

Ruby - Feature #15171

[patch] Use OpenSSL 1.1.1 for Appveyor MinGW build

09/27/2018 08:36 PM - MSP-Greg (Greg L)

Status:	Closed	
Priority:	Normal	
Assignee:		
Target version:		
Description @nobu (Nobuyoshi Nakada) Thanks for your work with the Appveyor testing. It took three revisions, but MSYS2 has jumped from OpenSSL 1.0.2 to 1.1.1. The package release 1.1.1-4 should build fine on Appveyor MinGW. I don't think it has hit the MSYS2 repository yet, but certainly in the next 24 hours. If you've got MSYS2 installed, pacman -S mingw-w64-x86_64-openssl should show availability. But you knew that. Hence, line 122 of the current Appveyor.yml file would need mingw-w64-x86_64-openssl added to it. At present, Appveyor has got 1.0.2p. Lastly, given any thought to moving the VC builds to Azure? I may see if I can make both VC12 and VC14 OpenSSL packages. Seems inappropriate to load different runtimes with (I think) VC12. Greg		

History

#1 - 09/27/2018 08:37 PM - MSP-Greg (Greg L)

- Subject changed from *OpenSSL soon available from MSYS2* to *OpenSSL 1.1.1 soon available from MSYS2*

#2 - 09/27/2018 09:09 PM - naruse (Yui NARUSE)

MSP-Greg (Greg L) wrote:

Lastly, given any thought to moving the VC builds to Azure? I may see if I can make both VC12 and VC14 OpenSSL packages. Seems inappropriate to load different runtimes with (I think) VC12.

It is bad if VC14 build uses OpenSSL libraries built with VC12.
But appveyor build is useful because it runs every commit.

Note that we already have msvc12 CI on rubyci.org.
(but adding msvc14 is difficult with our current budget)

I'm interested in Azure Pipelines but not familiar with it.

#3 - 09/27/2018 09:40 PM - MSP-Greg (Greg L)

[@naruse \(Yui NARUSE\)](#)

I guess I was referring to the two VC builds on Appveyor. With three serial builds on Appveyor, the CI can get quite backed up. Sometimes I've started a new ruby-loco build just to check...

I'm probably mistaken but I think OpenSSL package that Appveyor has (Shining Light Productions? slproweb.com) is built with VC14. Regardless, with one OpenSSL package and two VC builds, one isn't matched up.

Re Azure, it was my impression that it allowed parallel builds if one had more than one 'job', although, I believe like Appveyor, they're using two core VM's. Also, they don't (currently) have things like OpenSSL, 7Zip, MSYS2, etc. The parallel builds are available free for public OSS GitHub repos.

A while ago I had a Ruby VC14 build running, and I created a VC14 OpenSSL build for it (OpenSSL CI's on Appveyor with vc14). I'll see if I can get OpenSSL 1.1.1 to build with both VC12 & VC14, and then try Azure.

Greg

#4 - 10/09/2018 03:00 PM - jaruga (Jun Aruga)

The package release 1.1.1-4 should build fine on Appveyor MinGW.

Thank you for the working.

The OpenSSL 1.1.1 on Appveyor is a binary package built in advance right?

Not building from source code on Appveyor?

#5 - 10/09/2018 03:35 PM - MSP-Greg (Greg L)

[@jaruga \(Jun Aruga\)](#)

I'm not sure what you mean. Whenever I refer to a package, I'm referring to a pre-compiled package. MSYS2/MinGW has added CI on Azure pipelines, so I'm not sure if they build there or on Appveyor.

Appveyor's packages are not always current, and as of this morning 2018-10-09, OpenSSL 1.0.2.p is installed. The MYS2/MinGW 'system' currently has 1.1.1 available. So, one can update the package in script using pacman, just like one would for updating any other package.

#6 - 10/09/2018 05:36 PM - jaruga (Jun Aruga)

I'm not sure what you mean. Whenever I refer to a package, I'm referring to a pre-compiled package. MSYS2/MinGW has added CI on Azure pipelines, so I'm not sure if they build there or on Appveyor.

I mean that I wanted to know if the OpenSSL package has already compiled or not.

So, I could know it is pre-compiled. That's what I wanted to know.

Because on the ticket <https://bugs.ruby-lang.org/issues/15220>, I am working for this ticket's Linux CI version. I had to compile it from source code. And it took 2 min 14 sec totally.

#7 - 10/12/2018 08:49 PM - MSP-Greg (Greg L)

jaruga (Jun Aruga) wrote:

I mean that I wanted to know if the OpenSSL package has already compiled or not.

Yes, it is compiled. It just needs to be added to the other packages that are checked for updates, currently line 124 of appveyor.yml.

Just my opinion, but it might be helpful to have the following info logged in the Travis & Appveyor logs:

```
ruby -ropenssl -e "puts OpenSSL::OPENSSL_VERSION, OpenSSL::OPENSSL_LIBRARY_VERSION"
```

Sorry for the delay, and thanks, Greg

#8 - 10/14/2018 11:16 AM - jaruga (Jun Aruga)

Just my opinion, but it might be helpful to have the following info logged in the Travis & Appveyor logs:

```
ruby -ropenssl -e "puts OpenSSL::OPENSSL_VERSION, OpenSSL::OPENSSL_LIBRARY_VERSION"
```

That's a good idea!

#9 - 10/14/2018 02:35 PM - MSP-Greg (Greg L)

[@jaruga \(Jun Aruga\)](#)

I'm wondering if more info would be helpful, especially when people may be building/testing Ruby on other CI platforms.

An example of what I run along with the ruby-loco CRON jobs is here:

<https://ci.appveyor.com/project/MSP-Greg/appveyor-ruby>

At the bottom is the info output for the current trunk build. It's more than is needed, but, in addition to OpenSSL, there are packages that have version info that might be helpful when comparing builds...

#10 - 10/15/2018 11:47 AM - jaruga (Jun Aruga)

At the bottom is the info output for the current trunk build. It's more than is needed, but, in addition to OpenSSL, there are packages that have version info that might be helpful when comparing builds...

I saw the ruby-loco's log now.
I am not sure how much the additional version info is useful.

Because we can check default gems' version from ext/bigdecimal/bigdecimal.gemspec, ext/json/json.gemspec and etc, and bundled gems' version from gems/bundled_gems.

In the case of openssl, as we might want to see actual built openssl's version, I thought above command was useful.

#11 - 10/18/2018 08:51 PM - MSP-Greg (Greg L)

- File `av_openssl_1.1.1.patch` added

- Subject changed from `OpenSSL 1.1.1 soon available from MSYS2 to [patch] Use OpenSSL 1.1.1 for Appveyor MinGW build`

Attached is patch file that updates Appveyor.yml to:

1. Update to the current MSYS2/MinGW OpenSSL package for the mingw build (as of 2018-Oct, 1.1.1)
2. Outputs OpenSSL version info at end of each build

The patch passed on my fork, see
<https://ci.appveyor.com/project/MSP-Greg/ruby/builds/19619466>

Notes:

3. The current code for the mingw build does not update all MSYS2 packages. Due to some file casing issues, the old OpenSSL package must be removed prior to installing the new one.

1. At present, Appveyor has the OpenSSL-Win64 vc software installed. The dll's for it are incorrectly placed in the System32 directory. They need to be renamed before running (testing) the mingw build. I've had the same issue with ruby-loco for a few months.
2. The OpenSSL-Win64 package version is 1.0.2p. OpenSSL-Win64 is available for 1.1.0i or 1.1.1, I believe it is a vc140 build. It could be downloaded and used for the mswin builds.
3. I do have vc120 and vc140 builds of OpenSSL 1.1.1 on BinTray.

#12 - 12/03/2018 02:02 PM - MSP-Greg (Greg L)

Update:

1. There have been several recent commits re OpenSSL 1.1.1 issues.
2. A new install of MSYS2 will install OpenSSL 1.1.1. A pacman update will do the same. To install an older version, one must download a package (outside of pacman) and install the local file.
3. Re mswin installs, the first file listed at slproweb.com is 1.1.1a.
4. RubyInstaller2 2.5.3 & ruby-loco are both built with 1.1.1.

Hence, please consider something similar to [GitHub PR1998](https://github.com/ruby/ruby/pull/1998), which sets Appveyor testing to use 1.1.1 for mingw & vc140, vc120 is built with 1.0.2.

#13 - 12/03/2018 02:18 PM - MSP-Greg (Greg L)

Sorry for the second message. I did not mention one thing that non Windows users may not be aware of.

Publicly facing Windows Ruby MinGW builds are all self contained. For instance, I locally test on Rubies from 2.2 thru trunk. Those five builds use four different sets of OpenSSL dll's. All that is required to change versions is a change to ENV['PATH'].

JFYI, due to that, building/compiling extension gems can be non-trivial.

#14 - 08/18/2019 06:55 AM - k0kubun (Takashi Kokubun)

- Status changed from Open to Closed

Closing <https://github.com/ruby/ruby/pull/1998#issuecomment-437700528>

Files

av_openssl_1.1.1.patch	2.09 KB	10/18/2018	MSP-Greg (Greg L)
------------------------	---------	------------	-------------------