

<https://gist.github.com/luislavena/f064211759ee0f806c88>

[howto-manually-add-trust-cert-to-rubygems.md](#)

SSL upgrades on rubygems.org and RubyInstaller versions

UPDATE 2014-12-21: RubyGems 1.8.30, 2.0.15 and 2.2.3 have been released. It requires manual installation, please see instructions [below](#).

Hello,

If you reached this page, means you've hit this SSL error when trying to pull updates from RubyGems:

```
SSL_connect returned=1 errno=0 state=SSLv3 read server certificate B:
certificate verify failed
```

This error is produced by changes in rubygems.org infrastructure, please keep reading to better understand it.

If you're one of those *too long, didn't read* just skip to the guide on how to workaround it.

Background

For those who are not familiar with SSL and certificates, there are many parts that make secure serving of content possible.

SSL certificates are used on the website, which are obtained from a certificate authority (CA) and generated from a private key, along with its respective signature.

Normally and up until a few months ago, private key signatures used SHA-1 as way to provide a digest (or checksum) of the private key without distributing the key itself (remember, needs to remain private).

SHA-1 has been encountered weak and lot of web servers and sites have been upgrading towards SHA-2 (specifically SHA256 or higher) in order to prepare for the browsers changes.

Specific problem with RubyGems

The particular case of RubyGems (the command line tool) is that it requires to bundle inside of its code the trust certificates, which allow RubyGems to establish a connection with the servers even when base operating system is unable to verify the identity of them.

Up until a few months ago, this certificate was provided by one CA, but newer certificate is provided by a different one.

Because of this, existing installations of RubyGems would have to be updated before the switch of the certificate and give enough time for the change to spread (and people to update).

As what normally happens with software, things might get out of sync and coordinate such effort, to the size and usage of rubygems.org is almost impossible.

I've described this on [Issue #1050](#)

We had discussed also on IRC, and patches and backports were provided to all major branches of RubyGems: 1.8, 2.0, 2.2 and 2.4

You can find the commits associated with these changes here:

- [1.8 branch](#)
- [2.0 branch](#)
- [2.2 branch](#)
- [2.4 branch](#)

Problem is, only RubyGems 2.4.4 got released, leaving Ruby installation with 1.8, 2.0 and 2.2 in a broken state.

Specially since RubyGems 2.4 is broken on Windows.

Please understand this could happen to anyone. Release multiple versions of *any* software in a short span of time and be very time sensitive is highly complicated.

Even if we have official releases of any of the versions that correct the issue, it will not be possible install those via RubyGems (chicken-egg problem described before).

Once official releases are out, installation might be simpler. In the meantime, please proceed using the instructions described below.

Installing using update packages (NEW)

Now that RubyGems 1.8.x, 2.0.x and 2.2.x have been released, you can manually update to those versions.

First, download the proper version of RubyGems for your installation (eg. if running version 1.8.28, download 1.8.30).

Note: To find the version of RubyGems you're using, please run `gem --version` in the command line.

You can find download links at GitHub under [Releases](#).

Now, locate `rubygems-update-X.Y.Z.gem` where `X.Y.Z` will be the matching version for the version of RubyGems you need to update:

- Running 1.8.x: download [1.8.30](#)
- Running 2.0.x: donwload [2.0.15](#)

- Running 2.2.x: download [2.2.3](#)

Please download the file in a directory that you can later point to (eg. the root of your harddrive C:\)

Now, using your Command Prompt:

```
C:\>gem install --local C:\rubygems-update-1.8.30.gem
C:\>update_rubygems --no-ri --no-rdoc
```

After this, `gem --version` should report the new update version.

You can now safely uninstall `rubygems-update gem`:

```
C:\>gem uninstall rubygems-update -x
Removing update_rubygems
Successfully uninstalled rubygems-update-2.2.3
```

Manual solution to SSL issue

If you have read the above detail that describe the issue, thank you.

Now, you want to manually fix the issue with your installation.

Steps are simple:

- Step 1: Obtain the new trust certificate
- Step 2: Locate RubyGems certificate directory in your installation
- Step 3: Copy new trust certificate
- Step 4: Profit

Step 1: Obtain the new trust certificate

If you've read the previous sections, you will know what this means (and shame on you if you have not).

We need to download [AddTrustExternalCARoot-2048.pem](#).

Use the above link and place/save this file somewhere you can later find easily (eg. your Desktop).

IMPORTANT: File must have `.pem` as extension. Browsers like Chrome will try to save it as plain text file. Ensure you change the filename to have `.pem` in it after you have downloaded it.

Step 2: Locate RubyGems certificate directory in your installation

In order for us copy this file, we need to know where to put it.

Depending on where you installed Ruby, the directory will be different.

Take for example the default installation of Ruby 2.1.5, placed in `C:\Ruby21`

Open a Command Prompt and type in:

```
C:\>gem which rubygems  
C:/Ruby21/lib/ruby/2.1.0/rubygems.rb
```

Now, let's locate that directory. From within the same window, enter the path part up to the file extension, but using backslashes instead:

```
C:\>start C:\Ruby21\lib\ruby\2.1.0\rubygems
```

This will open a Explorer window inside the directory we indicated.

Step 3: Copy new trust certificate

Now, locate `ssl_certs` directory and copy the `.pem` file we obtained from previous step inside.

It will be listed with other files like `GeoTrustGlobalCA.pem`.

Step 4: Profit

There is actually no step 4. You should be able to install Ruby gems without issues now.