

Ruby - Bug #3889

Incorrectly detected i686-w64-mingw32 as x64-mingw

09/30/2010 12:56 AM - luislavena (Luis Lavena)

Status:	Closed	Backport:
Priority:	Normal	
Assignee:	nobu (Nobuyoshi Nakada)	
Target version:	2.0.0	
ruby -v:	1.9.3dev and 1.9.2	
Description		
<pre>=begin Hello, *-w64-mingw32 represent the GCC compilers and headers by mingw-w64 project: http://mingw-w64.sf.net/ This project provides compilers in 2 architectures: i686 for 32bits and x86_64 for 64bits Windows. There has been a previous discussion about this with wanabe at [ruby-core:32565] The latest changes and regularization introduced in r29324, r29347, r29354, r29365 complicates things further than wanabe patch in the above ruby-core thread. 1. configure --host=i686-w64-mingw32 generates "x64-mingw" configuration file This is incorrect, i686 indicates the architecture according to GCC tripled (arch-vendor-os) 1. mingw != mingw32 Dropping 32 of mingw (which is the os) now means possible breakage of RubyGems. \$ ruby -rubygems -ve "puts Gem::Platform.new 'x86-mingw'" ruby 1.9.2p0 (2010-08-18 revision 29036) [x86_64-darwin10.4.0] x86-unknown 1. i686-pc vs x86_64-w64 vs mingw32 vs mingw64 Until today there is no official (ala: mingw.org) 64bits version of GCC for Windows. this is because both teams fractured long ago. the mingw64 shortcut introduced in config.sub has no point as there is no "-pc-" vendor for these compilers It is important to differentiate that i686 and x86_64 has nothing to do with vendor (pc or w64) and os (mingw32) must remain for compatibility. There is not such thing as "mingw64". In an ideal world, it could be, but we don't live in an ideal world. Please consider these changes: 1. --host=i686-w64-mingw32 should be treated as i386-mingw32 2. --host=x86_64-w64-mingw32 should be treated as x86_64-mingw32 3. consider arch and os, please ignore vendor in your config.sub rules. Thank you. =end</pre>		

Associated revisions

Revision 89bd00d8 - 01/21/2011 12:18 AM - luislavena (Luis Lavena)

- configure.in: Fix incorrectly detected x86_64-w64-mingw32 due to canonicalization of target_os. Bug #3889 [ruby-core:32634]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@30620 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

History

#1 - 09/30/2010 12:58 AM - luislavena (Luis Lavena)

=begin

One thing I forgot to mention.

If the revisions I've mentioned above are backported to 1.9.2 (which seems to have been prepared for another release) these will break compilation of RubyInstaller packages.

Thank you.

=end

#2 - 10/03/2010 04:27 PM - luislavena (Luis Lavena)

- File 0001-Remove-mingw64-normalization.patch added

- File 0002-Unify-mkexport-symbols-for-32-64-MinGW.patch added

=begin

Hello,

Please find attached the corrections to tool/config.sub to properly detect i686-w64-mingw32 as i386-mingw32 and x86_64-w64-mingw32 as x86_64-mingw32

Also, find another patch that unifies MinGW 32 and 64 bits mkexports

Please consider them to replace the current ones.

Thank you.

=end

#3 - 10/04/2010 12:07 AM - wanabe (_ wanabe)

- Status changed from Open to Closed

- % Done changed from 0 to 100

=begin

This issue was solved with changeset r29402.

Luis, thank you for reporting this issue.

Your contribution to Ruby is greatly appreciated.

May Ruby be with you.

=end

#4 - 10/05/2010 01:36 PM - usa (Usaku NAKAMURA)

=begin

Who make the final decision?

Wanabe, are you (or, do you want to become) the maintainer of mingw port?

I'm very dissatisfied with the current state, but I don't have a mind to object to the maintainer's decision.

Therefore, I ask it again.

Who is the maintainer and who make the final decision?

=end

#5 - 10/05/2010 01:56 PM - shyouhei (Shyouhei Urabe)

- Status changed from Closed to Open

=begin

Reopening. We need a port maintainer.

=end

#6 - 10/05/2010 08:59 PM - wanabe (_ wanabe)

=begin

2010/10/5, Usaku NAKAMURA redmine@ruby-lang.org:

Who make the final decision?

Wanabe, are you (or, do you want to become) the maintainer of mingw port?

No, I am not. (and I do not want to)

I'm very dissatisfied with the current state, but I don't have a mind to object to the maintainer's decision.

Therefore, I ask it again.

Who is the maintainer and who make the final decision?

I'm very sorry to commit r29320 hastily.

This is really, really my fault.

I asked for a Nakada-san's advice of this issue.

Because he is the maintainer of mingw32.

He said config.sub should be reverted to automake-1.11.1's.

I did it because I thought it can be covering my losses.

And the change of win32/mkexports.rb is my fault.

I guessed that there is no effect for other platform by this change.

But now I realize that I exceeded my authority.

I want to ask back to the maintainer of win32/mkexports.rb (who?) for acceptability of it.

I guess I should revert the change until hear this decision, shouldn't I?

--

wanabe

=end

#7 - 10/05/2010 10:26 PM - wanabe (_ wanabe)

=begin

I committed r29411 to revert win32/mkexports.rb.

Sorry again.

=end

#8 - 10/05/2010 11:15 PM - luislavena (Luis Lavena)

=begin

Hello,

Please apologize my lack of understanding of Ruby development hierarchy, but the revert of 'mingw' specifics in mkexports breaks the mingw-w64 work.

Who is the maintainer that provides "best effort" in relation to MinGW support? I would like to discuss this and future MinGW work without causing you guys lot of conflicts.

Thank you.

=end

#9 - 10/06/2010 02:16 AM - tenderlovemaking (Aaron Patterson)

=begin

On Tue, Oct 05, 2010 at 08:59:06PM +0900, wanabe wrote:

2010/10/5, Usaku NAKAMURA redmine@ruby-lang.org:

Who make the final decision?

Wanabe, are you (or, do you want to become) the maintainer of mingw port?

No, I am not. (and I do not want to)

I'm very dissatisfied with the current state, but I don't have a mind to object to the maintainer's decision.

Therefore, I ask it again.

Who is the maintainer and who make the final decision?

I'm very sorry to commit r29320 hastily.
This is really, really my fault.

I asked for a Nakada-san's advice of this issue.
Because he is the maintainer of mingw32.

Maybe we should give Luis commit access. He does so much work shipping mingw builds. It seems that he would be a good person to help.

--

Aaron Patterson
<http://tenderlovmaking.com/>

Attachment: (unnamed)
=end

#10 - 10/06/2010 02:17 AM - tenderlovmaking (Aaron Patterson)

=begin
On Tue, Oct 05, 2010 at 02:03:23PM +0900, Shyouhei Urabe wrote:

Issue [#3889](#) has been updated by Shyouhei Urabe.

Status changed from Closed to Open

Reopening. We need a port maintainer.

I think Luis should be the port maintainer. Luis, what do you think?

--

Aaron Patterson
<http://tenderlovmaking.com/>

Attachment: (unnamed)
=end

#11 - 10/06/2010 03:41 AM - luislavena (Luis Lavena)

=begin
On Tue, Oct 5, 2010 at 2:17 PM, Aaron Patterson
aaron@tenderlovmaking.com wrote:

On Tue, Oct 05, 2010 at 02:03:23PM +0900, Shyouhei Urabe wrote:

Issue [#3889](#) has been updated by Shyouhei Urabe.

Status changed from Closed to Open

Reopening. We need a port maintainer.

I think Luis should be the port maintainer. Luis, what do you think?

Thank you Aaron for the vote, but I'm interested first in understand what are the roles and duties of a port maintainer.

100% of the work done for RubyInstaller is around mingw and mingw-w64 versions of GCC, I can provide a daily test base and work on this.

However, on this same thread wanabe committed a patch I generated which only touched mingw section of mkexports. If I has given commits bits, I would have done the same.

I would like to understand better the roles and responsibilities before offering myself as maintainer in the light of possible "mistakes" I could make in the quest of a better MinGW support for Ruby.

Thank you.

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

#12 - 10/06/2010 11:50 AM - tenderlovmaking (Aaron Patterson)

=begin

On Wed, Oct 06, 2010 at 03:41:01AM +0900, Luis Lavena wrote:

On Tue, Oct 5, 2010 at 2:17 PM, Aaron Patterson
aaron@tenderlovmaking.com wrote:

On Tue, Oct 05, 2010 at 02:03:23PM +0900, Shyouhei Urabe wrote:

Issue [#3889](#) has been updated by Shyouhei Urabe.

Status changed from Closed to Open

Reopening. We need a port maintainer.

I think Luis should be the port maintainer. Luis, what do you think?

Thank you Aaron for the vote, but I'm interested first in understand
what are the roles and duties of a port maintainer.

100% of the work done for RubyInstaller is around mingw and mingw-w64
versions of GCC, I can provide a daily test base and work on this.

However, on this same thread wanabe committed a patch I generated
which only touched mingw section of mkexports. If I has given commits
bits, I would have done the same.

It seems to me (maybe I am wrong) that wanabe only reverted because he
is not the mingw maintainer. As far as I can tell, nobody is. If you
were the mingw maintainer, I think you would decide how to fix mingw
problems.

I would like to understand better the roles and responsibilities
before offering myself as maintainer in the light of possible
"mistakes" I could make in the quest of a better MinGW support for
Ruby.

I think your responsibility would be "make sure ruby works with mingw".
;-)

--

Aaron Patterson
<http://tenderlovmaking.com/>

Attachment: (unnamed)
=end

#13 - 10/07/2010 03:33 PM - usa (Usaku NAKAMURA)

=begin

Hello,

In message "[[ruby-core:32695](#)] [Ruby 1.9-Bug#3889] Incorrectly detected i686-w64-mingw32 as x64-mingw"
on Oct.05,2010 23:16:07, redmine@ruby-lang.org wrote:

Please apologize my lack of understanding of Ruby development hierarchy, but the revert of 'mingw' specifics in mkexports breaks the
mingw-w64 work.

Oh, you don't have to apologize.

(And, of course, wanabe-san doesn't, too.)

Who is the maintainer that provides "best effort" in relation to MinGW support? I would like to discuss this and future MinGW work without causing you guys lot of conflicts.

Agree.

The problem is not wanabe-san's revert, but a lack of responsibility of MinGW port.

Currently, nobu is the maintainer of MinGW port.

I know him well, and everyone knows that he is the special about hacking ruby.

However, he is not living on Windows now.

Moreover, he doesn't have 64bit Windows environment.

I guess that it is too difficult to maintain it in such situation even if he is the special.

We should change the current state in some shape.

Regards,

U.Nakamura usa@garbagecollect.jp

=end

#14 - 10/07/2010 09:01 PM - luislavena (Luis Lavena)

=begin

On Thu, Oct 7, 2010 at 3:33 AM, U.Nakamura usa@garbagecollect.jp wrote:

Hello,

In message "[[ruby-core:32695](#)] [Ruby 1.9-Bug#3889] Incorrectly detected i686-w64-mingw32 as x64-mingw"

Who is the maintainer that provides "best effort" in relation to MinGW support? I would like to discuss this and future MinGW work without causing you guys lot of conflicts.

Agree.

The problem is not wanabe-san's revert, but a lack of responsibility of MinGW port.

Currently, nobu is the maintainer of MinGW port.

I know him well, and everyone knows that he is the special about hacking ruby.

However, he is not living on Windows now.

Moreover, he doesn't have 64bit Windows environment.

I guess that it is too difficult to maintain it in such situation even if he is the special.

We should change the current state in some shape.

Hello Mr. Nakamura,

Thank you for your answers.

I'm actively using MinGW/mingw-w64 in both native (Windows) and for cross-compilation (Linux/OSX targeting Windows)

All the work towards 64bits Ruby under MinGW is still young, but all the cross-compilation issues are needed right now to simplify things for Ruby developers and their Windows support.

I'm willing to provide instructions to nobu so he can cross-compile exactly the same way I'm doing from my OSX computer (or a Linux one, doesn't matter)

As for the native versions, I would happily report issues and provide patches to any encountered problem that blocks compilation or execution, has been done already over the past years.

I will be very happy if all the existing issues for the which I provided patches can be reviewed.

I'll happily accept become a maintainer if that is required to ensure

a proper MinGW support for Ruby.

Thank you.

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

#15 - 12/26/2010 02:53 PM - naruse (Yui NARUSE)

=begin

What's going on?

=end

#16 - 12/27/2010 05:26 AM - luislavena (Luis Lavena)

=begin

Hello,

The 32bits (i686-w64-mingw32) issue has been solved, and it correctly generates i386-mingw32 as RUBY_PLATFORM

However, the 64bits version one still fails as it generates x64-mingw64 which is incorrect.

Either the platform should be 'x86_64-mingw32' or 'x64-mingw32', to match VC x64 builds. There is an eternal discussion on mingw and mingw-w64 mailing lists about how 'mingw32' stuck and how hard is to change it. Usage of 'mingw64' do not correspond to any GNU triplet, as mentioned before.

Steps to reproduce this issue:

1. Download a Linux or Darwin binaries for x86_64-w64-mingw32 from the Automated build page:

<http://sourceforge.net/projects/mingw-w64/files/Toolchains%20targetting%20Win64/Automated%20Builds/>

(I've used mingw-w64-1.0-bin_i686-linux_20101224.tar.bz2)

1. Extract into a folder:

```
mkdir -p ~/mingw/w64
```

```
cd ~/mingw/w64
```

```
tar xf ../mingw-w64-1.0-bin_i686-linux_20101224.tar.bz2
```

1. prepend 'bin' directory into the PATH:

```
export PATH=~/mingw/w64/bin:$PATH
```

1. build ruby

```
cd ~/ruby
```

```
autoconf
```

```
mkdir build64 && cd build64
```

```
sh ../configure --enable-shared --disable-install-doc --host=x86_64-w64-mingw32
```

1. Notice the generated configuration path:

```
./ext/include/x64-mingw64/ruby/config.h updated
```

```
$ cat ./ext/include/x64-mingw64/ruby/config.h | grep PLAT
```

```
#define RUBY_PLATFORM "x64-mingw64"
```

===

I'm going to attempt correct this issue but my autoconf knowledge is not my best skill.

=end

#17 - 12/27/2010 06:40 AM - luislavena (Luis Lavena)

- Assignee set to usa (Usaku NAKAMURA)

=begin

Hello,

The following patch resolves the x64-mingw64 issue described in previous comment:

```
diff --git a/configure.in b/configure.in
index 3a1999c..1093a92 100644
--- a/configure.in
+++ b/configure.in
@@ -39,9 +39,7 @@ AS_CASE(["$target_os"], [mingwmsvc], [
  target_os="echo ${target_os} | sed 's/msvc$/'"
 ])
 AS_CASE(["$target_cpu-$target_os"], [x86_64-mingw], [
  -# canonicalize as like mswin version. see win32/setup.mak.
  target_cpu=x64
  -target_os="echo ${target_os} | sed 's/32$/64/'"
 ])
])
```

I would like to ask Mr. Usaku NAKAMURA to review it as mswin64 != mingw64, so there is no reason for canonicalize it as VC builds.

Of course, I might be missing something down the line, but will appreciate the feedback.

Thank you.
=end

#18 - 12/27/2010 09:33 AM - usa (Usaku NAKAMURA)

=begin
Hello,

In message "[[ruby-core:33911](#)] [Ruby 1.9-Bug#3889] Incorrectly detected i686-w64-mingw32 as x64-mingw"
on Dec.27,2010 06:41:09, redmine@ruby-lang.org wrote:

I would like to ask Mr. Usaku NAKAMURA to review it as mswin64 != mingw64, so there is no reason for canonicalize it as VC builds.

me? not nobu?

I am the maintainer of mswin32/mswin64, but not of mingw.

Ah, the name of port "mswin64" means that it is targeted to use Win64 API set.
"mswin32" and "mswince" means similar.
But "mingw*" does not seem to be similar, and the meaning is fundamentally different from the names of other platforms.
I cannot suggest any logical proposal with this state.
After all, the maintainer of this platform might draw a conclusion according to his reason.

Regards,

U.Nakamura usa@garbagecollect.jp

=end

#19 - 12/27/2010 11:13 AM - luislavena (Luis Lavena)

- Assignee changed from usa (Usaku NAKAMURA) to nobu (Nobuyoshi Nakada)

=begin

=end

#20 - 12/27/2010 11:13 AM - luislavena (Luis Lavena)

=begin

On Sun, Dec 26, 2010 at 9:33 PM, U.Nakamura usa@garbagecollect.jp wrote:

Hello,

In message "[[ruby-core:33911](#)] [Ruby 1.9-Bug#3889] Incorrectly detected i686-w64-mingw32 as x64-mingw"
on Dec.27,2010 06:41:09, redmine@ruby-lang.org wrote:

I would like to ask Mr. Usaku NAKAMURA to review it as mswin64 != mingw64, so there is no reason for canonicalize it as VC builds.

me? not nobu?

I am the maintainer of mswin32/mswin64, but not of mingw.

I understand, but according to the comment, canonicalize according to mswin*, was my understanding you discussed this with Nobu.

Ah, the name of port "mswin64" means that it is targeted to use Win64 API set.
"mswin32" and "mswince" means similar.
But "mingw*" does not seem to be similar, and the meaning is fundamentally different from the names of other platforms.
I cannot suggest any logical proposal with this state.
After all, the maintainer of this platform might draw a conclusion according to his reason.

Thank you for your observations, assigning to Nobu for approval.

Regards,

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

#21 - 01/21/2011 09:27 AM - luislavena (Luis Lavena)

- Status changed from Open to Closed

=begin
This issue was solved with changeset r30620.
Luis, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

-
- configure.in: Fix incorrectly detected x86_64-w64-mingw32 due canonicalization of target_os. Bug [#3889](#) [[ruby-core:32634](#)]
- =end

Files

0001-Remove-migw64-normalization.patch	1.22 KB	10/03/2010	luislavena (Luis Lavena)
0002-Unify-mkexport-symbols-for-32-64-MinGW.patch	1.11 KB	10/03/2010	luislavena (Luis Lavena)