

Ruby - Bug #14422

Ruby configuration options should not be reused for gem builds

01/30/2018 07:15 AM - vo.x (Vit Ondruch)

Status:	Closed	
Priority:	Normal	
Assignee:		
Target version:		
ruby -v:	ruby 2.5.0p0 (2017-12-25 revision 61468) [x86_64-linux]	Backport: 2.3: UNKNOWN, 2.4: UNKNOWN, 2.5: UNKNOWN

Description

When Fedora started to harden its packages, we quite often seen complains from our users about problems installing their gems, with errors such as [1](#):

```
gcc: error: /usr/lib/rpm/redhat/redhat-hardened-cc1: No such file or directory
```

The issue as analyzed by Mamoru TASAKA is [2](#):

Well, if I am not mistaken, the real problem here is that rpm's %optflags is always embedded into Fedora's ruby config file, that is

```
/usr/lib64/ruby/rbconfig.rb:167: CONFIG["CXXFLAGS"] = "-O2 -g -pipe -Wall -Werror=format-security
-Wp,-D_FORTIFY_SOURCE=2 -fexceptions -fstack-protector-strong --param=ssp-buffer-size=4 -grecord-gcc-switches
-specs=/usr/lib/rpm/redhat/redhat-hardened-cc1 -mtune=generic"
/usr/lib64/ruby/rbconfig.rb:171: CONFIG["CFLAGS"] = "-O2 -g -pipe -Wall -Werror=format-security
-Wp,-D_FORTIFY_SOURCE=2 -fexceptions -fstack-protector-strong --param=ssp-buffer-size=4 -grecord-gcc-switches
-specs=/usr/lib/rpm/redhat/redhat-hardened-cc1 -mtune=generic -fPIC"
```

on x86_64, for example.

Although I am not sure this is already discussed somewhere or not, basically I think changing the default CFLAGS of "system" ruby like this way is undesirable and ? installed "rbconfig.rb" should have some "minimal" CFLAGS / CXXFLAGS. (for example, just like CONFIG["CFLAGS"] = "-fPIC")

Only when we build Fedora gems or so (on koji), we should change CFLAGS / CXXFLAGS explicitly afterwards using %optflags.

and Red Hat toolchain team responds [3](#):

The current advice of the Red Hat toolchain team is to keep distribution build flags and toolchain default flags separate. This is why running "gcc" gives you the upstream defaults, and not the flags we use to compile Fedora packages. For consistency, Ruby (and other compilation support tools) follow this pattern: Use distribution flags when building for Fedora, but use upstream flags when the user compiles packages (i.e., what Ruby uses, probably something involving -O2).

Our build flags are fully ABI-compatible with each other, so mismatches will not cause any problems at the C/C++/ABI level.

The question is why Ruby does this and how we can avoid this behavior. We could force installation of redhat-rpm-config package, providing the "/usr/lib/rpm/redhat/redhat-hardened-cc1", to every ruby user, but that does not seems right. There are also other similar issues discussing this situation [4](#), [5](#). Any thoughts?

Associated revisions

Revision bb3db69e2a0c210cc3a63940622db96a97eb7947 - 03/02/2018 01:37 AM - nobu (Nobuyoshi Nakada)

configure.ac: library options to MAINLIBS

- configure.ac (MAINLIBS): moved library options for main program and static libruby, and append MAINLIBS to LIBRUBYARG_STATIC, as these libraries are not needed for linking to shared libruby. [ruby-core:85882] [Bug #14422]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@62627 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision bb3db69e - 03/02/2018 01:37 AM - nobu (Nobuyoshi Nakada)

configure.ac: library options to MAINLIBS

- configure.ac (MAINLIBS): moved library options for main program and static library, and append MAINLIBS to LIBRUBYARG_STATIC, as these libraries are not needed for linking to shared library. [ruby-core:85882] [Bug #14422]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@62627 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 0f3452f4c24a9a78b56c53cf646cb776374c99ae - 03/02/2018 07:24 AM - nobu (Nobuyoshi Nakada)

Makefile.sub: link user32.lib

- win32/Makefile.sub: always link user32.lib which provides a lot of very common library functions. [Bug #14422]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@62629 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 0f3452f4 - 03/02/2018 07:24 AM - nobu (Nobuyoshi Nakada)

Makefile.sub: link user32.lib

- win32/Makefile.sub: always link user32.lib which provides a lot of very common library functions. [Bug #14422]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@62629 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision d1d2235ed90af1d49af0b95cd520dcf90d593567 - 03/02/2018 07:49 AM - nobu (Nobuyoshi Nakada)

Makefile.sub: need suffix

- win32/Makefile.sub: fix missing suffix of user32.lib. [Bug #14422]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@62630 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision d1d2235e - 03/02/2018 07:49 AM - nobu (Nobuyoshi Nakada)

Makefile.sub: need suffix

- win32/Makefile.sub: fix missing suffix of user32.lib. [Bug #14422]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@62630 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

History

#1 - 01/30/2018 08:41 AM - nobu (Nobuyoshi Nakada)

vo.x (Vit Ondruch) wrote:

Our build flags are fully ABI-compatible with each other, so mismatches will not cause any problems at the C/C++/ABI level.

Possibly it can be true on Fedora, but no one guarantees it on other platforms, and for different \$(CC) especially, in general.

The question is why Ruby does this and how we can avoid this behavior. We could force installation of redhat-rpm-config package, providing the "/usr/lib/rpm/redhat/redhat-hardened-cc1", to every ruby user, but that does not seem right. There are also other similar issues discussing this situation [4], [5]. Any thoughts?

Options given to configure as XCFLAGS and XLDFLAGS are used to build ruby itself, but not for extension libraries. May this help you?

#2 - 01/30/2018 08:53 AM - Hanmac (Hans Mackowiak)

i had some problem when building wxWidgets gem for ruby (and testing it on osx)

i needed to check that both wxWidgets, ruby and my gem where both build with the same compiler, and in cases of wxWidgets and my gem with same compiler flags like -std

that's why i am unsure about removing configuration options

#3 - 01/30/2018 10:12 AM - shevegen (Robert A. Heiler)

This sounds very much like fedora-specific problems rather than a problem caused by ruby, to be honest.

#4 - 02/01/2018 01:28 PM - vo.x (Vit Ondruch)

nobu (Nobuyoshi Nakada) wrote:

vo.x (Vit Ondruch) wrote:

The question is why Ruby does this and how we can avoid this behavior. We could force installation of redhat-rpm-config package, providing the "/usr/lib/rpm/redhat/redhat-hardened-cc1", to every ruby user, but that does not seems right. There are also other similar issues discussing this situation [4], [5]. Any thoughts?

Options given to configure as XCFLAGS and XLDFLAGS are used to build ruby itself, but not for extension libraries. May this help you?

Well, for configuration of every Fedora package, I should be using standard %configure macro, which by default defines CFLAGS and similar env variables. Of course I might opt to not use the %configure macro, but this means diverging from defaults and possibly causing different issues (e.g. the hardening options will be forgotten etc).

#5 - 02/01/2018 01:33 PM - vo.x (Vit Ondruch)

shevegen (Robert A. Heiler) wrote:

This sounds very much like fedora-specific problems rather than a problem caused by ruby, to be honest.

This issue might suffer any binary distribution sooner or later, be it Fedora, Debian or even Homebrew on Mac. The only case where it doesn't really matter is when you build everything from scratch on the target system, because there you have the whole toolchain and it is run only on that specific environment.

Actually it would have never be issue if the -specs=/usr/lib/rpm/redhat/redhat-hardened-cc1 option, which pulls in some configuration file, was not among the parameters.

#6 - 03/01/2018 10:52 AM - vo.x (Vit Ondruch)

So I have another issue with the configuration options. I tried to enable GMP support in Ruby on Fedora. Therefore, the Ruby build now requires "gmp-devel" and the configuration scripts configures the ruby to link against GMP. So far so good.

But later, when I want to build binary packages, they fails and this is the mkmf.log content:

```
"pkg-config --exists libffi"
| pkg-config --libs libffi
=> "-lffi \n"
"gcc -o conftest -I/usr/include -I/usr/include/ruby/backward -I/usr/include -I. -O2 -g -pipe -Wall -Werror=
format-security -Wp,-D_FORTIFY_SOURCE=2 -Wp,-D_GLIBCXX_ASSERTIONS -fexceptions -fstack-protector-strong -greco
rd-gcc-switches -specs=/usr/lib/rpm/redhat/redhat-hardened-cc1 -specs=/usr/lib/rpm/redhat/redhat-annobin-cc1 -
m64 -mtune=generic -fasynchronous-unwind-tables -fstack-clash-protection -mcet -fcf-protection conftest.c -L.
-L/usr/lib64 -L. -Wl,-z,relro -Wl,-z,now -specs=/usr/lib/rpm/redhat/redhat-hardened-ld -fstack-protector -rd
ynamic -Wl,-export-dynamic -m64 -lruby -lpthread -lgmp -ldl -lcrypt -lm -lc"
/usr/bin/ld: cannot find -lgmp
collect2: error: ld returned 1 exit status
checked program was:
/* begin */
1: #include "ruby.h"
2:
3: int main(int argc, char **argv)
4: {
5:     return 0;
6: }
/* end */
```

Apparently, the "-lgmp" is problematic, since I don't have gmp-devel installed in the buildroot for other packages. The link option should not be propagated for gems. They should pick their requirements by their configuration scripts.

#7 - 03/02/2018 01:37 AM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Closed

Applied in changeset trunk|r62627.

configure.ac: library options to MAINLIBS

- configure.ac (MAINLIBS): moved library options for main program and static libruby, and append MAINLIBS to LIBRUBYARG_STATIC, as these libraries are not needed for linking to shared libruby. [\[ruby-core:85882\]](#) [Bug [#14422](#)]

#8 - 03/05/2018 03:32 PM - vo.x (Vit Ondruch)

- Status changed from Closed to Open

Thx for fixing the latter.

Unfortunately, this does not address the original issue. So is there any hope this will be improved upstream? I would like to avoid some rbconfig.rb hacks to remove the configuration flags in question.

#9 - 08/23/2023 08:07 PM - jeremyevans0 (Jeremy Evans)

- Status changed from Open to Closed

vo.x (Vit Ondruch) wrote in [#note-8](#):

Unfortunately, this does not address the original issue. So is there any hope this will be improved upstream? I would like to avoid some rbconfig.rb hacks to remove the configuration flags in question.

As [@nobu \(Nobuyoshi Nakada\)](#) stated in <https://bugs.ruby-lang.org/issues/14422#note-1>, use XCFLAGS and XLDFLAGS to specify options only for building Ruby, that you do not want used when building extensions. You implied you didn't want to do this, apparently without trying this approach. However, be aware that is the Ruby-supported way of supporting what you want.

[@nobu \(Nobuyoshi Nakada\)](#) pointed out the issues with not using Ruby's CFLAGS or LDFLAGS by default when building Ruby extensions, so we definitely cannot make the behavior you are requesting the default as it would break things on other platforms.

I don't think this is a general issue with building Ruby. If you are building packages for Ruby extensions, you should have all of Ruby's dependencies installed. If Fedora doesn't populate its buildroot with Ruby's dependencies when building packages for Ruby extensions, that does seem like a problem to be fixed in Fedora. I know not all packaging systems have Fedora's problem, because OpenBSD's packaging system does not.