Ruby 1.8 - Bug #1261

cross-compiling Ruby extensions using mkmf doesn't fully respect DESTDIR

03/10/2009 11:25 PM - dangole (Daniel Golle)

Status:	Assigned
Priority:	Normal
Assignee:	nobu (Nobuyoshi Nakada)
Target version:	Ruby 1.8.6
ruby -v:	ruby 1.8.6 (2008-08-11 patchlevel 287) [x86_64-linux]

Description

=begin
I'm having difficulties to cross-compile a ruby extension for a modified version OpenWRT... I couldn't get it working straight, but maybe I'm just getting something wrong...

mkmf doesn't find my Ruby headers because it looks for it only in Config::CONFIG["archdir"] which doesn't work in this case.
It is looking for /usr/lib/ruby/1.8/mips-linux/ruby.h which does not exist.
(actually /usr should not be accessed at all!)

See the rbconfig.rb of the target to understand why.
I changed line 136 in mkmf.rb to make it work:

- if not $extmk and File.exist?(Config::CONFIG["archdir"] + "/ruby.h")
- if not $extmk and File.exist?(Config::CONFIG["topdir"] + "/ruby.h")

But that was just a quick hack (because i got no idea what extmk is used for...). Somehow archdir in rbconfig is different from Config::CONFIG["archdir"] when read from mkmf...
Config::CONFIG["archdir"] in rbconfig.rb "../../../staging_dir/mips/usr/lib/ruby/1.8/mips-linux/"
Config::CONFIG["archdir"] in mkmf.rb "/usr/lib/ruby/1.8/mips-linux/"

strange...

../../../staging_dir/host/lib/ruby/1.8/x86_64-linux/ruby.h would be the host system's header
../../../staging_dir/mips/usr/lib/ruby/1.8/mips-linux/ruby.h is the correct location of the target system's header

extconf.rb get's called from inside a Makefile like this
DESTDIR=../../../staging_dir/mips/ ../../../staging_dir/host/bin/ruby -I ../../../staging_dir/mips/usr/lib/ruby/1.8/mips-linux/ extconf.rb

with my hack above the extensions get's build, but it seems that the linker paths are messed up :(
(it works if I LD_PRELOAD the shared libraries my extension is using)

maybe ENV["DESTDIR"] should be respected in mkmf.rb...?
=end

History

#1 - 03/11/2009 12:05 AM - rogerdpack (Roger Pack)

=begin
does rake compiler help any? http://rubyforge.org/projects/rake-compiler/
=end

#2 - 03/11/2009 12:09 AM - dangole (Daniel Golle)

=begin
i'll give it a go and let you know.
=end

#3 - 03/11/2009 02:55 AM - dangole (Daniel Golle)

=begin
i was fiddling around for hours to make gems, rake and rake-compiler part of the OpenWRT toolchain... now i tried it and it's not so helpful in my situation i believe
cons of rake-compiler for my project:

- i won't use gems but rather the OpenWRT's opkg to manage packages
- it manages it's own cross-compiled versions of ruby, which is unneeded, because i got it already as a part of the OpenWRT build
- it needs that you specify the target platform in the Rakefile. this is really bad, because i'll need to modify the Rakefile whenever OpenWRT supports a new architecture.
- in the current version 0.3.1 it fails cross-compiling for anything else than mingw32 target (see line 40 in cross-ruby.rake)

in the end, rake-compiler also does nothing else than calling ruby -I{...} extconf.rb and then make while unnecessarily complicating things...
i guess it's a nice tool to create gems which run on both, Unix and Windows and you don't want to take care of the whole toolchain (binutils, gcc, ld, ...) yourself. with the OpenWRT build-system I do have a nice tool to take care of all the non-ruby-specific parts of cross-compiling. I modified it to include a host-version of ruby having the same version as used on the target.

i'll revert back to rather not use rake-compiler and please, somebody tell me how to supply a DESTDIR to mkmf.rb... :)

cheers
=end

#4 - 03/11/2009 04:03 AM - luislavena (Luis Lavena)

=begin
On Tue, Mar 10, 2009 at 1:53 PM, Daniel Golle redmine@ruby-lang.org wrote:

> Issue #1261 has been updated by Daniel Golle.
>
> i was fiddling around for hours to make gems, rake and rake-compiler part of the OpenWRT toolchain... now i tried it and it's not so helpful in my situation i believe
>
> cons of rake-compiler for my project:
>
> - i won't use gems but rather the OpenWRT's opkg to manage packages

So you're not trying to generate a ruby gem, a ruby extension or what?

> - it manages it's own cross-compiled versions of ruby, which is unneeded, because i got it already as a part of the OpenWRT build

Well, last time I checked, ruby needs the information in rbconfig to
successfuly build any version of Ruby, cross or not cross platform.

To build ruby extensions (that then are packaged into gems) you need that file.

> - it needs that you specify the target platform in the Rakefile. this is really bad, because i'll need to modify the Rakefile whenever OpenWRT supports a new architecture.

You specify the target of the platform *inside* the block for cross
platform, which only applies if you're compiling the extension cross
platform.

Of course, all that assumes you're building a ruby extension that
needs to link against binaries for that platform you're targeting to.

> - in the current version 0.3.1 it fails cross-compiling for anything else than mingw32 target (see line 40 in cross-ruby.rake)

Is expecting MinGW as compiler, either i386-mingw32-gcc (OSX), or
i585-mingw32msvc-gcc (Ubuntu Linux)

Is not perfect and I'm working in some patches for that.

> in the end, rake-compiler also does nothing else than calling ruby -I{...} extconf.rb and then make while unnecessarily complicating things...

Well, you can simplify in just that statement, but is not complicating
things like you mention. Or at least I'm under the impresion that your
usage is not what the tool was meant for.

> i guess it's a nice tool to create gems which run on both, Unix and Windows and you don't want to take care of the whole toolchain (binutils, gcc, ld, ...) yourself. with the OpenWRT build-system I do have a nice tool to take care of all the non-ruby-specific parts of cross-compiling. I modified it to include a host-version of ruby having the same version as used on the target.

Then if you're cross-compiling successfully your tools are looking for
headers in the wrong place?

> i'll revert back to rather not use rake-compiler and please, somebody tell me how to supply a DESTDIR to mkmf.rb... :)

```
cd /target/directory
ruby ../source/directory/with/extconf.rb --options-here
make
```

The output results is stored there (the binary shared object).
--
Luis Lavena
AREA 17
-
Perfection in design is achieved not when there is nothing more to add,
but rather when there is nothing more to take away.
Antoine de Saint-Exupéry

=end

#5 - 03/11/2009 05:02 AM - dangole (Daniel Golle)

=begin
I agree with you, my usage is not what the tool was meant for. Using it complicated things for me, because I had to include gems and rake as well as rake-compiler in the OpenWRT toolchain to find out that it doesn't really help me.
I'm building a set of Ruby extensions to work on an embedded system. Using doesn't seem like a good approach to me if there already is a native packaging system (opkg), because both space and cpu-time are more constraint on such a system. (I'm targeting the Fonera2 or similarly dimensioned network devices)

Anyway, back to the original issue with mkmf:
I could get it to generate a Makefile using a workaround:
At the first line (*before* require 'mkmf') of extconf.rb I put
DESTDIR="#{ENV['DESTDIR']}"
which makes rbconfig.rb accept it.

=end

#6 - 03/13/2009 02:59 AM - dangole (Daniel Golle)

- *File Makefile added*
- *File mkmf.log added*

=begin
Update:
Including DESTDIR="#{ENV['DESTDIR']}" helps when compiling, and I gained a working shared library from that.
But the Makefile generated by mkmf just prepends the DESTDIR to the prefix which causes a horrible mess when make install is called given another DESTDIR.
This is a common situation when creating stripped-down binary distributions, as we don't want to include all the headers and development files needed for building a packet.
If you agree that this is supposed to work differently and indeed is a Bug, I'd happily work on mkmf.rb to fix it.
=end

#7 - 03/16/2009 03:16 PM - nobu (Nobuyoshi Nakada)

- *Category changed from core to build*
- *Status changed from Open to Assigned*
- *Assignee set to nobu (Nobuyoshi Nakada)*

=begin

=end

#8 - 03/31/2009 02:49 AM - dangole (Daniel Golle)

=begin
I happily noticed that the issue does not exists in Ruby 1.9.1p0. As I'll be using that instead, it doesn't matter if it ever gets fixed in 1.8.6.
=end

Files

rbconfig.rb	7.9 KB	03/10/2009	dangole (Daniel Golle)
rbconfig.rb	7.11 KB	03/10/2009	dangole (Daniel Golle)
Makefile	4.17 KB	03/13/2009	dangole (Daniel Golle)

mkmf.log 28.9 KB 03/13/2009 dangole (Daniel Golle)