

Ruby master - Misc #12124

Use Automake

02/28/2016 08:24 AM - cjcollier (C.J. Collier)

Status:	Open
Priority:	Normal
Assignee:	cjcollier (C.J. Collier)
Description	
It looks like there is a lot of duplicate code that could be removed by making use of automake.	
Are there any reasons why this should not be done?	
I've got some patches up for review here:	
https://github.com/ruby/ruby/compare/trunk...LLC-Technologies-Collier:trunk?expand=1	

History

#1 - 02/29/2016 01:14 AM - shyouhei (Shyouhei Urabe)

- I'm afraid automake contaminates this project with GPL.
- I'm afraid this world is not built on top of GNU. AFAIK automake forces us to use GNU make, which breaks compatibility.

#2 - 02/29/2016 06:21 AM - normalperson (Eric Wong)

shyouhei@ruby-lang.org wrote:

- I'm afraid automake contaminates this project with GPL.

Not true, automake gives something like the following:

```
# This Makefile.in is free software; the Free Software Foundation
# gives unlimited permission to copy and/or distribute it,
# with or without modifications, as long as this notice is preserved.
```

Either the above or special exception (same thing we already have for autoconf-generated stuff).

#3 - 02/29/2016 06:48 AM - naruse (Yui NARUSE)

I generally for this, but ...

C.J. Collier wrote:

It looks like there is a lot of duplicate code that could be removed by making use of automake.

Are there any reasons why this should not be done?

Just because Ruby was born before automake (and matz seems not be familiar with automake when he introduced configure.in).

I've got some patches up for review here:

<https://github.com/ruby/ruby/compare/trunk...LLC-Technologies-Collier:trunk?expand=1>

The patch breaks version.h's RUBY_RELEASE_DATE mechanism.

Shyouhei Urabe wrote:

- I'm afraid automake contaminates this project with GPL.

"Automake places no restrictions on the distribution of the resulting Makefile.ins."

http://www.gnu.org/savannah-checkouts/gnu/automake/manual/html_node/Distributing.html

- I'm afraid this world is not built on top of GNU. AFAIK automake forces us to use GNU make, which breaks compatibility.

With AM_INIT_AUTOMAKE([-Wportability foreign 1.9]) or something, you can keep portability.

#4 - 02/29/2016 08:31 AM - shyouhei (Shyouhei Urabe)

On 02/29/2016 03:17 PM, Eric Wong wrote:

shyouhei@ruby-lang.org wrote:

- I'm afraid automake contaminates this project with GPL.

Not true, automake gives something like the following:

This Makefile.in is free software; the Free Software Foundation gives unlimited permission to copy and/or distribute it, with or without modifications, as long as this notice is preserved.

Either the above or special exception (same thing we already have for autoconf-generated stuff).

What was in my mind was not Makefile.in but config.sub and other files that slip into a project. They are not generated, rather pure redistributions of GPLed software.

But it seems the current config.sub comes in GPL with exception clause so it might be OK. Not checked everything that ships with automake though. I'd like to withdraw this first point.

#5 - 03/03/2016 09:49 AM - naruse (Yui NARUSE)

Just I remind,

Ruby's Makefile must support nmake.

I know automake can generate Makefile which runs with bsdmake, but don't know with nmake.

#6 - 03/17/2016 10:21 AM - shyouhei (Shyouhei Urabe)

We looked at this issue on this month's developer meeting. Attendees agree that automake is basically a good thing to have. However:

- We also have to support nmake.
- In order to do so, we have a Makefile named common.mk which includes most part of rules that are in common between nmake and others.
- Automake migration should not break this part. The difficulty is here.

#7 - 03/17/2016 10:41 AM - cjcollier (C.J. Collier)

Shyouhei Urabe wrote:

We looked at this issue on this month's developer meeting. Attendees agree that automake is basically a good thing to have. However:

- We also have to support nmake.
- In order to do so, we have a Makefile named common.mk which includes most part of rules that are in common between nmake and others.
- Automake migration should not break this part. The difficulty is here.

That all sounds reasonable, and I'm happy to hear that there is interest in the community.

It will likely be much less effort if we get some buy-in from the nmake folks as well. To that end, I've reached out to a friend who was on the Visual Studio team for quite a long time. I hope to hear back from someone in Redmond who can help us to bridge the gap here.

#8 - 04/15/2016 10:09 PM - cjcollier (C.J. Collier)

C.J. Collier wrote:

Shyouhei Urabe wrote:

We looked at this issue on this month's developer meeting. Attendees agree that automake is basically a good thing to have. However:

- We also have to support nmake.
- In order to do so, we have a Makefile named common.mk which includes most part of rules that are in common between nmake and others.
- Automake migration should not break this part. The difficulty is here.

That all sounds reasonable, and I'm happy to hear that there is interest in the community.

It will likely be much less effort if we get some buy-in from the nmake folks as well. To that end, I've reached out to a friend who was on the Visual Studio team for quite a long time. I hope to hear back from someone in Redmond who can help us to bridge the gap here.

I had a meeting yesterday at Microsoft in Redmond with members of the Visual Studio team. They are excited by the idea of enabling support for autotools in Visual Studio. We spoke about the possibility of releasing the nmake source code under a license such as X11/MIT. I am hopeful about the possibility, but of course Microsoft is a large company and they will need to acquire clearance from the developers who wrote the code, management of the Visual Studio team, and LCA before being able to make such a thing possible.

If this were to happen, I expect that it would be relatively easy to convince the automake team to add native support for another Free Software make platform.

Thank you all for your patience!

C.J.

#9 - 04/19/2016 04:01 AM - naruse (Yui NARUSE)

C.J. Collier wrote:

I had a meeting yesterday at Microsoft in Redmond with members of the Visual Studio team. They are excited by the idea of enabling support for autotools in Visual Studio.

That sounds interesting.

If autotools support of Visual Studio, it will ease the maintenance of configure.in/Makefile.in!
(though it needs some Unix tools like grep.exe and sed.exe, and organize current win32/Makefile.sub into new separation)

We spoke about the possibility of releasing the nmake source code under a license such as X11/MIT. I am hopeful about the possibility, but of course Microsoft is a large company and they will need to acquire clearance from the developers who wrote the code, management of the Visual Studio team, and LCA before being able to make such a thing possible.

If this were to happen, I expect that it would be relatively easy to convince the automake team to add native support for another Free Software make platform.

It's great!

We sometimes hit nmake compatibility issue like folloing.
I wish it would help to avoid such issue.

```
e56b9b4 common.mk: dependency of ripper.c
655b18e common.mk: source dependency for nmake
184c7e6 common.mk: probes.dmyh for nmake
f6dcbf7 nmake VPATH
52ddf1f Revert r53482 "nmake VPATH"
64e2285 nmake VPATH
445e11d probes.h including dummy header
a1115a1 * ext/ripper/depend: Just like BSDmake, nmake also recognize the rule of ".eventids2.check" as inference one. but nmake is not cheated by macro. this fixes build failure introduced at r53448. see also the commit log of r53452.
fab8b4 enc/Makefile.in: get rid of nmake bug
bae170f common.mk: double quotes
```