNebula/one/archive/release-4.14-beta1.tar.gz`` but just as I was afraid of, there are no essential files included to the archive so building Debian package from it is nearly impossible. Re-including removed files seems to be the only way to fix this situation.

I am very concerned about this problem. Broken release tarball will not help anything...
How are you going to generate comprehensive release tarball and where it is going to be published?
We do not want to lose GitHub as a mirror or host for OpenNebula releases...

Although it is time consuming, we can drop non-DFSG files from release tarball but we can not re-insert missing ones...

This is the most outstanding problem in packaging of OpenNebula-4.14...

#11 - 08/04/2015 09:41 AM - Ruben S. Montero

- *Category set to Packaging*
- *Target version set to Release 4.14*

You are right those files should be in the package... thanks for the heads up

#12 - 08/31/2015 10:11 AM - Ignacio M. Llorente

- *Assignee set to Jaime Melis*

#13 - 08/31/2015 11:45 AM - Dmitry Smirnov

FYI I wrote a short summary about minification: <https://wiki.debian.org/onlyjob/no-minification>
I hope you might find it useful.

#14 - 09/03/2015 08:25 AM - Jaime Melis

Hi Dmitry. We've been discussing this issue, these are our conclusions.

First thing, due to a release process mistake, the tarball that was made publicly available was not the correct one, it didn't include the files we wanted it to include. Please take a look at these two files:

- <http://downloads.opennebula.org/packages/.testing/opennebula-4.13.80/opennebula-4.13.80.tar.gz>
- <http://downloads.opennebula.org/packages/.testing/opennebula-4.13.80/Debian-8/Debian-8-opennebula-4.13.80-1.tar.gz> (it includes the sources)

When we build the tarball (the 'good' one, the one reference above) we do this:

```
git clone -q https://github.com/OpenNebula/$GITREPO $TMPDIR/one

(
  cd $TMPDIR/one
  REF=$COMMIT
  [ -z "$REF" ] && REF=HEAD
```

```

git archive --format=tar --prefix=$PACKAGE/ $REF | (cd $TARDIR; tar xf -)

# sunstone
cd $TARDIR/$PACKAGE/src/sunstone/public
npm install
bower install --config.interactive=false
grunt sass
grunt requirejs

cd $TARDIR
fakeroot tar czf $PACKAGE.tar.gz $PACKAGE
)

```

As you can see, instead of simply doing a git-archive, which is what we used to do for OpenNebula <= 4.12, we now additionally do:

- npm install => downloads node_modules
- bower install => downloads bower_components
- grunt sass & grunt requirejs => compile the minified css and js

We don't need to ship node_modules or bower_components, so we intend to add:

```
rm -rf bower_components node_modules
```

In order to clean those files from the tarball.

Why we are doing all of this? mainly because we are building the package with pbuilder, and grunt is not packaged for debian (or ubuntu), so we need to install it before hand. We could run npm install and bower install inside the pbuilder chroot, and then do the grunt sass, so the tarball is not tarnished with these binary files, but we believe that is much more complicated than adding the files to the tarball.

So, ideally we would like to have:

- a pristine tarball
- build the minified css and js inside the pbuilder environment

But we can't since we require 'grunt' to be packaged and available in all the distros we support.

So, our question is: since you may not have internet connection inside your building environment, how are you planning on building the minified the css and the js? The only solution, given that we don't have a grunt package, is to do what we did, that is, include the minified css and js into the tarball.

#15 - 09/03/2015 09:42 PM - Dmitry Smirnov

So, our question is: since you may not have internet connection inside your building environment, how are you planning on building the minified the css and the js?

I'm not planning to minify javascripts and CSS. What problem are you trying to solve by minification?

I would understand if minification would be justified on popular and busy web site when many new users congest bandwidth.

But OpenNebula is clearly not such case -- please remember that only first request to pre-minified resource can have little benefit from reduced size of files. Subsequent requests will be satisfied from browser cache.

I imagine that typical pattern of Sunstone use would be several usual users repeatedly accessing web interface hence there are no benefits of minification whatsoever. You are optimising for use case that OpenNebula never exhibit. And the cost of maintenance is significant not to mention all

the cons.

| *The only solution, given that we don't have a grunt package, is to do what we did, that is, include the minified css and js into the tarball.*

You do not need Grunt for minification. There are two compressors available in Debian (I don't remember their names) so minification can be done by few lines of shell code (probably by using `find` command with file name pattern).

We have to remove all pre-minified files from tarball because we can't distribute 'em and because one should not trust pre-built binaries. Please include no minified files to release tarball.

#16 - 09/04/2015 12:21 PM - Jaime Melis

Understood. We agree and we will make the necessary modifications so we can run Sunstone with minimized and compiled code, and without it. However, we don't have the time to make such modifications for 4.14. Would it be possible for you to use temporarily the minified and compiled code for this release, and for the next one use the plain files?

Please check the tarball I mentioned in my last comment, it should have all the necessary files for building there, without requiring an internet connection.

One last thing, compilation is needed, but minification is not, although it's there by default. You can disable the minification by patching this file:

```
a/src/sunstone/public/Gruntfile.js b/src/sunstone/public/Gruntfile.js
index 93dbb05..bf88ce6 100644
--- a/src/sunstone/public/Gruntfile.js
+++ b/src/sunstone/public/Gruntfile.js
@@ -48,7 +48,7 @@ module.exports = function(grunt) {
    //name: 'vendor/almond', // 3
    mainConfigFile: './app/main.js', // 5
    preserveLicenseComments: false,
-   optimize: 'uglify2',
+   optimize: 'none',
    generateSourceMaps: true,
    removeCombined: true,
    //skipDirOptimize: false,
```

#17 - 09/04/2015 04:20 PM - Dmitry Smirnov

Thank you. Fair enough, I am happy with any progress as long as you are working towards resolution of this issue. :)
Plain files in release following 4.14 sounds good enough.

In the official Debian package I've already made an effort to replace all minified files.
Certainly next release can be packaged in a similar manner.

I could not check new tarball yet and I have no time frame regarding when I'll be able to do that...

I understand that grunt compilation should be done before generating release tarball because we will have to build from tarball without grunt and without internet connection. Maybe use of grunt can be avoided entirely? Please excuse me if this is a silly idea -- I'm not familiar with grunt so I'm not sure how important it is for OpenNebula.

#18 - 09/14/2015 02:29 PM - Jaime Melis

- *Tracker changed from Bug to Backlog*
- *Target version changed from Release 4.14 to Release 5.0*

Moving this issue to backlog, so we can reach out before 5.0

#19 - 01/04/2016 01:08 PM - Ruben S. Montero

- *Tracker changed from Backlog to Feature*

#20 - 01/04/2016 01:08 PM - Ruben S. Montero

- *Status changed from Pending to New*

#21 - 05/03/2016 10:34 AM - Ruben S. Montero

- *Tracker changed from Feature to Backlog*

#22 - 05/20/2016 01:30 PM - Javi Fontan

The tarball for 5.0 Beta comes with both minimized and unminimized versions of all libraries except for these:

- `src/sunstone/public/vendor/crypto-js`: We can add the unminimized files there
- `src/sunstone/public/bower_components/vis/docs/js`: These are examples not used by OpenNebula

Adding those unminimized files is enough?

The tarball can be found here: <http://downloads.opennebula.org/packages/opennebula-4.90.0/opennebula-4.90.0.tar.gz>

To search for minimized files without original counterparts I've used this:

```
for f in $(find . | grep -E 'min\.js$'); do ls $(echo $f | sed 's/\.min\.js/.js/'); done
```

#23 - 05/20/2016 01:55 PM - Javi Fontan

crypto-js libraries are not needed anymore. Removed.

#24 - 09/13/2016 10:45 AM - Ruben S. Montero

- *Target version changed from Release 5.0 to Release 5.4*

#25 - 07/12/2017 04:44 PM - Javi Fontan

- *Status changed from New to Closed*